

JAVA FULL STACK DEVELOPMENT

A PROJECT REPORT

Submitted by

V LATHA

*in partial fulfillment for the award of the degree
of*

MASTER OF COMPUTER APPLICATIONS



ETHIRAJ COLLEGE FOR WOMEN, EGMORE

affiliated to

UNIVERSITY OF MADRAS, CHENNAI 600 005

MAY 2021

BONAFIDE CERTIFICATE

ETHIRAJ COLLEGE FOR WOMEN (AUTONOMOUS)



This is to certify that the report entitled

JAVA FULL STACK DEVELOPMENT

Being submitted to the Ethiraj College for Women (Autonomous)

Affiliated to the University of Madras, Chennai

by

LATHA V

1813323037017

Submitted in partial fulfilment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Signature of the Guide

Head of the Department

Place: Chennai

Date: 08/05/2021

Submitted for the viva-voce examination at.....

on.....

Examiner – 1.....

(Signature and Name of the Examiner)

Examiner – 2.....

(Signature and Name of the Examiner)

CERTIFICATE OF ORIGINALITY

I hereby declare that the project entitled **JAVA FULL STACK DEVELOPMENT** submitted for the **MCA DEPARTMENT, ETHIRAJ COLLEGE FOR WOMEN (AUTONOMOUS)** in partial fulfilment for the award of the degree of **MASTER OF COMPUTER APPLICATIONS** in session 2019-2021 is an authentic record of my own work carried out under the guidance of **Dr. Shantha Visalakshi U, MCA, MPhil, PhD.**, and that the project has not previously formed the basis for the award of any other degree.

Signature of the candidate,

Place: Chennai

LATHA V

Date: 08/05/2021

1813323037017

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

1. Signature of Internal guide

Name

Designation

2. Signature of External Guide

Name

Designation

ACKNOWLEDGEMENT

I praise the god almighty for his unfailing source of strength in completion of this project. I wish to acknowledge great gratitude towards my management.

I take immense pleasure in expressing my sincere gratitude and heartfelt thanks to Principal, Ethiraj College for Women for her constant encouragement.

I express my deep gratitude to **Mrs. JOSEPHINE ANITHA, MCA, M.Phil.** Head of Department of Master of Computer Application and **Dr. Shantha Visalakshi U, MCA, MPhil, PhD.,** and internal guide for her moral support for helping me out with her timely advice.

I am grateful to my family and friends who have extend their support and cooperation during my project for the completion. And finally, I thank all who directly or indirectly helped in successful completion of my project.

INTERNSHIP OFFER LETTER

December 22nd 2020

To,

Ms. V Latha
No 17, Jayalalitha Street,
Bhavani Nagar,
Chennai 600 077.

Dear Ms. V Latha,

Sub: Internship Letter for the position of Intern

Welcome to the SRM Technologies.

We are pleased to inform you that, you have been selected as "Intern". The duration of internship is approximately for a period of 3 months from 18 January 2021 and the end date of internship may change based on the requirement. During this internship, you will be paid a stipend of Rs.15,000/- (Rupees Fifteen thousand only) per month.

You will follow the company rules and regulations mentioned in the annexure.

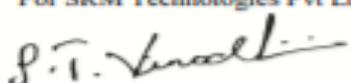
You will be evaluated during and after the internship program. You may be considered for employment with us based on your performance in the internship program.

You are requested to report to our office on 18 January 2021 at 9:30 am.

A copy of this Internship letter with annexure is herewith enclosed. Please sign and return one copy as a token of your acceptance.

Thanking you

Yours faithfully,
For SRM Technologies Pvt Ltd



Vinodhini Sivaraman
Senior Manager – Talent Acquisition

Terms and Conditions (Annexure) attached to Internship letter dated January 18, 2021

1. We will arrange training programs on communication skills and you are required to undergo this training mandatorily.
2. You will not accept any other training or employment, part-time or otherwise, nor should you be interested, directly or indirectly, in any business activity likely to compete with those in which the company is interested, during your internship with us.
3. The Company also reserves its right to alter or modify its training schedule/program.
4. All programs, system designs, manuals, literature etc. developed by you, during your services at the company will, at all times, be deemed to be the exclusive property of the Company.
5. Any technical and non-technical information of confidential nature, concerning the company, its associates or clients and their products and services, must not be communicated directly or indirectly to any person, firm or company that are not entitled or authorized to receive the same, during your service and also after separation from the company.
6. Except as authorized in writing by the Company, you will not duplicate any software / document belonging to the Company or take them out of the Company's premises for your personal use. You will not also load any unauthorized copies of software either in the computers at the Company's premises or at the Client's premises except with the prior approval and authorization by the Company to do so.
7. While joining the training program, please produce for our inspection together with copies of the following documents for our record and reference: -
 - Educational Certificates
 - Proof of previous employment, if any
 - Four Passport size Colour Photographs
8. Our Dress Code: _

Gentlemen are required to wear full-sleeved shirts, trousers, tie and leather shoes (black or brown).

Ladies are required to wear sarees or salwar kameez or any such formal attire ensuring elegance and dignity.



SRM Technologies Pvt. Ltd.

Global Headquarters: Yathrammal Building, Plot No.20, North Phase Developed Plots, Industrial Estate, Ekkaduthangal, Guindy,

Chennai - 600 032, Tamil Nadu, INDIA. Tel +91-98-4030 3000

Branch Office : Landa Block - B Plot No. 1 G 2 (2F), Northern Extension Area, Thirumala Industrial Estate, Ekkaduthangal, Guindy,

Chennai - 600 032, Tamil Nadu, INDIA. Tel +91-94-4036 9000

Registered Office: No. 24, G.N.Chetty Road, Thiragar, Chennai - 600 017, Tamil Nadu, INDIA. E. CR Number U03000971N000971C0404097

www.srmtech.com

3 of 3

A copy of this internship letter is herewith enclosed. Please sign and return one copy as a token of your acceptance.

We welcome you on board and look forward for a long term employment with us.

Yours faithfully,
For SRM Technologies Pvt Ltd



Authorised Signatory

I have read, understood and I accept the above terms and conditions of employment mentioned above.



SRM Technologies Pvt. Ltd.

Global Headquarter: V-Banner Building, Plot No.203, North Phase Developed Plots, Industrial Estate, Ekkaduthangal, Guntur,
Chennai - 600 032, Tamilnadu, INDIA. Tel: +91-9840530000

Branch Office : Lord's Block - B Plot No. 1 & 2 (NP), Northern Extension Area, Thirumalai Industrial Estate, Ekkaduthangal, Guntur,
Chennai - 600 032, Tamilnadu, INDIA. Tel: +91-9446088900

Registered Office: No.204, G.N.Chetty Road, Thiragar, Chennai - 600 027, Tamilnadu, INDIA. E-OR number U000097H0008PTC042467
www.srmtech.com

PROJECT COMPLETION CERTIFICATE

TABLE OF CONTENTS

CHAPTER NO.	CONTENTS	PAGE NO.
	Bonafide Certificate	i
	Certificate of Originality	ii
	Acknowledgement	iii
	Offer Letter	iv
	Industry/Company Project Certificate	v
1	Introduction	
	1.1 Abstract	1
	1.2 About the Company	12
	1.3 Existing System	
	1.4 Proposed System	
2	System Analysis	
	2.1 Requirement Specification	
	2.2 Hardware Requirement	
	2.3 Software Requirements	
	2.4 Software Description	
3	System Design	
	3.1 Data flow Diagram	
	3.2 Tables Description	
	3.3 Data Dictionary	
4	SYSTEM ARCHITECTURE	
	4.1 Overall Architecture	
5	MODULE DESCRIPTION	
	5.1 Home Module	
	5.2 Admin Module	
	5.3 Checker Module	
	5.4 Supplier Module	
	5.5 Checker Module	
	5.6 Manager Module	

6	IMPLEMENTATION	
	6.1 Business Scenario 1	
	6.2 Business Scenario 2	
	6.3 Business Scenario 3	
	6.4 Business Scenario 4	
7	TESTING	
	7.1 Testing Strategy Adapted	
8	Reports	
9	Future Enhancements	
10	Conclusion	
11	List of Tables	
12	List of Figures	
13	References	

1. INTRODUCTION

1.1 ABSTRACT

Java Full Stack Development is the Skill Development which helps developers in working with the Presentation Layer – The Front End, the Logical Layer – The Back End and the Database Layer – The Database. Java Full Stack Developer is the one with intense knowledge and expertise in full stack tools and framework that effectively works with java. Full Stack Developer is the one who works both on the client-side and server-side of the software application. Full stack developer translates user requirements into the overall architecture and implement the new systems. The Focus is mainly on the Front-end development, Back-end development, Database, Server, API, and version controlling systems. Java Full Stack Development gives an extensive knowledge on Core Java, Front End Engineering, Server-Side Engineering, Data Engineering and DevOps etc. The tools used are Git, Dockers and Jenkins etc. A full-stack developer adds value to the organization and team, owing to developing a varied skill-set and the ability to work on a project independently, thereby reducing the operational costs. The Frameworks include Angular JS, Spring Boot, Hibernate, Java Persistence API etc.

Angular:

The grocery business is a rapidly changing area of the retail industry. The rising tide of digital commerce is creating unprecedeted roadblocks for many grocery store owners worldwide.

The Customer E-mart is into Grocery business by running chain of stores across the NA Region, and does quite a lot of retail and bulk sales in given day. The Customer is facing challenge in terms of managing the inventory it gets on the daily or regular basis. On one side, the products arrive from the distributors or direct sellers into the customer warehouse, and on the other side, the same product is being stacked in the shelf for the people to buy.

The challenge the customer faces are • At any point of time what is the stock available in the inventory (i.e., in warehouse), and available in the shelf • The edible stocks that does not move too fast, are tend of get spoiled too frequently. The system should give warning 2-3 days in advance when that situation arises. • To categories & identify the damaged goods • Whether during transport

- During shopping or by expiry dates • Invoice to the suppliers to be managed for every stock that moves in and reverse invoices for the goods that moves out • Alert should be made to the administrator, when some stocks go below the threshold.

1.2 ABOUT THE COMPANY

SRM Technologies is a global IT services provider offering Cloud and Infrastructure, Digital Transformation, Managed IT Services, Application Lifecycle, Quality Assurance, eCommerce and Product Engineering services.

SRM Technologies, part of the SRM Group, was established in 1998 and provides Cloud and Infrastructure, Digital Transformation, Managed IT Services, Application Lifecycle, Quality Assurance, eCommerce and Product Engineering services. These are offered to the Education, Automotive, Manufacturing, Consumer, Transportation & Logistics, Supply Chain and Healthcare industries.

The SRM Group is a multinational conglomerate that has been in business for over four decades and operates in the Education, Technology, Healthcare and Media sectors.

Value Proposition.

1. COMPREHENSIVE SOLUTIONS

Comprehensive solutions across the spectrum that includes Cloud, Application Lifecycle (Development and Maintenance), Quality Assurance, eCommerce through to Product Engineering (embedded, hardware and mechanical). This is backed by Centres of Excellence (CoE's) in capabilities and languages too.

2. IN-HOUSE AUTOMATION

In-house automation framework that brings consumer-grade experience to the enterprise through a services catalog model. Results in zero-touch, zero-wait and superior customer satisfaction.

3. DIGITAL FIRST

Our Digital First mindset helps enterprises embark on a digital transformation journey. We optimize heritage application debt, unlock value from core systems, digitize customer facing layer and innovate in application development. This ensures you keep pace with technology developments and modernize your IT landscape. You get an omni-channel interface too. This is delivered by a

multi-faceted team with the capabilities digitization requires (Software and DevOps engineers, Product managers and Engineering managers).

4. HOLISTIC SOLUTION

Operating at the intersection of domain, technologies and innovation helps us provide a holistic solution to the most pressing enterprise problems.

The key industries that benefit from our offerings include Automotive, Education, Manufacturing, Consumer, Transportation & Logistics, Supply Chain and Healthcare industries. We offer our customers the ability to not only manage their existing IT landscape but also help them embrace the digital future - by crafting their roadmap to enhance frontend customer touchpoints, in addition to their backend processes.

The passionate employees exemplify core belief - ‘ideas at work’. At the heart of it, is our belief that ideas and the resulting innovation are useful only when they make a difference to our customer’s business and the lives of their end-customers. SRM Technologies believe in the practical application of these ideas and their transformative impact. SRM Technologies are part of the SRM Group, a multinational conglomerate that has been in business for over four decades and operates in the Education, Technology, Healthcare and Media sectors. Disclaimer: All references to other trademarks belonging to third parties or referring to other companies, that appear on the design, shall be understood to refer to those registered trademarks owned by others, and not to any trademark belonging to SRM Technologies.

- SRM Technologies is more than just about cost or quality or timely delivery. Rather we view ourselves as a crucial partner to enterprises, helping them solve their most pressing business and technology challenges by operating at the intersection of domain, technologies and innovation.
- Eliminate, simplify, standardize, industrialize and automate “Application Debt” through a combination of our Portfolio Analysis and Application Debt Analyzer and digitalization platform.
- Help enterprises embark on a Digital Transformation journey by bringing together the right ingredients - product manager, software and UI/UX engineer, DevOps engineer, best practices, agile methodology, design for security and leading-edge technologies.
- eCommerce platform development spanning the three components of customer facing, business facing and operations facing systems.

- Legacy modernization through application enhancement and development.
- Deep business modeling to provide enterprises with the insights for data driven decision making.
- Develop eCommerce Data Hub that can be leveraged to build various business drivers.
- Supply Chain transformation to make it resilient, agile and intelligent.
- Developed Advanced Driver Assistance Systems (ADAS).

Cloud and Infrastructure Services

SRM Technologies in the midst of a major digital shift, and the twin forces of rapidly evolving technology and changing user and business expectations is accelerating this. It is forcing enterprises to digitize rapidly. We are a new age digital service provider that helps customers navigate through this shift successfully while at the same time helping them deliver services faster, cheaper and better. Navigate the digital shift successfully through our Digital Service model.

There is a major digital shift underway driven by technology changes (Cloud, Automation, Artificial Intelligence, Analytics and Serverless), and this is coupled with changing business and user expectations (speed, agility and changing business models).

Digital Transformation

Digital transformation has become an integral part of the way any business operates. And projects that seed and accelerate that journey are increasingly important. What is needed is a partner equipped with the digital skills, data and analytics expertise and a strategic mindset. Your partner must be cognizant of the employee pushback and the resulting cultural changes by instilling a data-based decision-making process, and bring all interests together on this journey.

Digital Transformation requires the right culture, attitude, skill and people. We believe we are in the right place to provide that to our clients.

Digital transformation projects are all the rage. Unsurprising given the number of benefits that accrue from embarking on this journey. They include – improved customer experience (through the introduction of modern and digital touchpoints and rich UI/UX for a consumer grade experience and personalized interactions), deep insights (by using data marts and analytics to provide never before insights), accurate understanding of consumer personas (by incorporating data driven insights with advanced machine learning algorithms), create new products and services (innovate and create new

products or repurpose existing products), and most importantly, reduced costs (data based insights to refine processes very tightly resulting in elimination of waste and better optimization).

We also see Artificial Intelligence and Machine Learning getting embedded in business processes to improve efficiencies, and also processing the vast amount of data being thrown out by digital interactions. Robotic Process Automation (RPA) will continue to gain footing in the digitization process because there is always a need to automate traditional and repetitive tasks.

We see deeper investments made in user interfaces, well designed user experiences, conversational UI, AR and VR (Augmented and Virtual Reality) and simple mobile apps. The result is Greater user interaction.

And lastly continuous investments in privacy and security, not just due to regulatory requirements (GDPR and CCPA), but because it helps build the brand and create trust.

Application Services

Our highly skilled team with technology and customer domain knowledge transitions complex applications quickly, maintain critical application and provide holistic solution for incidents and problem tickets in Manufacturing, Healthcare, Financial Services and Education.

We measure engagement success not just through SLA compliance, but in terms of automation, digitization and cost savings yielded.

Enterprises are saddled by legacy systems which while core to their business need to be transformed. More so in today's world of "digital anytime anywhere" transactions and where consumerization of IT is essential to ease of doing business transactions.

At the other end of the spectrum, the pace of technology advancement and the evolving needs of the customer necessitates new development or major application enhancement. More so when technology enables business ex. Mobile Commerce, Data Analytics and Intelligence on customer likes, preferences and behaviour, Artificial Intelligence and Deep Learning.

eCommerce

The ongoing COVID-19 pandemic has changed the world and set a new normal in consumer expectations to buy products and services conveniently and more importantly, securely. More and

more businesses will need to go online. This pandemic has provided a unique new entry to many players both as an opportunity and as a threat for survival.

We are the right partner that can help businesses go online by helping them with their Minimum Viable Product (MVP), developing their eCommerce platform, managing its day-to-day operations and monitor the eCommerce platform from a business operations readiness perspective. In addition, we provide a pulse of the business metrics through data marts and analytics.

Product Engineering

Our Product Engineering practice comprises of a large talent pool of engineers and is equipped with providing services across the product life cycle. Our consultants excel at every phase of the embedded product development life cycle. Their expertise spans product ideation, hardware, firmware and middleware, application development, complemented by testing, validation, verification and product sustenance.

Our solutions encompass a whole gamut of cutting-edge tools & technologies, backed by the expertise of a multi-disciplinary system and software engineering team that helps drive technological innovation in the field of embedded systems.

The demands for Security & Safety, Energy & Environment, and the market demands in transport, communications, entertainment products require innovation and technical leadership. Nowadays embedded and real-time systems contain complex software. The complexity of embedded systems is increasing, and the amount and variety of software in the embedded products growing. This creates a big challenge for embedded and real-time software development processes and there is a need to develop separate metrics and benchmarks. Enabling embedded systems require new and challenging solutions like multi-physics devices, millions of interconnected nodes, very low power for autonomy, trusted and safe operations, and reliability.

Quality Assurance

Leading enterprises trust our Quality Assurance services for their products and mission critical applications for predictable and sustained quality. These services are offered to more than 100 enterprises across Healthcare, Manufacturing, Banking and Financial Services, eCommerce and High-Tech Engineering.

The team undertakes Functional testing, Devices testing, Test Automation, Performance testing and API and services testing. A powerful edge is our Managed Testing Center of Excellence (CoE).

Insourcing

GIC's (Global In-house Center) today is considered a strategic initiative that has the potential to deliver competitive advantage. The question is, how do you make that happen? How can your GIC create business value for your enterprise? And how do you access the best talent? There is also a greater need to innovate and stay ahead of the disruptions sweeping industries.

You need a partner who brings in-depth knowledge of the regulatory environment, local ecosystem, industry best practices and source right-fit people and technologies.

Supply Chain

Within a span of three months, we saw failure of global supply chains, a very short demand surge and a prolonged demand drop. All events happening in quick succession. Which begs the question "How do we build supply chains and networks that can withstand such changes?"

The new normal is here to stay and it demands two things from every supply chain – stay resilient and stay lean, apart from the changes in demand patterns. Thus learning to be adaptive and change with the new consumer demands will become the need of the hour.

The trends shaping supply chain transformation are the ability of a company and its ecosystem to address the 3 Vs – Volatility, Visibility and Value and the 3 Rs – Responsiveness, Resilience and Responsibility.

Demand volatility has increased multifold with disruptive changes to customer purchase and consumption patterns due to a variety of purchase channels and other social media influence and options. More than a single company, end-to-end supply chain ecosystems need to be tailored to the market needs and be responsive to be able to address this volatility faster. With multiple partners across the value chain, product visibility and status is critical to manage both customer and value chain partner expectations and decisions. For the supply chain ecosystem to function smoothly value needs to be created and realized appropriately for alignment of goals and for smoother functioning of the supply chain.

Re-invention of supply chain strategy and transformation are sought by companies to align to the changing market demands and have taken multiple overlapping forms, a few of which are listed below:

- Customer centric supply chain – addressing volatility
- Transparent supply chain – addressing visibility and value transactions
- Strategic supply and network management – managing value and cost
- Digital supply chain with advanced digital technologies – Managing risk and resilience and quicker value generation and realization

Insights from advanced analytics on customer preferences and behaviors drive better demand sensing and planning capabilities supported by end-to-end supply chain alignment towards customer value is the focus in customer centric supply chains. Transparent supply chains support the ability to maintain enterprise and ecosystem wide visibility.

Strategic sourcing and supply management and an agile and optimized network enables value creation and realization within the supply chain ecosystem. Data integration and advanced analytics enable better preparedness for the supply chain to manage risk.

Traceability

Traceability of products through its fulfilment supply chain, followed by manufacturing and all the way back into the raw material supply chain is still a very important aspect of modern supply chain systems. Being able to provide the proverbial farm-to-fork or factory-to-shelf capability is important to the modern consumer and retail.

Agility

Ability to ramp up production during market demand and ability to scale it down when there is a slump are two key patterns that are fundamental to the supply chain systems. Measuring the total cost of such ramp up / ramp down through simulation using digital twins for various actors in the supply chain would be a great feature to battle test our planning processes to see if it holds up to actual situations.

Anti-fragility

Anti-fragility is now a norm for supply chain systems. While it is not a directly measurable and verifiable attribute, we believe in a combination of collaboration and transparency across the network through open API on the cloud and open data platforms. Data and information exchange should be powered by analytics driven insights. So a robust analytics platform to look at data across the entire network is necessary.

Assisted by automation

Data and insights are great but the warehouses of yesterday where human labor ruled supreme should now be augmented by autonomous machine assisted labor force. Robots in warehouses working in an automated fashion assisting the human labor force in improving efficiencies, tracking warehouse inventory and providing real-time data from warehouse floor through computer vision algorithms are some of the key features of a modern supply chain system.

Intelligence

Data gathered from partners, suppliers, shop and factory floors that are aggregated into analytics and insights is great, but this still requires human intervention even for simple data based decisions. The advent of machine learning (ML) in the supply chain hopes to alleviate the need for human decision. ML algorithms to bring in high accuracy require high quality training data and large computation power to be effective. This has been the roadblock to better adoption in the supply chain. By bringing in high quality training data we can increase the applicability of ML in the supply chain.

The applicability of intelligence is not just within the organization. By extending this intelligence to be made available to partners, we can help partners take advantage and participate in a collaborative network planning. To enable partners to take advantage of the intelligence a simple chatbot style interface can be enabled so that partners can ask freeform questions that can be responded to by the ML engine via the chatbot interface.

Sustainability

Companies that have adopted environmentally responsible supply chain strategies have incorporated fair labor practices and environmental responsibility throughout their supply networks. They have started developing and disseminating industry wide sustainability standards. Few have joined hands to start industry associations (for example Responsible Business Alliance (RBA), whose members include Intel, HP, IBM, Dell, Philips, and Apple) and have started offering sustainability training to Tier-1 and Tier-2 suppliers.

1.3 EXISTING SYSTEM

The current system operates manual supermarket management system, from stocks, products, ordering and purchases etc. recorded in a book. This is faced with errors, incompleteness, and insufficient data for analysis. Information regarding stocks, products, sales and purchases are still in black and white which is not properly organized and managed. From the wholesalers to retailer bills, tickets, vouchers, receipts of products are recorded in a book but further operations are not being properly handled. As a result, it is difficult in processing, updating and managing.

The factors for these difficulties are:

Labor-Intensive: A manual Super Market management systems is that they can be highly labor-intensive to operate. They require continuous monitoring to ensure that each transaction is accounted for and that products are maintained at the appropriate stocking levels. It is also more difficult to share inventory information throughout the business, because the lack of computerization makes accessing inventory records a more cumbersome process. The time spent monitoring inventory levels could be used on more productive activities for the business.

Human Error: A manual Supermarket management system relies heavily on the actions of people, which increases the possibility of human error. People might forget to record a transaction or simply miscount the number of goods. This results in needless additional orders that increase the company's inventory carrying costs and use up precious storage space. Inaccurate physical counts could also result in not ordering enough of a product, meaning the business could run out of a crucial item at the wrong time.

Time Wasting: A manual Supermarket management system has a huge tendency of time wasting as the sales manager could have a lot to tackle while many customers seek attention and this is really affecting the business.

1.4 PROPOSED SYSTEM

In order to enhance the existing system various modules have been developed for handling

online web application functionalities such as Login to the application using secure credentials, Users are able to enter or upload data pertaining to income inventories, Rule and Category management, so that the system generates needed alerts. Ex if goods is reaching low in inventory or getting damaged etc., On Daily / Regular basis when goods move from warehouse to the shop, the system has to update the inventory details accordingly.

There will be a maker-checker role within the shop, where inventory uploaded / entered by a maker, would be verified by the checker. Once approved by checker, the data should not get tampered with the details about the incoming inventories come as file feed to the system.

It also handles challenging tasks like what is the stock available in the inventory (i.e., in warehouse), and available in the shelf, the edible stocks that does not move too fast, are tend of get spoiled too frequently. The system should give warning 2-3 days in advance when that situation arises, to categories & identify the damaged goods, whether during transport, during shopping Or by expiry dates, Invoice to the suppliers to be managed for every stock that moves in and reverse invoices for the goods that moves out, Alert should be made to the administrator, when some stocks go below the threshold

2. SYSTEM ANALYSIS

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. These are the requirements for doing the project. Without using these tools and software's we can't do the project. So, we have two requirements to do the project.

They are 1. Hardware Requirements.

2. Software Requirements.

2.1 REQUIREMENT SPECIFICATION

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. The software requirements provide a basis for creating the software requirements specification.

2.1.1 HARDWARE REQUIREMENT

- **Ram:** 256 MB or more
- **Processor:** Intel
- **CPU Speed:** 700 Mb or more

2.1.2 SOFTWARE REQUIREMENTS

- **Operating System:** Windows 10
- **Database Server:** Spring Boot and JSON
- **Client Server:** Microsoft Internet Explorer
- **User Interface:** HTML, CSS, Angular, Java Script
- **Tools:** Microsoft Visual Studio Code

2.1.3 SOFTWARE DESCRIPTION

HTML:

Formatting by specifying the relevant CSS in a separate CSS file, and reduce complexity and repetition in the structural content.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable. The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/CSS is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents.

In addition to HTML, other markup languages support the use of CSS, including XHTML, plain XML, SVG, and XUL.

ANGULAR:

Angular is an application design framework and development platform for creating efficient and sophisticated single-page apps.

These Angular docs help you learn and use the Angular framework and development platform, from your first application to optimizing complex single-page apps for enterprises. Tutorials and guides include downloadable examples to accelerate your projects.

Angular can manage the dependency structure among your components and injectable services to optimize bundle size by using tree-shakeable-providers. This normally ensures that if a provided component or service is never actually used by the app, the compiler can eliminate its code from the bundle. However, due to the way Angular stores injection tokens, it is possible that such an unused component or service can end up in the bundle anyway. This page describes a

dependency-injection design pattern that supports proper tree-shaking by using lightweight injection tokens.

The lightweight injection token design pattern is especially important for library developers. It ensures that when an application uses only some of your library's capabilities, the unused code can be eliminated from the client's application bundle.

When an application uses your library, there might be some services that your library supplies which the client application doesn't use. In this case, the application developer should expect that service to be tree-shaken, and not contribute to the size of the compiled application. Because the application developer cannot know about or remedy a tree-shaking problem in the library, it is the responsibility of the library developer to do so. To prevent the retention of unused components, your library should use the lightweight injection token design pattern.

Components of Angular: Components are the main building block for Angular applications. Each component consists of:

- An HTML template that declares what renders on the page
- A Typescript class that defines behaviour
- A CSS selector that defines how the component is used in a template
- Optionally, CSS styles applied to the template

CSS in Angular: Angular applications are styled with standard CSS. That means you can apply everything you know about CSS stylesheets, selectors, rules, and media queries directly to Angular applications. Additionally, Angular can bundle *component styles* with components, enabling a more modular design than regular stylesheets.

Easy dynamic content in an Angular app: Transforming a component to a custom element provides an easy path to creating dynamic HTML content in your Angular app. HTML content that you add directly to the DOM in an Angular app is normally displayed without Angular processing, unless you define a *dynamic component*, adding your own code to connect the HTML tag to your app data, and participate in change detection. With a custom element, all of that wiring is taken care of automatically.

Content-rich applications: If you have a content-rich app, such as the Angular app that presents this documentation, custom elements let you give your content providers sophisticated Angular functionality without requiring knowledge of Angular. For example, an Angular guide like this one is added directly to the DOM by the Angular navigation tools, but can include special elements like <code-snippet> that perform complex operations. All you need to tell your content provider is the syntax of your custom element. They don't need to know anything about Angular, or anything about your component's data structures or implementation.

Security: Angular's built-in protections against common web-application vulnerabilities and attacks such as cross-site scripting attacks. It doesn't cover application-level security, such as authentication and authorization.

Accessibility: The web is used by a wide variety of people, including those who have visual or motor impairments. A variety of assistive technologies are available that make it much easier for these groups to interact with web-based software applications. In addition, designing an application to be more accessible generally improves the user experience for all users.

SPRING BOOT

Spring is widely used for creating scalable applications. For web applications Spring provides

Spring MVC which is a widely used module of spring which is used to create scalable web applications.

But main **disadvantage of spring projects is that configuration is really time-consuming and can be a bit overwhelming for the new developers.** Making the application production-ready takes some time if you are new to the spring.

Solution to this is Spring Boot. Spring Boot is built on the top of the spring and contains all the features of spring. And is becoming favourite of developer's these days because of it's a rapid production-ready environment which enables the developers to directly focus on the logic instead of struggling with the configuration and set up.

Spring Boot is a microservice-based framework and making a production-ready application in it takes very less time.

Prerequisite for Spring Boot is the basic knowledge Spring framework.

For revising the concepts of spring framework.

Features of Spring Boot

Spring Boot is built on the top of the conventional spring framework. So, it provides all the features of spring and is yet easier to use than spring.

- **It allows to avoid heavy configuration of XML which is present in spring:**

Unlike the Spring MVC Project, in spring boot everything is auto-configured. We just need to use proper configuration for utilizing a particular functionality.

- **It provides easy maintenance and creation of REST end points:**

Creating a REST API is very easy in Spring Boot. Just the annotation **@RestController** and **@RequestMapping(/endPoint)** over the controller class does the work.

- **It includes embedded Tomcat-server:**

Unlike Spring MVC project where we have to manually add and install the tomcat server, Spring Boot comes with an embedded Tomcat server, so that the applications can be hosted on it.

- **Deployment is very easy, war and jar file can be easily deployed in the tomcat server:**

war or **jar** files can be directly deployed on the Tomcat Server and Spring Boot provides the facility to convert our project into war or jar files. Also, the instance of Tomcat can be run on the cloud as well.

- **Microservice Based Architecture:**

Microservice, as the name suggests is the name given to a module/service which focuses on a single type of feature, exposing an API(application peripheral interface).

Let us consider an example of a hospital management system.

- In case of monolithic systems, there will be a single code containing all the features which are very tough to maintain on a huge scale.
- But in the microservice-based system, each feature can be divided into smaller subsystems like service to handle patient registration, service to handle database management, service to handle billing etc.

Microservice based system can be easily migrated as only some services need to be altered which also makes debugging and deployment easy. Also, each service can be integrated and can be made in different technologies suited to them.

Spring Boot Architecture

To understand the architecture of Spring Boot, let us first see different layers and classes present in it.

- **Layers in Spring Boot:** There are four main layers in Spring Boot:
- **Presentation Layer:** As the name suggests, it consists of views(i.e. frontend part)
- **Data Access Layer:** CRUD (create, retrieve, update, delete) operations on the database comes under this category.
- **Service Layer:** This consist of service classes and uses services provided by data access layers.
- **Integration Layer:** It consists of web different web services(any service available over the internet and uses XML messaging system).
- Then we have utility classes, validator classes and view classes.
- All the services provided by the classes are implemented in their corresponding classes and are retrieved by implementing the dependency on those interfaces.

Setup Spring Boot:

1. Setup Java JDK from Oracle's official site.
 2. Download and Setup STS(Spring Tools Suite).
 3. Start a new spring starter project
- Click on File -> New -> Spring starter project
 - Fill the appropriate details and add dependency and finish.
 - Edit the application properties.
 - Run the main file as a Java application.

JAVASCRIPT:

JavaScript is a high-level, interpreted programming language. It is a language which is also characterized as dynamic, weakly typed, prototype- based and multi- paradigm. Alongside HTML and CSS, JavaScript is one of the three core technologies of the World Wide Web. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it, [8] and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event- driven, functional,

and imperative (including object-oriented and prototype-based) programming styles. It has an API for working with text, arrays, dates, regular expressions, and basic manipulation of the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

Characteristics

Let's take a step back and count the merits of JavaScript:

JavaScript is very easy to implement. All you need to do is put your code in the HTML document and tell the browser that it is JavaScript.

JavaScript works on web users' computers — even when they are offline!

- JavaScript allows you to create highly responsive interfaces that improve the user experience and provide dynamic functionality, without having to wait for the server to react and show another page.

JavaScript can help fix browser problems or patch holes in browser support — for example fixing CSS layout issues in certain browsers.

That is a lot for a language that until recently was laughed at by programmers favoring “higher programming languages”. One part of the renaissance of JavaScript is that we are building more and more complex web applications these days, and high interactivity either requires Flash (or other plugins) or scripting. JavaScript is arguably the best way to go, as it is a web standard, it is supported natively across browsers (more or less — some things differ across browsers, and these differences are discussed in appropriate places in the articles that follow this one), and it is compatible with other open web standards.

Common uses of JavaScript

The usage of JavaScript has changed over the years we have been using it. At first, JavaScript interaction with the site was mostly limited to interacting with forms, giving feedback to the user, and detecting when they do certain things. We used alert () to notify the user of something, confirm () to ask if something is OK to do and either prompt () or a form field to get user input. This led mostly to validation scripts that stopped the user to send a form to the server when there was a mistake, and simple converters and calculators. In addition, we managed to build highly useless things like prompts asking the user for their name just to print it out immediately afterwards. Another thing we used was document. Write () to add content to the document. We also worked with pop-up windows and frames and lost many a nerve and pulled out hair trying to make them talk to each other. Thinking about most of these technologies should make any developer rock forward and backward and curl up into a fetal position stammering “make them go away”, so let’s not dwell on these things — there are better ways to use JavaScript.

3. SYSTEM DESIGN

3.1 CLASS DIAGRAM

ADMIN:

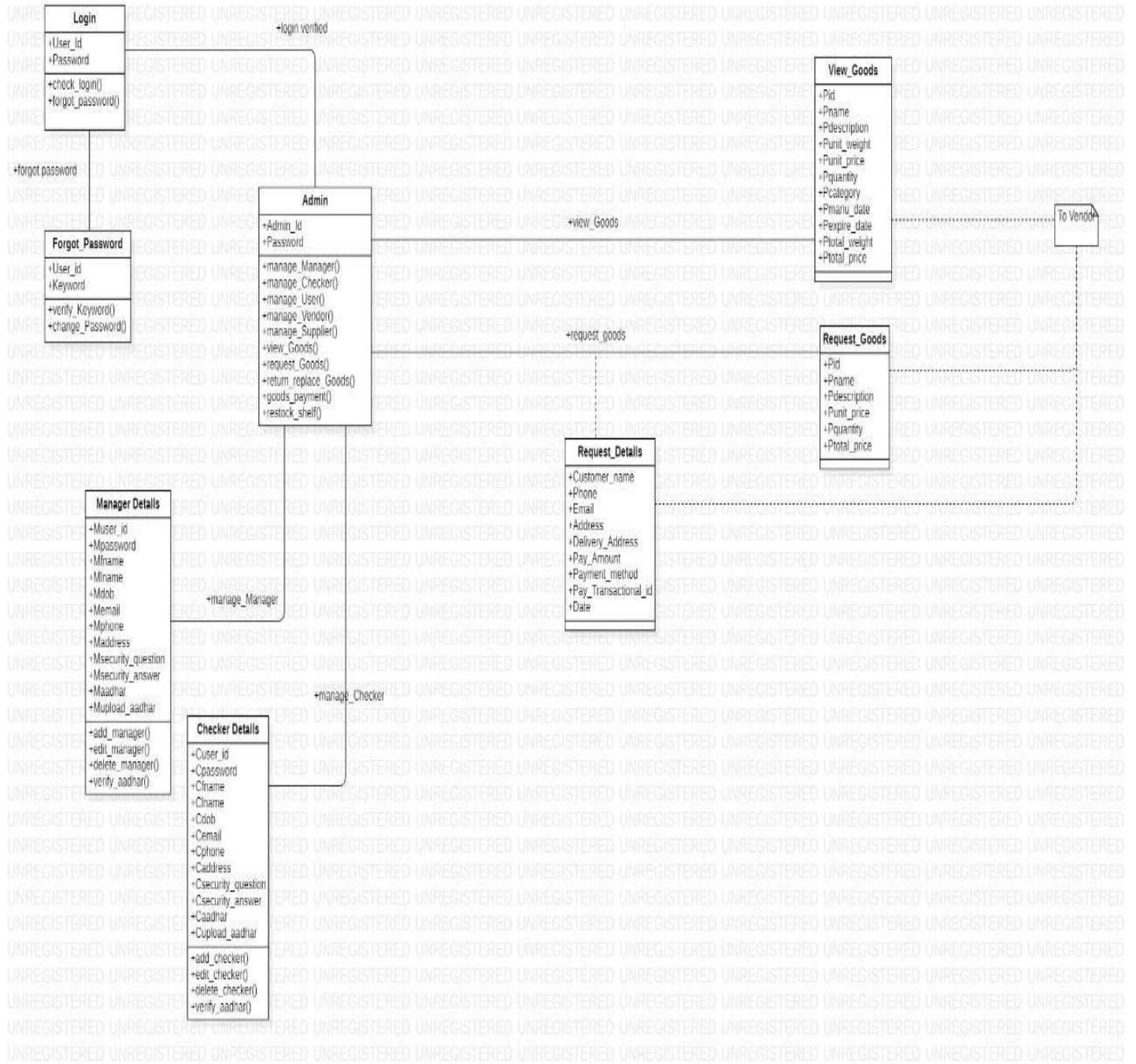


Figure 1: Admin - Use Case Diagram

VENDOR DIAGRAM:

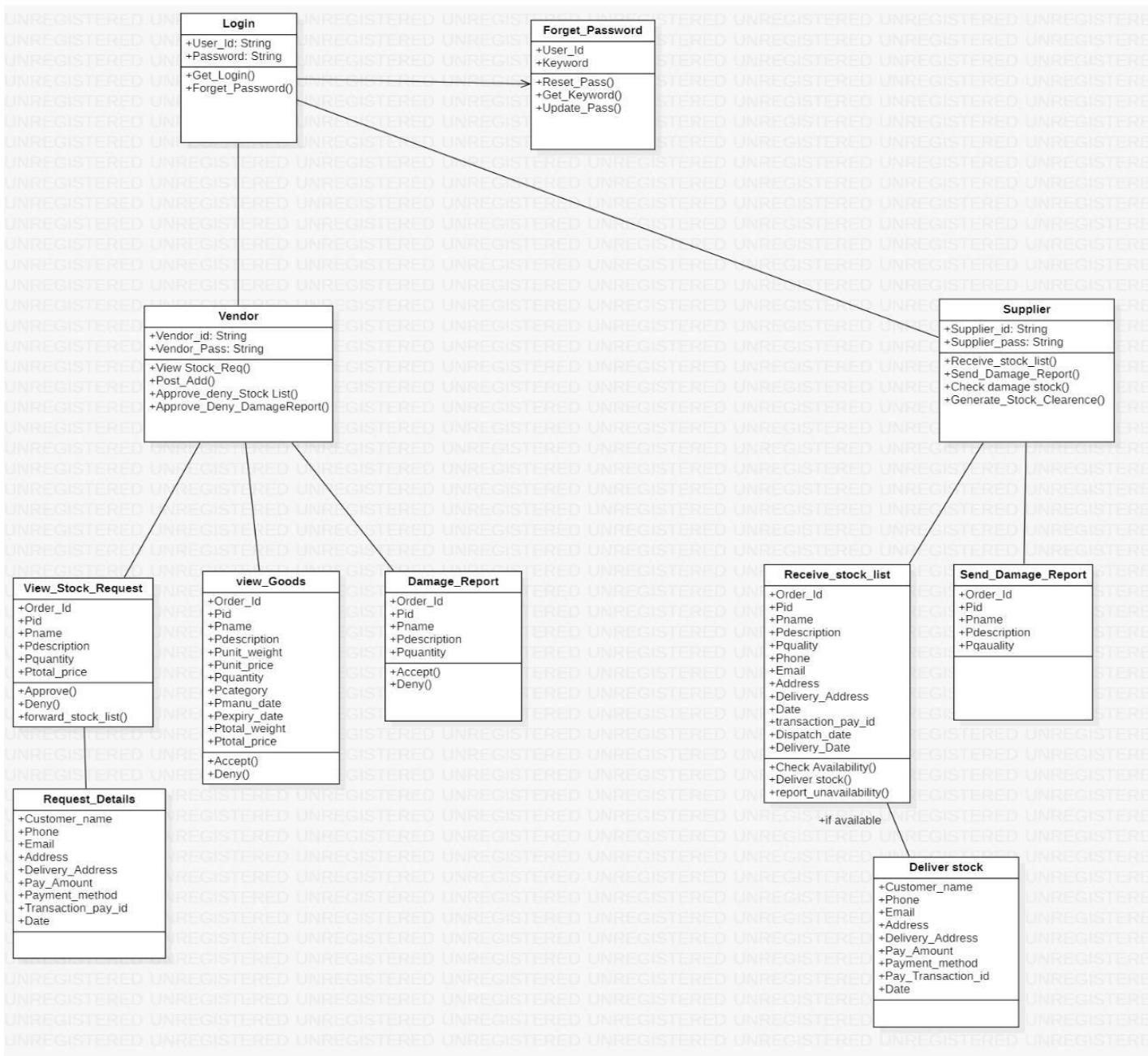


Figure 2: Vendor- Use Case Diagram

WAREHOUSE DIAGRAM

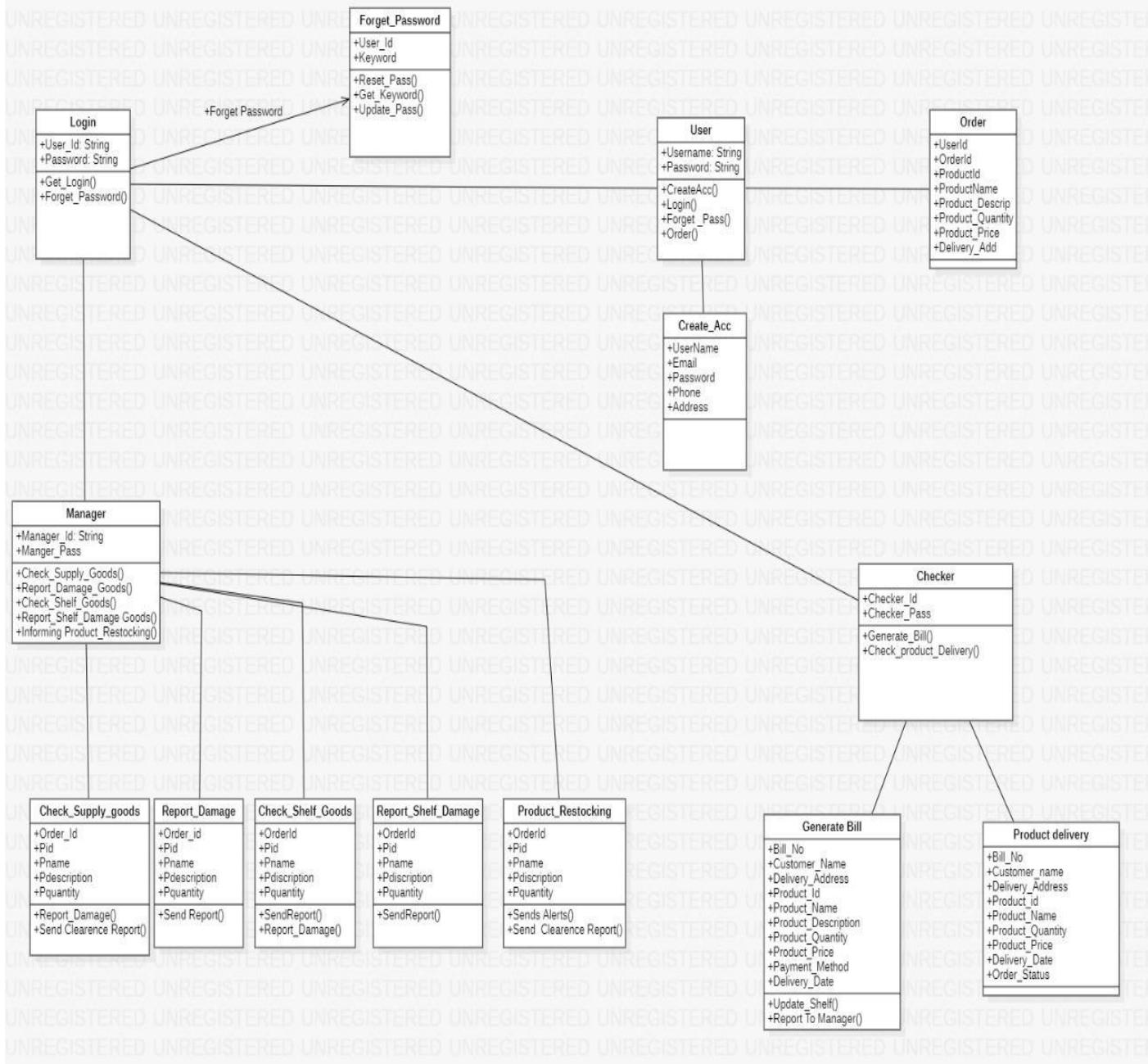


Figure 3: Warehouse- Use Case Diagram

3.2 DATA DICTIONARY

A data dictionary is a file or set of files that contains a database's metadata. The data dictionary contains records about objects in the database, such as data ownership, data relationships to other objects, and other data. Data dictionary consists of the following information:

- Name of the tables in the database
- Constraints of a table (keys, relationship, etc.)
- Columns of the tables that related to each other

DATABASE NAME: inventory

3.2.1 admin_request

COLUMN	DATATYPE	DESCRIPTION
Product_Type	Any	Type of the product
ProductName	Any	Product name
VendorId	Any	Vendor id
Vendorname	Any	Vendor name
SupplierName	Any	Supplier Name
ProductDescription	Any	Product Description
ProductQty	Any	Product Quantity

3.2.2 user_profile

COLUMN	DATATYPE	DESCRIPTION
FirstName	Any	Product id
LastName	Any	Type of the product
Designation	any	Product name
Department	any	Vendor id

DOB	any	Vendor name
Gender	any	Supplier Name
PanId	any	Product Description
AadharId	any	Product Quantity
UserName	any	Name of the user
Id	any	Id of the user
Password	any	User Password
MailId	any	Mail id of the user
Phone	any	Mobile number of the user

3.2.3 Shop_stock

COLUMN	DATATYPE	DESCRIPTION
OrderId	any	Order id
ProductId	any	Product id details
ProductType	any	Type of product available in shop
ProductName	any	Name of the product
InDate	any	Stock arrival date
OutDate	any	Stock sold date
VendorId	any	Vendor id
ProductQty	any	Quantity of the product

\

3.2.4 Perishablegoods

COLUMN	DATATYPE	DESCRIPTION
ProductId	any	Product id details
ProductType	any	Type of product available in shop
ProductName	any	Name of the product
InDate	any	Stock arrival date
OutDate	any	Stock sold date
VendorId	any	Vendor id
ProductQty	any	Quantity of the product

3.2.5 Warehouse_stock

COLUMN	DATATYPE	DESCRIPTION
OrderId	any	Order id
ProductId	any	Product id details
ProductType	any	Type of product available in shop
ProductName	any	Name of the product
InDate	any	Stock arrival date
OutDate	any	Stock sold date
VendorId	any	Vendor id
ProductQty	any	Quantity of the product

3.2.6 Damage_stock

COLUMN	DATATYPE	DESCRIPTION
ProductType	any	Type of product available in shop
ProductName	any	Name of the product
VendorId	any	Vendor id
Vendor_name	any	Name of the vendor
SupplierName	any	Name of the Supplier
ProductDescription	any	Detailed description of the product
ProductQty	any	Quantity of the product
Damage	any	Damaged goods
DamageQty	any	No of items Damaged

3.2.7 Damage_stock

COLUMN	DATATYPE	DESCRIPTION
ProductType	any	Type of product available in shop
ProductName	any	Name of the product
VendorId	any	Vendor id
Vendor_name	any	Name of the vendor
SupplierName	any	Name of the Supplier
ProductDescription	any	Detailed description of the product
ProductQty	any	Quantity of the product

Price	any	Price of product
Discount	any	Discount of each product

3.3 TABLE DESCRIPTION

Table Name: Requestgoods

```
[  
  {  
    "id": 1,  
    "ProductType": "Beverages",  
    "ProductName": "Beer",  
    "VendorId": "SV10287",  
    "VendorName": "Latta",  
    "SupplierName": "Lokesh",  
    "ProductDiscription": "Non-Perishable",  
    "ProductQty": "10"  
  },  
  {  
    "id": 2,  
    "ProductType": "Beverages",  
    "ProductName": "Pepsi",  
    "VendorId": "SV10287",  
    "VendorName": "Latta",  
    "SupplierName": "Lokesh",  
    "ProductDiscription": "Non-Perishable",  
    "ProductQty": "20"  
  },  
  {  
    "id": 3,  
    "ProductType": "Beverages",  
    "ProductName": "Coke",  
    "VendorId": "SV10287",  
    "VendorName": "Latta",  
    "SupplierName": "Lokesh",  
    "ProductDiscription": "Non-Perishable",  
    "ProductQty": "30"  
  },  
  {  
    "id": 4,  
    "ProductType": "Dairy",  
    "ProductName": "Bread",  
    "VendorId": "SV10268",  
    "VendorName": "Ram",  
    "SupplierName": "Sham",  
    "ProductDiscription": "Perishable",  
    "ProductQty": "20"  
  },  
  {  
    "id": 5,  
    "ProductType": "Dairy",  
    "ProductName": "Milk",  
    "VendorId": "SV10268",  
    "VendorName": "Ram",  
    "SupplierName": "Sham",  
    "ProductDiscription": "Perishable",  
    "ProductQty": "25"  
},  
,
```

Table Name: perishablegoods

```
[  
  {  
    "ProductId": "FA10234",  
    "ProductType": "Fruits",  
    "ProductName": "Kiwi",  
    "InDate": "12/3/20",  
    "OutDate": "16/3/20",  
    "VendorId": "SV10287",  
    "ProductQty": "10Kg"  
  },  
  {  
    "ProductId": "FA36897",  
    "ProductType": "Fruits",  
    "ProductName": "Mango",  
    "InDate": "18/3/20",  
    "OutDate": "12/3/20",  
    "VendorId": "SV12345",  
    "ProductQty": "5kg"  
  },  
  {  
    "ProductId": "FA36987",  
    "ProductType": "Fruits",  
    "ProductName": "WaterMelon",  
    "InDate": "16/8/20",  
    "OutDate": "20/4/20",  
    "VendorId": "SV12356",  
    "ProductQty": "15kg"  
  },  
  {  
    "ProductId": "FA36856",  
    "ProductType": "Fruits",  
    "ProductName": "Orange",  
    "InDate": "11/4/20",  
    "OutDate": "15/4/20",  
    "VendorId": "SV12369",  
    "ProductQty": "50kg"  
  },  
  {  
    "ProductId": "VE36897",  
    "ProductType": "Vegetables",  
    "ProductName": "Brocoli",  
    "InDate": "04/4/20",  
    "OutDate": "09/4/20",  
    "VendorId": "SV35321",  
    "ProductQty": "5kg"  
  },  
  {  
    "ProductId": "VE96325",  
    "ProductType": "Vegetables",  
    "ProductName": "Potato",  
    "InDate": "30/3/20",  
    "OutDate": "03/4/20",  
    "VendorId": "SV12856",  
    "ProductQty": "9kg"  
  },
```

Table Name: Shopstock

```
[  
  {  
    "OrderId": "Lan963214",  
    "ProductId": "FA10692",  
    "ProductType": "Fruits",  
    "ProductName": "Orange",  
    "InDate": "09/6/20",  
    "OutDate": "10/6/20",  
    "VendorId": "SV10963",  
    "ProductQty": "30kg"  
  },  
  {  
    "OrderId": "Lan158742",  
    "ProductId": "FG10234",  
    "ProductType": "Foodgrains",  
    "ProductName": "Wheat",  
    "InDate": "30/5/20",  
    "OutDate": "05/6/20",  
    "VendorId": "SV10254",  
    "ProductQty": "30Kg"  
  },  
  {  
    "OrderId": "Lan951234",  
    "ProductId": "DA10234",  
    "ProductType": "Dairy",  
    "ProductName": "Bread",  
    "InDate": "21/5/20",  
    "OutDate": "22/5/20",  
    "VendorId": "SV10285",  
    "ProductQty": "10"  
  },  
  {  
    "OrderId": "Lan753159",  
    "ProductId": "CL10287",  
    "ProductType": "Cleaning",  
    "ProductName": "Handwash",  
    "InDate": "18/5/20",  
    "OutDate": "20/5/20",  
    "VendorId": "SV10225",  
    "ProductQty": "25"  
  },  
  {  
    "OrderId": "Lan456321",  
    "ProductId": "BP10234",  
    "ProductType": "Baby Products",  
    "ProductName": "Diapers",  
    "InDate": "10/5/20",  
    "OutDate": "12/5/20",  
    "VendorId": "SV10963",  
    "ProductQty": "40"  
  },  
]
```

Table Name: warehouse

```
[  
  {  
    "OrderId": "Lan321456",  
    "ProductId": "FA10234",  
    "ProductType": "Fruits",  
    "ProductName": "Kiwi",  
    "InDate": "12/3/20",  
    "OutDate": "16/3/20",  
    "VendorId": "SV10287",  
    "ProductQty": "10Kg"  
  },  
  {  
    "OrderId": "Lan753357",  
    "ProductId": "VE10234",  
    "ProductType": "Vegetables",  
    "ProductName": "Tomato",  
    "InDate": "15/3/20",  
    "OutDate": "20/3/20",  
    "VendorId": "SV10298",  
    "ProductQty": "15Kg"  
  },  
  {  
    "OrderId": "Lan753357",  
    "ProductId": "VE10234",  
    "ProductType": "Vegetables",  
    "ProductName": "Tomato",  
    "InDate": "15/3/20",  
    "OutDate": "20/3/20",  
    "VendorId": "SV10298",  
    "ProductQty": "15Kg"  
  },  
  {  
    "OrderId": "Lan954876",  
    "ProductId": "BA102742",  
    "ProductType": "Beverages",  
    "ProductName": "Coke",  
    "InDate": "21/3/20",  
    "OutDate": "25/3/20",  
    "VendorId": "SV10295",  
    "ProductQty": "20"  
  },  
  {  
    "OrderId": "Lan951247",  
    "ProductId": "DA10234",  
    "ProductType": "Dairy",  
    "ProductName": "Milk",  
    "InDate": "25/3/20",  
    "OutDate": "30/3/20",  
    "VendorId": "SV10265",  
    "ProductQty": "15"  
  },  
]
```

Table Name: Damage goods

```
[  
 {  
   "id": 1,  
   "ProductType": "Beverages",  
   "ProductName": "Beer",  
   "VendorId": "SV10287",  
   "VendorName": "Latta",  
   "SupplierName": "Lokesh",  
   "ProductDiscription": "Non-Perishable",  
   "ProductQty": "10",  
   "Damage": "Yes",  
   "DamageQty": "5"  
 },  
 {  
   "id": 2,  
   "ProductType": "Beverages",  
   "ProductName": "Pepsi",  
   "VendorId": "SV10287",  
   "VendorName": "Latta",  
   "SupplierName": "Lokesh",  
   "ProductDiscription": "Non-Perishable",  
   "ProductQty": "20",  
   "Damage": "Yes",  
   "DamageQty": "10"  
 },  
 {  
   "id": 3,  
   "ProductType": "Beverages",  
   "ProductName": "Coke",  
   "VendorId": "SV10287",  
   "VendorName": "Latta",  
   "SupplierName": "Lokesh",  
   "ProductDiscription": "Non-Perishable",  
   "ProductQty": "30",  
   "Damage": "No",  
   "DamageQty": "0"  
 },  
 {  
   "id": 4,  
   "ProductType": "Dairy",  
   "ProductName": "Bread",  
   "VendorId": "SV10268",  
   "VendorName": "Ram",  
   "SupplierName": "Sham",  
   "ProductDiscription": "Perishable",  
   "ProductQty": "20",  
   "Damage": "Yes",  
   "DamageQty": "15"  
 },  
 {  
   "id": 5,  
   "ProductType": "Dairy",  
   "ProductName": "Milk",  
   "VendorId": "SV10268",  
   "VendorName": "Ram",  
   "SupplierName": "Sham",  
   "ProductDiscription": "Perishable",  
   "ProductQty": "20",  
   "Damage": "Yes",  
   "DamageQty": "15"  
 }]
```

Table Name: Approved stock

```
[  
  {  
    "id": 1,  
    "ProductType": "Beverages",  
    "ProductName": "Beer",  
    "VendorId": "SV10287",  
    "VendorName": "Latta",  
    "SupplierName": "Lokesh",  
    "ProductDiscription": "Non-Perishable",  
    "ProductQty": "10",  
    "Price": "360",  
    "Discount": "20%"  
  },  
  {  
    "id": 2,  
    "ProductType": "Beverages",  
    "ProductName": "Pepsi",  
    "VendorId": "SV10287",  
    "VendorName": "Latta",  
    "SupplierName": "Lokesh",  
    "ProductDiscription": "Non-Perishable",  
    "ProductQty": "20",  
    "Price": "300",  
    "Discount": "50%"  
  },  
  {  
    "id": 3,  
    "ProductType": "Beverages",  
    "ProductName": "Coke",  
    "VendorId": "SV10287",  
    "VendorName": "Latta",  
    "SupplierName": "Lokesh",  
    "ProductDiscription": "Non-Perishable",  
    "ProductQty": "30",  
    "Price": "250",  
    "Discount": "250%"  
  },  
  {  
    "id": 4,  
    "ProductType": "Dairy",  
    "ProductName": "Bread",  
    "VendorId": "SV10268",  
    "VendorName": "Ram",  
    "SupplierName": "Sham",  
    "ProductDiscription": "Perishable",  
    "ProductQty": "20",  
    "Price": "250",  
    "Discount": "20%"  
  },  
  {  
    "id": 5,  
    "ProductType": "Dairy",  
    "ProductName": "Milk",  
    "VendorId": "SV10268",  
    "VendorName": "Ram",  
    "SupplierName": "Sham",  
  }]
```

Table Name: User profile

```
[  
 {  
   "FirstName": "Srinath",  
   "LastName": "narayan",  
   "Designation": "Manager",  
   "Department": "Warehouse",  
   "DOB": "12/12/1998",  
   "Gender": "Male",  
   "PanId": "WEA123456",  
   "AadharId": "3214 6542 6544",  
   "UserName": "shri",  
   "Id": "MA12587469",  
   "Password": "Qwerty@123",  
   "MailId": "waremanager@gamil.com",  
   "Phone": "9632587412"  
 },  
 {  
   "FirstName": "Kishore",  
   "LastName": "Kumar",  
   "Designation": "Manager",  
   "Department": "Shop",  
   "DOB": "22/105/1998",  
   "Gender": "Male",  
   "PanId": "JKA253456",  
   "AadharId": "3698 2587 1473",  
   "UserName": "Kish",  
   "Id": "MA1236547892",  
   "Passward": "Chennai@123",  
   "MailId": "shopmanager@gamil.com",  
   "Phone": "9512364785"  
 },  
 {  
   "FirstName": "Latta",  
   "LastName": "Naidu",  
   "Designation": "Suppiler",  
   "Department": "Shop",  
   "DOB": "14/08/1998",  
   "Gender": "FeMale",  
   "PanId": "MJA135789",  
   "AadharId": "3547 2587 1478",  
   "UserName": "latta",  
   "Id": "SP123654789",  
   "Password": "Zxy@1254",  
   "MailId": "suppiler@gamil.com",  
   "Phone": "9965488745"  
 },  
 {  
   "FirstName": "Summi",  
   "LastName": "shree",  
   "Designation": "Vendor",  
   "Department": "Shop",  
   "DOB": "22/5/1998",  
   "Gender": "FeMale",  
   "PanId": "POA1258256",  
   "AadharId": "2221 5872 3578",  
   "UserName": "Dog",  
   ".....  
 ]
```

4.1 OVERALL ARCHITECTURE

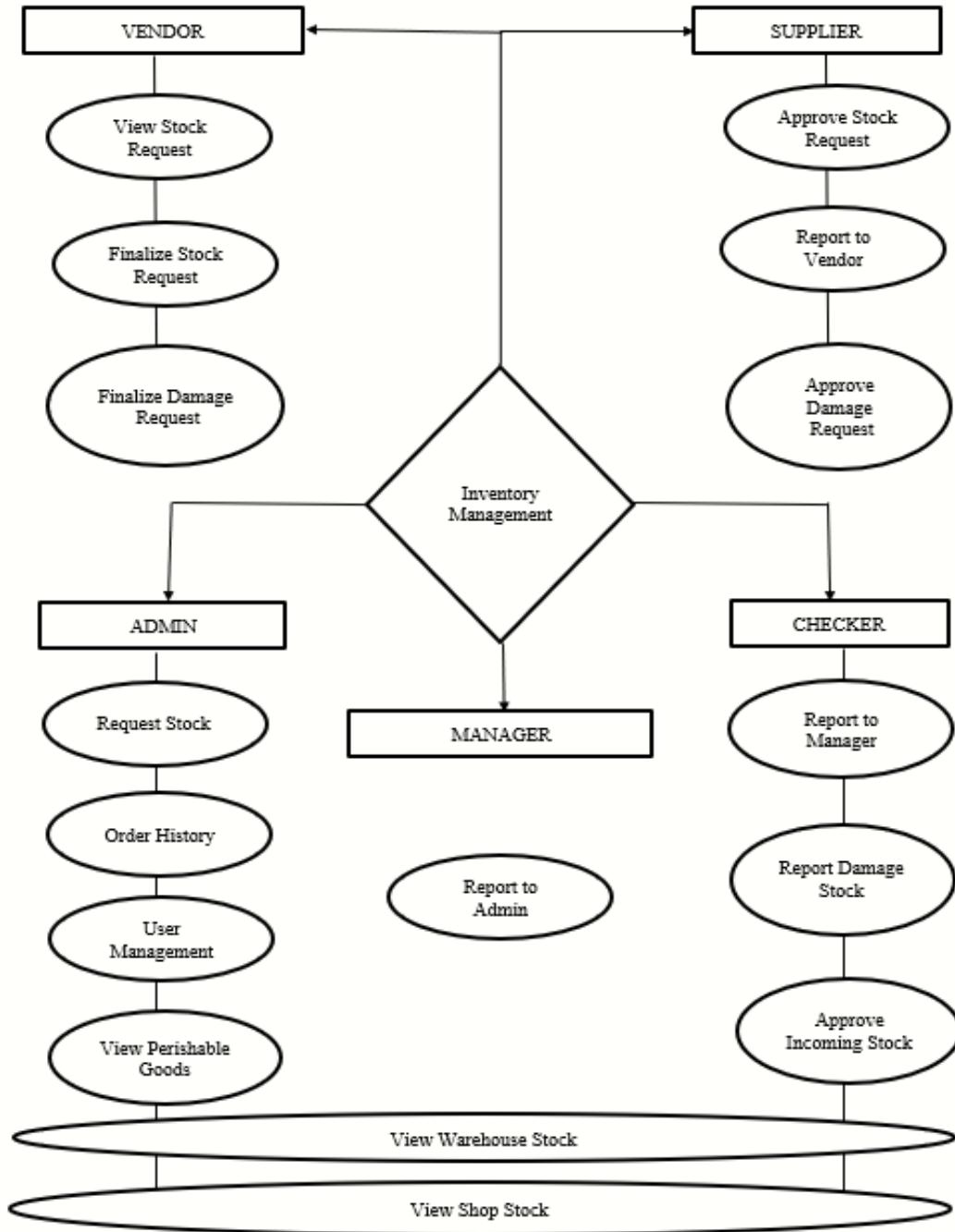


Figure 4: Overall Architecture

5. MODULE DESCRIPTION

5.1 HOME MODULE

The **Homepage** module lets users choose a specific page as their homepage. Users with a homepage will be redirected to this page upon successful login on the site. The module provides three main blocks. One of them allows users to login as **Admin**, second as **The Contact module** allows **site** visitors to send emails to other authenticated users and to the **site** administrator. **Vendor** and other as **Warehouse** person. On click of the specific icon, user is redirected login page and then to their respective page after login in. The **Contact module** allows **site** visitors to send emails to other authenticated users and to the **site** administrator. The features module enables the capture and management of features in Inventory Management.

INVENTORY MANGEMENT HOME PAGE:

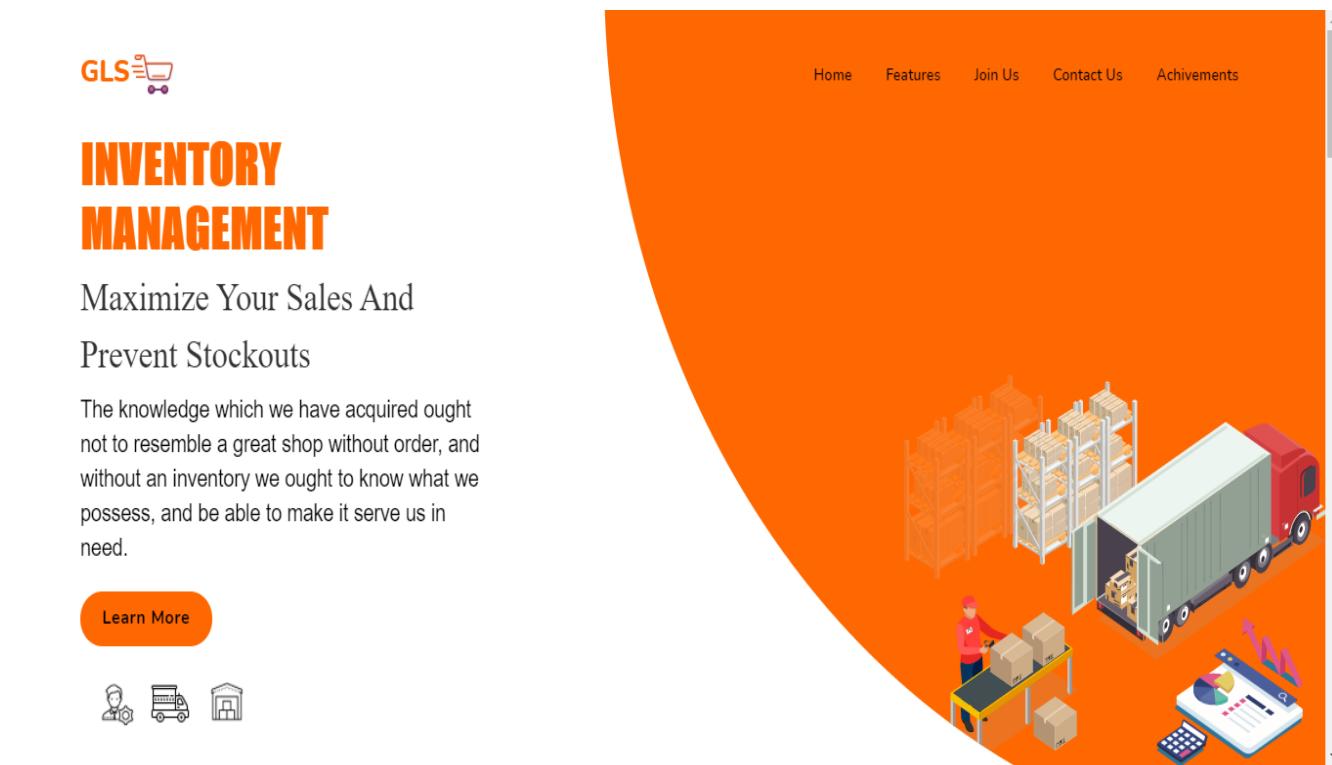


Figure 5: Home page

5.2 ADMIN MODULE

Admin module allows inventory administrator to set up back-end of the inventory and perform basic operations like Requesting order, Check Perishable goods, Check shop shelf stock, Check warehouse shop, Add user profile, Check payment, View order history.

INVENTORY MANGEMENT ADMIN DASHBOARD

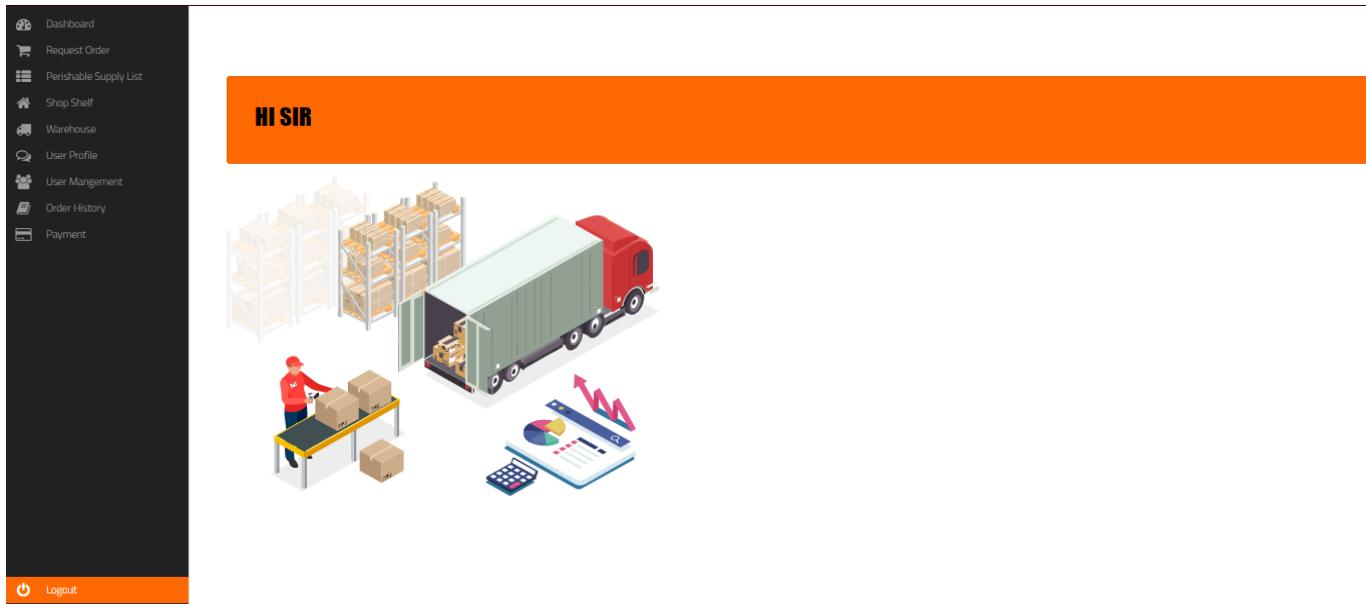


Figure 6: Admin Dashboard

5.3 VENDOR MODULE

INVENTORY MANGEMENT VENDOR DASHBOARD



Figure 7: Vendor Dashboard

5.4 SUPPLIER MODULE

INVENTORY MANAGEMENT SUPPLIER LOGIN PAGE

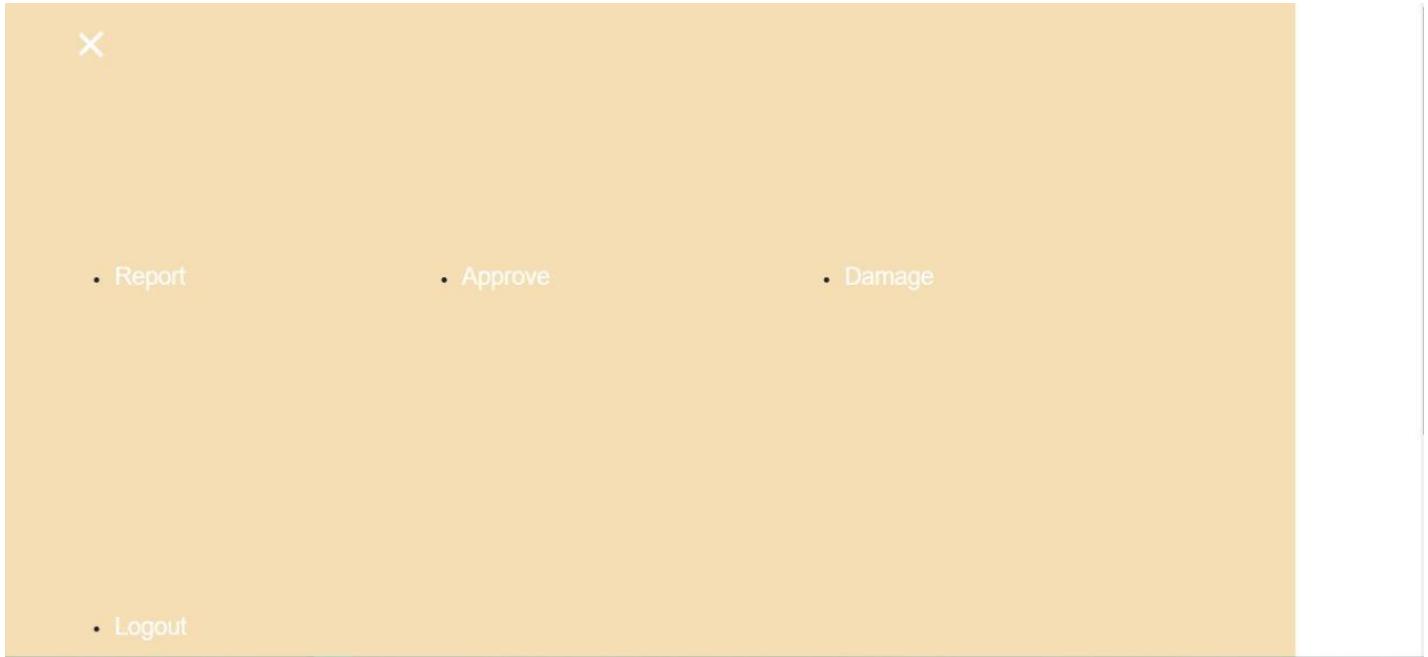


Figure 8: Supplier Dashboard page

5.5 CHECKER MODULE

INVENTORY MANAGEMENT CHECKER DASHBOARD

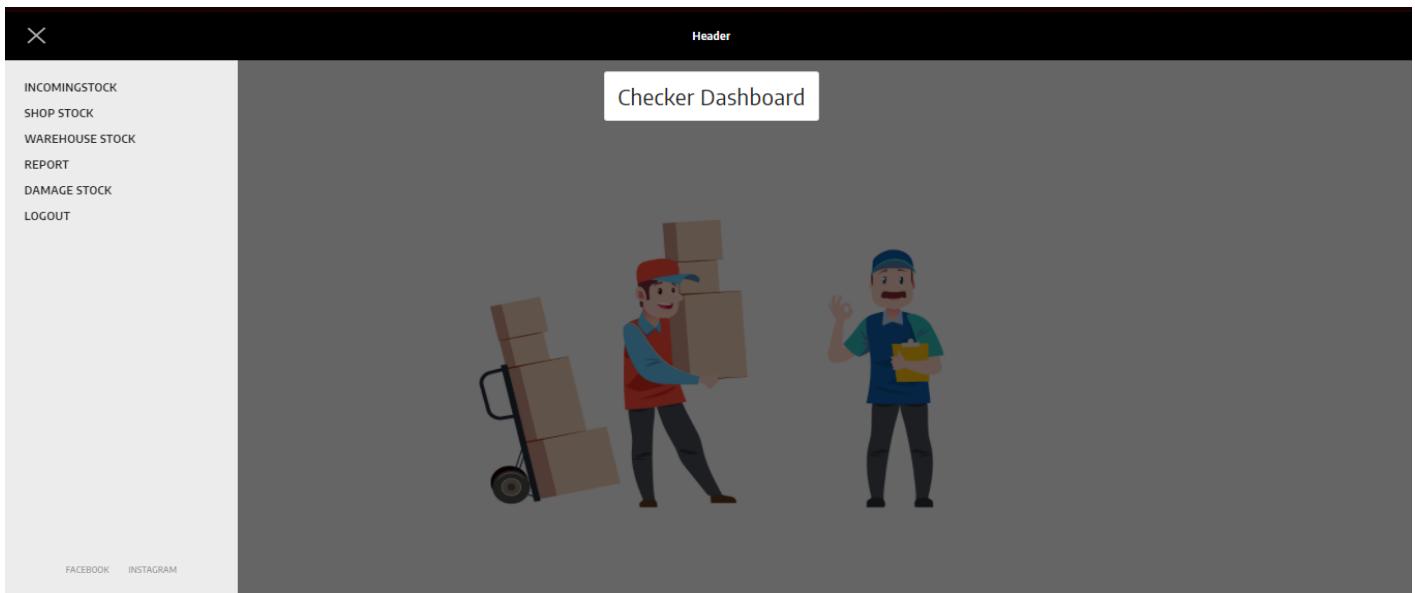


Figure 9: Checker Dashboard

5.6 MANAGER MODULE

INVENTORY MANGEMENT MANAGER DASHBOARD PAGE

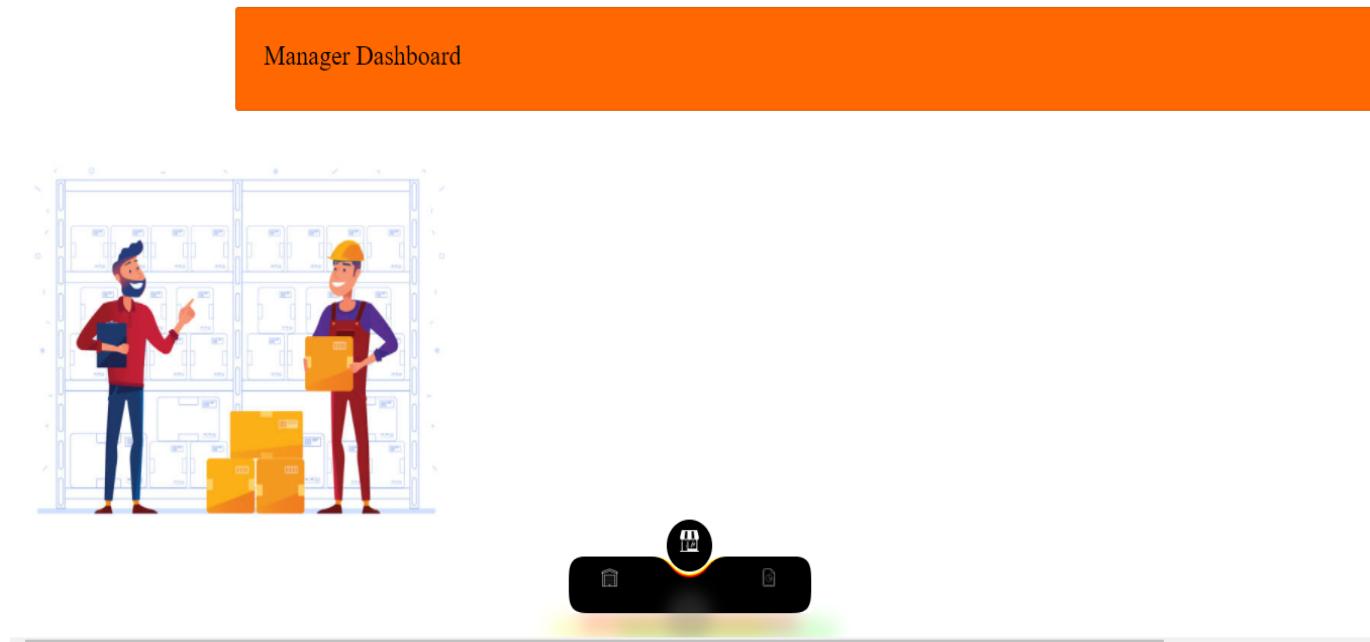


Figure 10: Manager Dashboard

6. IMPLEMENTATION

6.1 BUSINESS SCENARIOS:

BUSINESS SCENARIO 1

Shop.java:

```
package com.fruitshop;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Shop {
    public static void main(String[] args) {

        List<FruitsDetails> fd = new ArrayList<FruitsDetails>();

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of fruit detail that you wanna insert : ");
        int n=sc.nextInt();
        for (int i=0;i<n;i++) {

            System.out.println("Enter the fruit name");
            String fruitName= sc.next();
            System.out.println("Enter the fruit Type");
            String fruitType= sc.next();
            System.out.println("Enter weight of the fruit");
            double fruitWeight=sc.nextDouble();
            System.out.println("Enter the Distributor name");
            String distributor= sc.next();
            System.out.println("Enter the fruit category");
            String fruitcategory= sc.next();
            System.out.println("Enter the location");
        }
    }
}
```

```

String location= sc.next();
System.out.println("Enter the expiryDate");
String expiryDate= sc.next();
System.out.println("Enter the quantity");
int quantity= sc.nextInt();
System.out.println("Enter the cost");
double cost= sc.nextDouble();

fd.add(new FruitsDetails(fruitName, fruitType, distributor, fruitWeight, fruitcategory, location, expiryDate, quantity, cost,quantity*cost));

}

FruitsOperation fo=new FruitsOperation();
fo.getDetails((ArrayList<FruitsDetails>)fd);
fo.findByName((ArrayList<FruitsDetails>)fd,"latha");
fo.findByName((ArrayList<FruitsDetails>)fd,"apple");
fo.findByloc((ArrayList<FruitsDetails>)fd,"market");
sc.close();
}
}

```

FruitOperation.java:

```

package com.fruitshop;

import java.util.ArrayList;

public class FruitsOperation {
    void getDetails(ArrayList list )
    {
        for(int i=0;i<list.size();i++)
        {
            System.out.println("Entered List values are: ");
            System.out.println(list.get(i));
        }
    }
}

```

```

    }

}

void findByDis(ArrayList<FruitsDetails> list, String distName)
{
    System.out.println("-----");
    for(FruitsDetails a:list)
    {
        if(a.distributor.equals(distName))
        {
            System.out.println("DISTRIBUTOR MATCHED! " +distName);
            System.out.println("=====");
            System.out.println(a);
            System.out.println("=====");
        }
        else
        {
            System.out.println("No records found for the Distributor Name" +distName);
        }
    }
}

void findByName(ArrayList<FruitsDetails> list, String name )
{
    System.out.println("-----");
    for(FruitsDetails b:list)
    {
        if(b.fruitName.equals(name))
        {
            System.out.println("FRUIT NAME MATCH FOUND! "+name);
        }
    }
}

```

```

        System.out.println("=====");
        System.out.println(b);
        System.out.println("=====");
    });

} else {
    System.out.println("No records found for the Fruit name"+name);
}

}

}

void findByloc(ArrayList<FruitsDetails> list,String location)
{
    System.out.println("=====");
    System.out.println("-----");
}

for(FruitsDetails c:list)
{
    if(c.location.equals(location))
    {
        System.out.println("LOCATION MATCH FOUND! " +location);
        System.out.println("=====");
        System.out.println(c);
        System.out.println("=====");
    }
    else
    {
        System.out.println("No records found for the loation " +location);
    }
}
}

```

```
}
```

FruitDetails.java

```
package com.fruitshop;

public class FruitsDetails {
    String fruitName;
    String fruitType;
    String distributor;
    double fruitWeight;
    String fruitCategory;
    String location;
    String expiryDate;
    int quantity;
    double cost;
    double totalcost;

    public FruitsDetails (String fruitName, String fruitType, String distributor, double fruitWeight, String fruitCategory,
        String location, String expiryDate, int quantity, double cost, double totalcost) {
        super();
        this.fruitName = fruitName;
        this.fruitType = fruitType;
        this.distributor = distributor;
        this.fruitWeight = fruitWeight;
        this.fruitCategory = fruitCategory;
        this.location = location;
        this.expiryDate = expiryDate;
        this.quantity = quantity;
        this.cost = cost;
        this.totalcost = totalcost;
    }
}
```

```

public double costDetails(String fruitName)
{
switch(fruitName)
{
case "apple" :
    cost=100;
    totalcost= quantity*cost;
    break;
case "kiwi" :
    cost=200;
    totalcost= quantity*cost;
    break;
case "orange":
    cost=300;
    totalcost= quantity*cost;
    break;
case "muskmelon":
    cost=400;
    totalcost= quantity*cost;
    break;
case "grapes":
    cost=500;
    totalcost= quantity*cost;
    break;
default:
    System.out.println("Specified fruit not available");
}
return totalcost;
}

@Override
public String toString() {
    return "Fruitdetails [fruitName=" + fruitName + ", fruitType=" + fruitType + ", Distributor=" + distributor
        + ", fruitWeight=" + fruitWeight + ", fruitCategory=" + fruitCategory + ", location=" + location

```

```
+ ", expiryDate=" + expiryDate + ", quantity=" + quantity + ", Cost=" + cost + ", TotalCost=" + totalcost+
"]";
}
}
```

GROCERY CLASSS:

Shop.java:

```
package com.fruitshop.GroceryShop;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Shop {
    public static void main(String[] args) {

        List<GroceryDetails> fd = new ArrayList<GroceryDetails>();

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of grocery detail that you wanna insert : ");
        int n=sc.nextInt();
        for (int i=0;i<n;i++) {

            System.out.println("Enter the Grocery name");
            String groceryName= sc.next();
            System.out.println("Enter the Grocery Type");
            String groceryType= sc.next();
            System.out.println("Enter the Grocery weight");
            double groceryWeight= sc.nextDouble();
            System.out.println("Enter the Distributor name");
            String distributor= sc.next();
        }
    }
}
```

```

        System.out.println("Enter the Grocery category");
        String grocerycategory= sc.next();
        System.out.println("Enter the location");
        String location= sc.next();
        System.out.println("Enter the Expiry Date");
        String expiryDate= sc.next();
        System.out.println("Enter the quantity of the grocery");
        int quantity= sc.nextInt();
        System.out.println("Enter the cost");
        double cost= sc.nextDouble();
        fd.add(new GroceryDetails(groceryName, groceryType, distributor, groceryWeight, grocerycategory, location, expiryDate, quantity, cost, quantity*cost));
    }
    Groceryoperations go=new Groceryoperations();
    go.getDetails((ArrayList<GroceryDetails>)fd);
    go.findByDis((ArrayList<GroceryDetails>)fd,"Latta");
    go.sortByDistAndGroceryType((ArrayList<GroceryDetails>)fd);
    go.sortByCostAndExpiry((ArrayList<GroceryDetails>)fd);
    sc.close();
}

}

```

GrocerOperation.java:

```

package com.fruitshop.GroceryShop;
import java.util.ArrayList;
import java.util.Collections;

public class Groceryoperations {

    void getDetails(ArrayList list )
    {
        for(int i=0;i<list.size();i++)
        {

```

```

        System.out.println("Entered List values are: ");
        System.out.println(list.get(i));
    }
}

void sortByDistAndGroceryType(ArrayList<GroceryDetails> list)
{
    Collections.sort(list,new SortByDistributorAndGroceryType());
    for(int i=0;i<list.size();i++)
    {
        System.out.println("SORTED");
        System.out.println(list.get(i));
    }
}

void sortByCostAndExpiry(ArrayList<GroceryDetails> list)
{
    Collections.sort(list, new SortByCostAndExpirydate());
    for(int i=0;i<list.size();i++)
    {
        System.out.println("SORTED");
        System.out.println(list.get(i));
    }
}

void findByDis(ArrayList<GroceryDetails> list,String distName)
{
    for(GroceryDetails g:list)
    {
        if(g.Distributor.equals(distName))
        {
            System.out.println("DISTRIBUTOR NAME MATCHED! " +distName);
            System.out.println("=====");
            ====="");
            System.out.println(g);
        }
    }
}

```

```

        System.out.println("=====");
        =====");
    }
    else
    {
        System.out.println("No records found for the Distributor Name and grocery type" );
    }
}
}
}

```

GroceryDetails.java:

```

package com.fruitshop.GroceryShop;

public class GroceryDetails {

    String GroceryName;
    String GroceryType;
    String Distributor;
    double GroceryWeight;
    String GroceryCategory;
    String location;
    String expiryDate;
    int quantity;
    double cost;
    double totalcost;
    public GroceryDetails(String groceryName, String groceryType, String distributor, double groceryWeight,
        String groceryCategory, String location, String expiryDate, int quantity, double cost, double totalcost) {
        super();
        GroceryName = groceryName;
        GroceryType = groceryType;
        Distributor = distributor;
        GroceryWeight = groceryWeight;
        GroceryCategory = groceryCategory;
        this.location = location;
    }
}

```

```

        this.expiryDate = expiryDate;
        this.quantity = quantity;
        this.cost = cost;
        this.totalcost = totalcost;
    }

    public double costDetails(String GroceryName)
    {
        switch(GroceryName)
        {
            case "rice":cost=50;
            totalcost=quantity*cost;
            break;
            case "dhal":cost=180;
            totalcost=quantity*cost;
            break;
            case "oil":cost=160;
            totalcost=quantity*cost;
            break;
            case "snacks":cost=50;
            totalcost=quantity*cost;
            break;
            case "beverage":cost=75;
            totalcost=quantity*cost;
            break;
            default:System.out.println("Grocery not available");
        }
        return totalcost;
    }

    @Override
    public String toString() {
        return "Grocerydetails [GroceryName=" + GroceryName + ", GroceryType=" + GroceryType + ", Distributo
r="
        + Distributor + ", GroceryWeight=" + GroceryWeight + ", GroceryCategory=" + GroceryCategory
        + ", location=" + location + ", expiryDate=" + expiryDate + ", quantity=" + quantity + ", cost=" + cost
    }
}

```

```
+ ", totalcost=" + totalcost + "]";  
}  
  
}
```

SortByCostAndExpirydate.java

```
package com.fruitshop.GroceryShop;  
  
import java.util.Comparator;  
  
public class SortByCostAndExpirydate implements Comparator<GroceryDetails> {  
  
    @Override  
    public int compare(GroceryDetails o1, GroceryDetails o2) {  
        if(o1.GroceryType.compareTo(o2.GroceryType)==0)  
        {  
            return o1.expiryDate.compareTo(o2.expiryDate);  
        }  
        else  
        {  
            return o1.expiryDate.compareTo(o2.expiryDate);  
        }  
    }  
  
}
```

SortByDistibutorAndGrocyType.java

```
package com.fruitshop.GroceryShop;  
  
import java.util.Comparator;  
  
public class SortByDistibutorAndGrocyType implements Comparator<GroceryDetails> {
```

```
@Override
public int compare(GroceryDetails o1, GroceryDetails o2) {
    if(o1.Distributor.compareTo(o2.Distributor)==0)
    {
        return o1.GroceryType.compareTo(o2.GroceryType);
    }
    else
    {
        return o1.Distributor.compareTo(o2.Distributor);
    }
}
{
}
}
```

SortByNameAndLocation.java

```
package com.fruitshop.GroceryShop;
import java.util.Comparator;
public class SortByNameAndLocation implements Comparator<GroceryDetails> {

    @Override
    public int compare(GroceryDetails o1, GroceryDetails o2) {
        if(o1.GroceryName.compareTo(o2.GroceryName)==0)
        {
            return o1.location.compareTo(o2.location);
        }
        else
        {
            return o1.GroceryName.compareTo(o2.GroceryName);
        }
    }
}
```

```
}
```

OUTPUT:

FRUIT SHOP:

```
Enter the number of fruit detail that you wanna insert :
2
Enter the fruit name
apple
Enter the fruit Type
exotic
Enter weight of the fruit
20
Enter the Distributor name
latha
Enter the fruit category
fruit
Enter the location
chennai
Enter the expiryDate
25/2/1010
Enter the quantity
30
Enter the cost
300
Enter the fruit name
kiwi
Enter the fruit Type
frsh
Enter weight of the fruit
30
Enter the Distributor name
anbu
Enter the fruit category
frsh
Enter the location
market
Enter the expiryDate
34/4/2020
Enter the quantity
39
Enter the cost
```

```
anbu
Enter the fruit category
frsh
Enter the location
market
Enter the expiryDate
34/4/2020
Enter the quantity
39
Enter the cost
3000
Entered List values are:
Fruitdetails [fruitName=apple, fruitType=exotic, Distributor=latha, fruitWeight=20.0, fruitCategory=fruit, location=chennai, expiryDate=25/2/1010, quantity=30, Cost=300.0, TotalCost=9000.0]
Entered List values are:
Fruitdetails [fruitName=kiwi, fruitType=frsh, Distributor=anbu, fruitWeight=30.0, fruitCategory=frsh, location=market, expiryDate=34/4/2020, quantity=39, Cost=3000.0, TotalCost=117000.0]
=====
DISTRIBUTOR MATCHED! latha
=====
Fruitdetails [fruitName=apple, fruitType=exotic, Distributor=latha, fruitWeight=20.0, fruitCategory=fruit, location=chennai, expiryDate=25/2/1010, quantity=30, Cost=300.0, TotalCost=9000.0]
=====
No records found for the Distributor Name latha
=====
FRUIT NAME MATCH FOUND! apple
=====
Fruitdetails [fruitName=apple, fruitType=exotic, Distributor=latha, fruitWeight=20.0, fruitCategory=fruit, location=chennai, expiryDate=25/2/1010, quantity=30, Cost=300.0, TotalCost=9000.0]
=====
No records found for the Fruit name apple
=====
No records found for the location market
=====
LOCATION MATCH FOUND! market
=====
Fruitdetails [fruitName=kiwi, fruitType=frsh, Distributor=anbu, fruitWeight=30.0, fruitCategory=frsh, location=market, expiryDate=34/4/2020, quantity=39, Cost=3000.0, TotalCost=117000.0]
```

GROCERY:

```
Enter the number of grocery detail that you wanna insert :  
2  
Enter the Grocery name  
RICE  
Enter the Grocery Type  
CEREALS  
Enter the Grocery weight  
5  
Enter the Distributor name  
Latta  
Enter the Grocery category  
grains  
Enter the location  
chennai  
Enter the Expiry Date  
23/3/2020  
Enter the quantity of the grocery  
2  
Enter the cost  
5000  
Enter the Grocery name  
wheat  
Enter the Grocery Type  
cereals  
Enter the Grocery weight  
200  
Enter the Distributor name  
Anbu  
Enter the Grocery category  
grains  
Enter the location  
chennai  
Enter the Expiry Date  
26/3/2021  
Enter the quantity of the grocery  
2  
Enter the cost
```

```
Enter the number of grocery detail that you wanna insert :  
2  
Enter the Grocery name  
RICE  
Enter the Grocery Type  
CEREALS  
Enter the Grocery weight  
5  
Enter the Distributor name  
Latta  
Enter the Grocery category  
grains  
Enter the location  
chennai  
Enter the Expiry Date  
23/3/2020  
Enter the quantity of the grocery  
2  
Enter the cost  
5000  
Enter the Grocery name  
wheat  
Enter the Grocery Type  
cereals  
Enter the Grocery weight  
200  
Enter the Distributor name  
Anbu  
Enter the Grocery category  
grains  
Enter the location  
chennai  
Enter the Expiry Date  
26/3/2021  
Enter the quantity of the grocery
```

```
Enter the quantity of the grocery
2
Enter the cost
2900
Entered List values are:
Grocerydetails [GroceryName=RICE, GroceryType=CEREALS, Distributor=Latta, GroceryWeight=5.0, GroceryCategory=grains, location=chennai, expiryDate=23/3/2020
, quantity=2, cost=5000.0, totalcost=10000.0]
Entered List values are:
Grocerydetails [GroceryName=wheat, GroceryType=cereals, Distributor=Anbu, GroceryWeight=200.0, GroceryCategory=grains, location=chennai, expiryDate=26/3/2020
21, quantity=2, cost=2900.0, totalcost=5800.0]
DISTRIBUTOR NAME MATCHED! Latta
=====
Grocerydetails [GroceryName=RICE, GroceryType=CEREALS, Distributor=Latta, GroceryWeight=5.0, GroceryCategory=grains, location=chennai, expiryDate=23/3/2020
, quantity=2, cost=5000.0, totalcost=10000.0]
=====
No records found for the Distributor Name and grocery type
SORTED
Grocerydetails [GroceryName=wheat, GroceryType=cereals, Distributor=Anbu, GroceryWeight=200.0, GroceryCategory=grains, location=chennai, expiryDate=26/3/2020
21, quantity=2, cost=2900.0, totalcost=5800.0]
SORTED
Grocerydetails [GroceryName=RICE, GroceryType=CEREALS, Distributor=Latta, GroceryWeight=5.0, GroceryCategory=grains, location=chennai, expiryDate=23/3/2020
, quantity=2, cost=5000.0, totalcost=10000.0]
SORTED
Grocerydetails [GroceryName=RICE, GroceryType=CEREALS, Distributor=Latta, GroceryWeight=5.0, GroceryCategory=grains, location=chennai, expiryDate=23/3/2020
, quantity=2, cost=5000.0, totalcost=10000.0]
Grocerydetails [GroceryName=wheat, GroceryType=cereals, Distributor=Anbu, GroceryWeight=200.0, GroceryCategory=grains, location=chennai, expiryDate=26/3/2020
21, quantity=2, cost=2900.0, totalcost=5800.0]
```

BUSINESS SCENARIO 2

Sales Report Generation

Mart.java

```
package com.example.javausecase.corejava.generatereport;

import java.util.Scanner;

public class Emart {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int categoryChoice;
        int productChoice1 = 0;
        int productChoice2 = 0;
        int productChoice3 = 0;

        System.out.println("-----");
        System.out.println("\t\t\t\t JK TRADERS");
        System.out.println("\t\t\t Best Price With Best Quality");
        System.out.println("-----");

        System.out.println("Enter Your Name");
        String username = sc.next();

        System.out.println("\t\t\t\t WELCOME " +username);
        System.out.println("\t\t\t\t Here is the List of Category Available:");

        System.out.println("\t\t\t\t 1. Drinks \n \t\t\t\t 2. Vegetables \n \t\t\t\t 3. Foodgrains");
```

```
System.out.println("-----  
-----");
```

```
System.out.println("\t\t\t\t\t Enter your option");  
categoryChoice = sc.nextInt();  
Categories types = new Categories();  
Products p1 = new Products();  
if (categoryChoice == 1) {  
    types.beverages();  
    System.out.println("\t\t\t\t\t Choose your Product:");  
    productChoice1 = sc.nextInt();  
  
    if (productChoice1 == 1) {  
        p1.beveragecocooorder();  
    }  
    if (productChoice1 == 2) {  
        p1.beveragepepsiorder();  
    }  
}  
if (categoryChoice == 2) {  
    types.vegetables();  
    System.out.println("\t\t\t\t\tChoose your Product:");  
    productChoice2 = sc.nextInt();  
  
    if (productChoice2 == 1) {  
        p1.vegetablesExoticProducts();  
    }  
    if (productChoice2 == 2) {  
        p1.vegetablesOrganicProducts();  
    }  
}  
if (categoryChoice == 3) {  
    types.foodgrains();  
    System.out.println("\t\t\t\t\tChoose your Product:");
```

```

productChoice3 = sc.nextInt();

if (productChoice3 == 1) {
    p1.foodGrainFlourProducts();
}

if (productChoice3 == 2) {
    p1.foodGrainRiceProducts();
}

else{
    System.out.println("You have entered undefined option");
}

if (catogoryChoice == 1 && productChoice1 == 1) {
    FinalPayment pay = new FinalPayment();
    pay.cocolapay();
}

else if (catogoryChoice == 1 && productChoice1== 2) {
    FinalPayment pay = new FinalPayment();
    pay.pepsipay();
}

else if (catogoryChoice == 2 && productChoice2 == 1) {
    FinalPayment pay = new FinalPayment();
    pay.tomatopay();
}

else if (catogoryChoice == 2 && productChoice2 == 2) {
    FinalPayment pay = new FinalPayment();
    pay.potatopay();
}

else if (catogoryChoice == 3 && productChoice3 == 1) {
    FinalPayment pay = new FinalPayment();
    pay.ricepay();
}

```

```
    }

    else if (categoryChoice == 3 && productChoice3 == 2) {

        FinalPayment pay = new FinalPayment();
        pay.attapay();
    }

    sc.close();
}

}
```

BeveragesProducts.java

```
package com.example.javausecase.corejava.generatereport;

public class BeveragesProducts {

    int pid ;
    int qty;
    int rate ;
    int gst;
    int totalAmt;
    String brand;

    public BeveragesProducts(int pid, int qty, int rate, int gst, int totalAmt, String brand) {
        super();
        this.pid = pid;
        this.qty = qty;
        this.rate = rate;
        this.gst = gst;
        this.totalAmt = totalAmt;
        this.brand = brand;
    }
}
```

```

public void showcocoladetails() {

    System.out.println("-----");
    System.out.println("ProductDetails:\n-----\nProduct Id = " + this.pid + "\nProduct Price = Rs." + this.rate + "\nProduct Brand = " + this.brand +
        "\nGST Amount = " + this.gst + "%" + "\nQuantity= " + this.qty+ "\nTotalPrice = Rs." + this.qty*this.rate );
    this.totalAmt= this.rate*this.qty;
}

public void showPepsidetails() {

    System.out.println("-----");
    System.out.println("ProductDetails:\n-----\nProduct Id = " + this.pid + "\nProduct Price = Rs." + this.rate + "\nProduct Brand = " + this.brand +
        "\nGST Amount = " + this.gst + "%" + "\nQuantity= " + this.qty+ "\nTotalPrice = Rs." + this.qty*this.rate);
    this.totalAmt= this.rate*this.qty;
}

public void cocoprince() {

    System.out.println("YOUR BILL");
    System.out.println("[Total Price * Gst Charged/100 ]");
    System.out.println("-----");
    System.out.println("Product Price = Rs." + this.totalAmt*this.gst/100 + "\nGst Charged = " + this.gst + "%");
}

public void pepsiPrice() {

    System.out.println("YOUR BILL");
    System.out.println("[Total Price * Gst Charged/100 ]");
    System.out.println("-----");
}

```

```
        System.out.println("Product Price = Rs." + this.totalAmt*this.gst/100 + "\nGst Charged = " + this.gst + "%");
    );
}
}
```

Catogories.java

```
package com.example.javausecase.corejava.generatereport;

public class Catogories {
    public void beverages() {

        System.out.println("\t\t\t\t Welcome to Emart");
        System.out.println("\t\t\tWe Sell Approved and Legal Beverages Only");
        System.out.println("\t\t\t\t 1.SoftDrinks\n\t\t\t\t 2.Alchohol");
    }

    public void foodgrains() {

        System.out.println("\t\t\t\t Welcome to Emart");
        System.out.println("\t\t\tWe Sell All Kinds of Foodgrains ");
        System.out.println("\t\t\t\t 1.Flour\n\t\t\t\t 2.Rice");
    }

    public void vegetables() {

        System.out.println("\t\t\t\t Welcome to Emart");
        System.out.println("\t\t\tWe Sell Fresh Vegetables ");
        System.out.println("\t\t\t\t 1.Exotic\n\t\t\t\t 2.Organic");
    }
}
```

FinalPayment.java

```
package com.example.javausecase.corejava.generatereport;

public class FinalPayment {
    Products p = new Products();
    Products prod = new Products();
    public void cocolapay() {
        System.out.println(" ");
        System.out.println("Order Summary");
        System.out.println("=====");
        p.cocorate();
        System.out.println("Only Cash On Delivary Available");
        System.out.println("Your Order Will be Delivered within a day");
        System.out.println("No ReFund will be Allowed ");
        System.out.println("HAPPY SHOPPING");
    }
    public void pepsipay() {
        System.out.println(" ");
        System.out.println("Order Summary");
        System.out.println("=====");
        p.pepsirate();
        System.out.println("Only Cash On Delivary Available");
        System.out.println("Your Order Will be Delivered within a day");
        System.out.println("No ReFund will be Allowed ");
        System.out.println("HAPPY SHOPPING");
    }
}
```

```
}
```

```
public void tomatopay() {  
    System.out.println(" ");  
    System.out.println("Order Summary");  
    System.out.println("=====");  
    prod.tomatorate();  
    System.out.println("Your Order Will be Delivered within a day");  
    System.out.println("No ReFund will be Allowed ");  
    System.out.println("HAPPY SHOPPING");  
}
```

```
public void potatopay() {  
    System.out.println(" ");  
    System.out.println("Order Summary");  
    System.out.println("=====");  
    prod.potatorate();  
    System.out.println("Your Order Will be Delivered within a day");  
    System.out.println("No ReFund will be Allowed ");  
    System.out.println("HAPPY SHOPPING");  
}
```

```
public void ricepay() {  
    System.out.println(" ");  
    System.out.println("Order Summary");  
    System.out.println("=====");  
    prod.ricerate();  
    System.out.println("Your Order Will be Delivered within a day");  
    System.out.println("No ReFund will be Allowed ");  
    System.out.println("HAPPY SHOPPING");  
}
```

```
public void attapay() {  
    System.out.println(" ");  
    System.out.println("Order Summary");
```

```

        System.out.println("=====");
        prod.attarate();
        System.out.println("Your Order Will be Delivered Next day");
    }
}

```

FoodgrainsProducts.java

```

package com.example.javausecase.corejava.generatereport;

public class FoodgrainsProducts {
    int pid ;
    int qty;
    int rate ;
    int gst;
    int totalAmt;
    String brand;

    public FoodgrainsProducts(int pid, int qty, int rate, int gst, int totalAmt, String brand) {
        super();
        this.pid = pid;
        this.qty = qty;
        this.rate = rate;
        this.gst = gst;
        this.totalAmt = totalAmt;
        this.brand = brand;
    }

    public void showattadetails() {

        System.out.println("-----");
        System.out.println("ProductDetails:\n-----\nProduct Id = " + this.pid + "\nProduct Price = Rs." + this.rate + "\nProduct Brand = " + this.brand +
                "\nGST Amount = " + this.gst + "%" + "\nQuantity= " + this.qty+ "\nTotalPrice = Rs." + this.qty*this.rate );
    }
}

```

```

        this.totalAmt= this.rate*this.qty;
    }

public void showricedetails() {
    System.out.println("Product Details");
    System.out.println("-----");
    System.out.println("ProductDetails:\n-----\nProduct Id = " + this.pid + "\nProduct Price = Rs." + this.rate + "\nProduct Brand = " + this.brand +
    "\nGST Amount = " + this.gst + "%"\nQuantity= " + this.qty + "\nTotalPrice = Rs." + this.qty*this.rat
e );
    this.totalAmt= this.rate*this.qty;

}

public void ricePrice() {
    System.out.println("Product Details");
    System.out.println("[Total Price * Gst Charged/100 ]");
    System.out.println("-----");
    System.out.println("Product Price = Rs." + this.totalAmt*this.gst/100 + "\nGst Charged = " + this.gst + "%"
);
}

public void attaPrice() {
    System.out.println("Product Details");
    System.out.println("[Total Price * Gst Charged/100 ]");
    System.out.println("-----");
    System.out.println("Product Price = Rs." + this.totalAmt*this.gst/100 + "\nGst Charged = " + this.gst + "%"
);
}

}

```

Products.java

```
package com.example.javausecase.corejava.generatereport;

public class Products {
    static BeveragesProducts b;
    static VegetablesProducts v;
    static FoodgrainsProducts fg;

    public void beveragecocooorder() {
        b = new BeveragesProducts(8, 700, 130, 12, 2, "COCOLA");
        b.showcocoladetails();
    }

    public void beveragepepsiorder() {
        b = new BeveragesProducts(1236, 200, 145, 12, 3, "Beer");
        b.showPepsidetails();
    }

    public void vegetablesOrganicProducts() {
        v = new VegetablesProducts(4506, 450, 150, 12, 4, "Organic");
        v.showtomatodetails();
    }

    public void vegetablesExoticProducts() {
        v = new VegetablesProducts(1065, 550, 180, 14, 5, "Exotic");
        v.showpotatodetails();
    }

    public void foodGrainRiceProducts() {
        fg = new FoodgrainsProducts(1302, 900, 250, 16, 6, "BASMATI");
        fg.showattadetails();
    }
}
```

```

    }

public void foodGrainFlourProducts() {
    fg = new FoodgrainsProducts(1201, 800, 350, 18, 7, "CHAAKI ATTA");
    fg.showricedetails();
}

public void cocorate() {
    b.cocoprice();
}

```

VegetablesProducts.java

```

package com.example.javausecase.corejava.generatereport;

public class VegetablesProducts {
    int pid ;
    int qty;
    int rate ;
    int gst;
    int totalAmt;
    String brand;
    public VegetablesProducts(int pid, int qty, int rate, int gst, int totalAmt, String brand) {
        super();
        this.pid = pid;
        this.qty = qty;
        this.rate = rate;
        this.gst = gst;
        this.totalAmt = totalAmt;
        this.brand = brand;
    }
    public void showtomatodetails() {
        System.out.println("Product Details");
    }
}

```

```

System.out.println("-----");
System.out.println("ProductDetails:\n-----\nProduct Id = " + this.pid + "\nProduct Price = Rs." + this.rate + "\nProduct Brand = " + this.brand +
"\nGST Amount = " + this.gst + "%" + "\nQuantity= " + this.qty+ "\nTotalPrice = Rs." + this.qty*this.rate );
this.totalAmt= this.rate*this.qty; }

public void showpotatodetails() {
System.out.println("Product Details");
System.out.println("-----");
System.out.println("ProductDetails:\n-----\nProduct Id = " + this.pid + "\nProduct Price = Rs." + this.rate + "\nProduct Brand = " + this.brand +
"\nGST Amount = " + this.gst + "%" + "\nQuantity= " + this.qty+ "\nTotalPrice = Rs." + this.qty*this.rate );
this.totalAmt= this.rate*this.qty;
}

public void tomatoprice() {

System.out.println("Product Details");
System.out.println("-----");
System.out.println("Product Price = Rs." + this.totalAmt*this.gst/100 + "\nGst Charged = " + this.gst + "%");
}

public void potatoPrice() {
System.out.println("Product Details");
System.out.println("-----");
System.out.println("Product Price = Rs." + this.totalAmt*this.gst/100 + "\nGst Charged = " + this.gst + "%");
}
}

```

OUTPUT:

```
JK TRADERS
Best Price With Best Quality

Enter Your Name
Latha
WELCOME Latha
Here is the List of Category Available:
1. Drinks
2. Vegetables
3. Foodgrains

Enter your option
1
Welcome to Emart
We Sell Approved and Legal Beverages Only
1.SoftDrinks
2.Alchohol
Choose your Product:
1

ProductDetails:
-----
Product Id = 8
Product Price = Rs.130
Product Brand = COCOLA
GST Amount = 12%
Quantity= 700
TotalPrice = Rs.91000
You have entered undefined option

Order Summary
=====
YOUR BILL
YOUR BILL
[Total Price * Gst Charged/100 ]

Product Price = Rs.10920
Gst Charged = 12%
Only Cash On Delivery Available
Your Order Will be Delivered within a day
No Refund will be Allowed
HAPPY SHOPPING

D:\CoreJava>
```

BUSINESS SCENARIO 3

AirVoice In Memory Processing

AirProcess.java:

```
package com.example.javausecase.corejava.AirTelecomProcess;

import java.util.ArrayList;
import java.util.Scanner;

public class AirProcess {
    public static void main(String[] args) {
        Filtering f= new Filtering();
        Scanner sc = new Scanner(System.in);

        String name;
        System.out.println("\t\t\t\t\t Air Voice Process");
        System.out.println("-----");
-----");
        System.out.println("Enter Employee Name");

        name= sc.next();
        System.out.println("-----");
        System.out.println("Welcome To Air Voice " + name);

        System.out.println("-----");
-----");
        System.out.println("\t\t\t\t\t Customer Billing History");

        ArrayList<Customer> custList = new ArrayList<Customer>();

        custList.add(new Customer(101, "a", "Uganda", "Junipe street",true, "6111111111", "10%", "no", 3577, "Pre
paid" ));
    }
}
```

```

        custList.add(new Customer(102, "d", "South Africa", "Jacobs street",true, "2222222222", "20%", "no", 2000
, "Prepaid"));

        custList.add(new Customer(103, "c", "Uganda", "Austin Corner",true, "6133333333", "30%", "yes", 1500, "
Postpaid"));

        custList.add(new Customer(104, "b", "South Africa", "Duck Manor",true, "4444444444", "40%", "no", 700,
"Prepaid"));

        custList.add(new Customer(105, "e", "Tanzania", "Castle Lanes",false, "5655555555", "50%", "no", 300, "
Prepaid"));

        custList.add(new Customer(106, "j", "Tanzania", "Crawford Beeches",true, "6666666666", "60%", "yes", 3
50, "Postpaid"));

        custList.add(new Customer(107, "k", "Nigeria", "Bryn Nebo",true, "5677777777", "70%", "yes", 5000, "Pre
paid"));

        custList.add(new Customer(108, "h", "Rwanda", "Bryn Nebo",true, "6188888888", "80%", "yes", 15000, "P
repaid"));

        custList.add(new Customer(109, "i", "Tanzania", "Morrison Corner",false, "9999999999", "120%", "no", 1
90, "Postpaid"));

        custList.add(new Customer(101, "g", "Rwanda", "Marina Manor",true, "6112121212", "106%", "yes", 2300,
"Prepaid"));

        custList.add(new Customer(111, "f", "Rwanda", "Dove Corner",true, "5613131313", "109%", "no", 560, "pr
epaid"));
    
```

```

System.out.println("Enter How Many Records you Need To print");
int n = sc.nextInt();
if(n >= 10)
{
    System.out.println("Printing Minimal Amount of data");
    f.getFirstTenRecords(custList, 10);
}
else
{
    System.out.println("Here is the Requested Records");
}
    
```

```

f.getFirstTenRecords(custList, n);
}

String custName ;
String phoneNumber;
Integer cid;
System.out.println("-----");
-----");
System.out.println("\t\t\t\t\t Filter Records");
System.out.println("-----");
-----");
System.out.println("Enter The Customer Name");
custName = sc.next();

System.out.println("Enter The Customer Phone Number");
phoneNumber = sc.next();

System.out.println("-----");
-----");
System.out.println("Sort By Name And Phone Number");
f.searchByCustNameAndPhoneNumber(custList, custName, phoneNumber);
System.out.println("Entered Data has been Fetched");
System.out.println("=====");
=====");
System.out.println("customer base country is Tanzania, Postpaid connection with no VAS enabled");
f.searchByCountryAndVas1(custList);
System.out.println("=====");
=====");
System.out.println("Customer is from “South Africa” & “Rwanda” and has Prepaid connection with atleast one VAS enabled");
f.searchByCountryAndVas2(custList);
System.out.println("=====");
=====");

```

```

        System.out.println("customer base country is Uganda, 4g Activated, phones that starts with 61");
        f.sortByBaseNumber(custList);
        System.out.println("=====");
=====");
=====");

System.out.println("Customers having common id");
System.out.println("Enter The Customer id to search: ");
cid= sc.nextInt();
f.printByCustomerId(custList,cid);
//f.printByCustomerId(custList, custid);
System.out.println("-----");
-----");
System.out.println("Customer base country is Nigeria, phones that starts with 61, and Name has been converted to uppercase");
f.sortByBaseNumber2(custList);
sc.close();

}
}

```

Customer.java

```

package com.example.javausecase.corejava.AirTelecomProcess;

public class Customer {
    public Integer custId;
    public String customerName;
    public String baseCountry;
    public String address;
    public boolean active;
    public String phoneNumber;
    public String avlCredits;
    public String is4g;
    public Integer billAmount;
    public String description;
}

```

```

public Customer(Integer custId, String customerName, String baseCountry, String address, boolean active,
String phoneNumber, String avlCredits, String is4g, Integer billAmount, String description) {
super();
this.custId = custId;
this.customerName = customerName;
this.baseCountry = baseCountry;
this.address = address;
this.active = active;
this.phoneNumber = phoneNumber;
this.avlCredits = avlCredits;
this.is4g = is4g;
this.billAmount = billAmount;
this.description = description;
}
@Override
public String toString() {
return "Customer [customerId=" + custId + ", customerName=" + customerName + ", baseCountry=" + base
Country
+ ", address=" + address + ", isActive=" + active + ", phoneNumber=" + phoneNumber + ", avlCredits="
+ avlCredits + ", is4g="
+ is4g + ", billAmount=" + billAmount + ", description=" + description + " ]";
}
public Integer getCustomerId() {
return custId;
}
public void setCustomerId(Integer custId) {
this.custId = custId;
}
public String getCustomerName() {
return customerName;
}

```

```
}
```

```
public void setCustomerName(String customerName) {
```

```
    this.customerName = customerName;
```

```
}
```

```
public String getBaseCountry() {
```

```
    return baseCountry;
```

```
}
```

```
public void setBaseCountry(String baseCountry) {
```

```
    this.baseCountry = baseCountry;
```

```
}
```

```
public String getAddress() {
```

```
    return address;
```

```
}
```

```
public void setAddress(String address) {
```

```
    this.address = address;
```

```
}
```

```
public boolean isActive() {
```

```
    return active;
```

```
}
```

```
public void setActive(boolean active) {
```

```
    this.active = active;
```

```
}
```

```
public String getPhoneNumber() {
```

```
    return phoneNumber;
```

```
}
```

```
public void setPhoneNumber(String phoneNumber) {  
    this.phoneNumber = phoneNumber;  
}
```

```
public String getAvlCredits() {  
    return avlCredits;  
}
```

```
public void setAvlCredits(String avlCredits) {  
    this.avlCredits = avlCredits;  
}
```

```
public String getIs4g() {  
    return is4g;  
}
```

```
public void setIs4g(String is4g) {  
    this.is4g = is4g;  
}  
public Integer getBillAmount() {  
    return billAmount;  
}
```

```
public void setBillAmount(Integer billAmount) {  
    this.billAmount = billAmount;  
}
```

```
public String getDescription() {  
    return description;  
}
```

```
public void setDescription(String description) {  
    this.description = description;  
}
```

```
}
```

Filtering.java

```
package com.example.javausecase.corejava.AirTelecomProcess;

import java.util.ArrayList;
import java.util.List;
import java.util.stream.Collectors;

public class Filtering {
    public void getFirstTenRecords(ArrayList<Customer> cust,int n)
    {
        for(int i= 0; i < n ;i++)
        {
            System.out.println(cust.get(i));
        }
    }

    public void searchByCustNameAndPhoneNumber(ArrayList<Customer> cust, String custName ,String phone number)
    {
        cust.stream().filter(c-> c.customerName.equals(custName) && c.phoneNumber.equals(phone number))
        .map(c->c)
        .forEach(System.out::println);
    }

    public void searchByCountryAndVas1(ArrayList<Customer> cust)
    {
        cust.stream().filter(c-> c.baseCountry.equals("Tanzania") && c.description.equals("Postpaid") && c.active
== true)
        .collect(Collectors.toSet())
        .forEach(System.out::println);
    }
}
```

```

public void searchByCountryAndVas2(ArrayList<Customer> cust)
{
    cust.stream().filter(c-> c.baseCountry.equals("South Africa") && c.baseCountry.equals("Rwanda") && c.description.equals("Prepaid") && c.active == true)
        .map(c->c)
        .forEach(System.out::println);
}

//public void printByCustomerId(ArrayList<Customer> cust, String custName ,String cid)

public void printByCustomerId(ArrayList<Customer> cust , Integer cid)
{
    List list = cust.stream().filter(c-> c.custId.equals(cid)).collect(Collectors.toList());

    if(list.isEmpty())
    {
        System.out.println("No Record Found");
    }
    else
    {
        System.out.println("Customers with matching id found");
        System.out.println(list);
    }
}

public void sortByBaseNumber(ArrayList<Customer> cust)
{
    cust.stream().filter(c-> c.baseCountry.equals("Uganda") || c.phoneNumber.startsWith(c.phoneNumber, 61) &
& c.is4g.equals("yes"))
        .map(c->c)
        .forEach(System.out::println);
}

public void sortByBaseNumber2(ArrayList<Customer> cust)

```

```

    {
        cust.stream().filter(c-> c.baseCountry.equals("Nigeria") || c.phoneNumber.startsWith(c.phoneNumber, 56))
        .map(c -> c.customerName.toUpperCase())
        .foreach(System.out::println);
    }
}

```

OUTPUT:

Air Voice Process

Enter Employee Name
Latha

Welcome To Air Voice Latha

Customer Billing History

Enter How Many Records you Need To print
10

Printing Minimal Amount of data

Customer [customerId=101, customerName=a, baseCountry=Uganda, address=Junipe street,isActive=true, phoneNumber=6111111111, avlCredits=10%, is4g=no, billAmount=3577, description=Prepaid]
Customer [customerId=102, customerName=d, baseCountry=South Africa, address=Jacobs street,isActive=true, phoneNumber=2222222222, avlCredits=20%, is4g=no, billAmount=2000, description=Prepaid]
Customer [customerId=103, customerName=c, baseCountry=Uganda, address=Austin Corner,isActive=true, phoneNumber=6133333333, avlCredits=30%, is4g=yes, billAmount=1500, description=Postpaid]
Customer [customerId=104, customerName=b, baseCountry=South Africa, address=Duck Manor,isActive=true, phoneNumber=4444444444, avlCredits=40%, is4g=no, billAmount=700, description=Prepaid]
Customer [customerId=105, customerName=e, baseCountry=Tanzania", address=Castle Lanes,isActive=false, phoneNumber=5655555555, avlCredits=50%, is4g=no, billAmount=300, description=Prepaid]
Customer [customerId=106, customerName=j, baseCountry=Tanzania", address=Crawford Beeches,isActive=true, phoneNumber=6666666666, avlCredits=60%, is4g=yes, billAmount=350, description=Postpaid]
Customer [customerId=107, customerName=k, baseCountry=Nigeria, address=Bryn Nebo,isActive=true, phoneNumber=5677777777, avlCredits=70%, is4g=yes, billAmount=5000, description=Prepaid]
Customer [customerId=108, customerName=h, baseCountry=Rwanda, address=Bryn Nebo,isActive=true, phoneNumber=6188888888, avlCredits=80%, is4g=yes, billAmount=15000, description=Prepaid]
Customer [customerId=109, customerName=i, baseCountry=Tanzania", address=Morrison Corner,isActive=false, phoneNumber=9999999999, avlCredits=120%, is4g=no, billAmount=190, description=Postpaid]
Customer [customerId=101, customerName=g, baseCountry=Rwanda, address=Marina Manor,isActive=true, phoneNumber=6112121212, avlCredits=106%, is4g=yes, billAmount=2300, description=Prepaid]

Filter Records

Enter The Customer Name
d

Enter The Customer Phone Number
2222222222

Sort By Name And Phone Number

Customer [customerId=102, customerName=d, baseCountry=South Africa, address=Jacobs street,isActive=true, phoneNumber=2222222222, avlCredits=20%, is4g=no, billAmount=2000, description=Prepaid]
Entered Data has been Fetched

customer base country is Tanzania, Postpaid connection with no VAS enabled

Customer is from "South Africa" & "Rwanda" and has Prepaid connection with atleast one VAS enabled

customer base country is Uganda, 4g Activated, phones that starts with 61

Customer [customerId=101, customerName=a, baseCountry=Uganda, address=Junipe street,isActive=true, phoneNumber=6111111111, avlCredits=10%, is4g=no, billAmount=3577, description=Prepaid]
Customer [customerId=103, customerName=c, baseCountry=Uganda, address=Austin Corner,isActive=true, phoneNumber=6133333333, avlCredits=30%, is4g=yes, billAmount=1500, description=Postpaid]

Customers having common id

7. TESTING

7.1 TEST STRATEGIES

Web Testing is checking our web application or website for potential bugs before it is made live and is accessible to general public. Web Testing checks for functionality, usability, security, compatibility, performance of the web application or website. Depending on our web testing requirements different techniques can be enhanced.

FUNCTIONALITY TESTING:

The webpages need to have appropriate links. There should be no broken links. The data in the forms is submitted to a live database or is linked to a working email address. The forms in the webpages should have better readability.

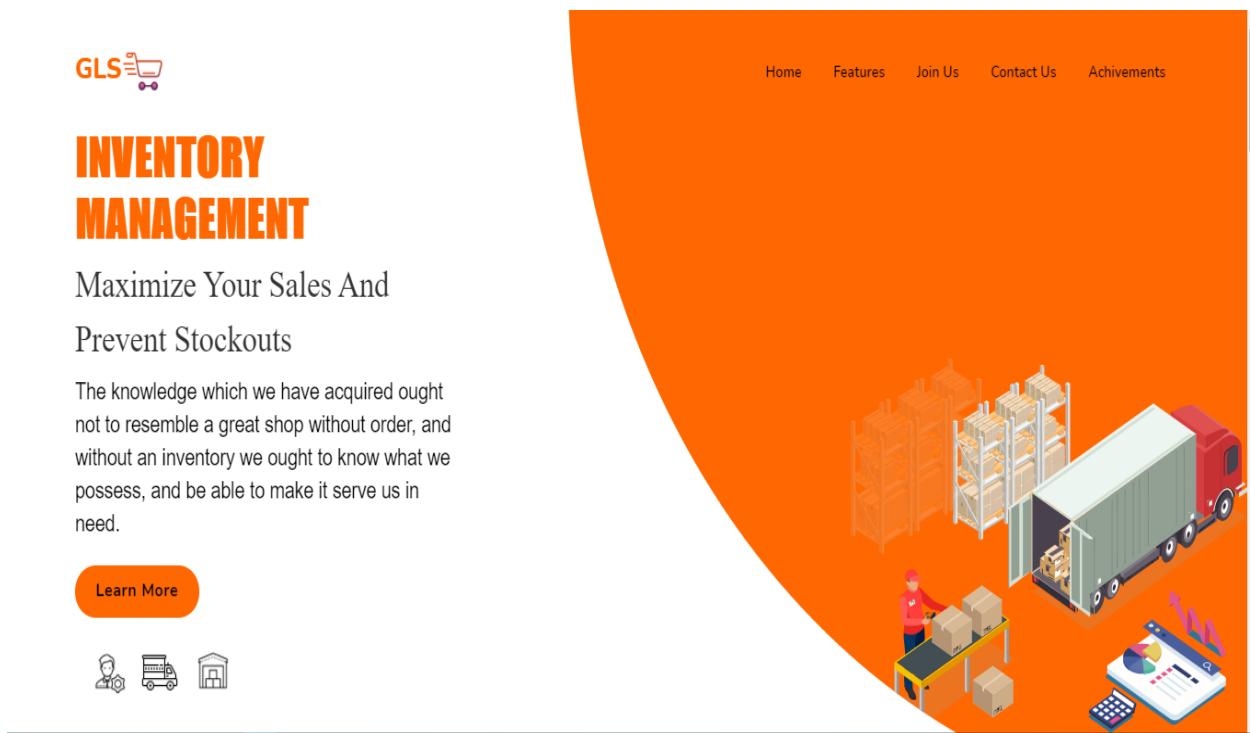


Figure 11: Inventory Management

USABILITY TESTING:

Usability Testing is been a vital part of any web-based project. The navigation on the

site is to be checked. Menu, buttons or links to different pages on our site should be easily visible and consistent on all webpages. The content on the website should be legible with no spelling or grammatical errors.



Figure 12: Navigation menu on Vendor Dashboard

Figure 13: Product Type in Dropdown

Request Products

Type Of The Product

- Type Of The Product
- Beverages
- Baby Products
- Vegetables
- Fruits
- Food Grains
- Cleaning

Perishable

10kgs

NEW

Add Submit

INTERFACE TESTING:

Interface Testing includes checking of the requests are sent correctly to the database and output at the client side is displayed correctly. We need to make sure that queries sent to the database give expected results.

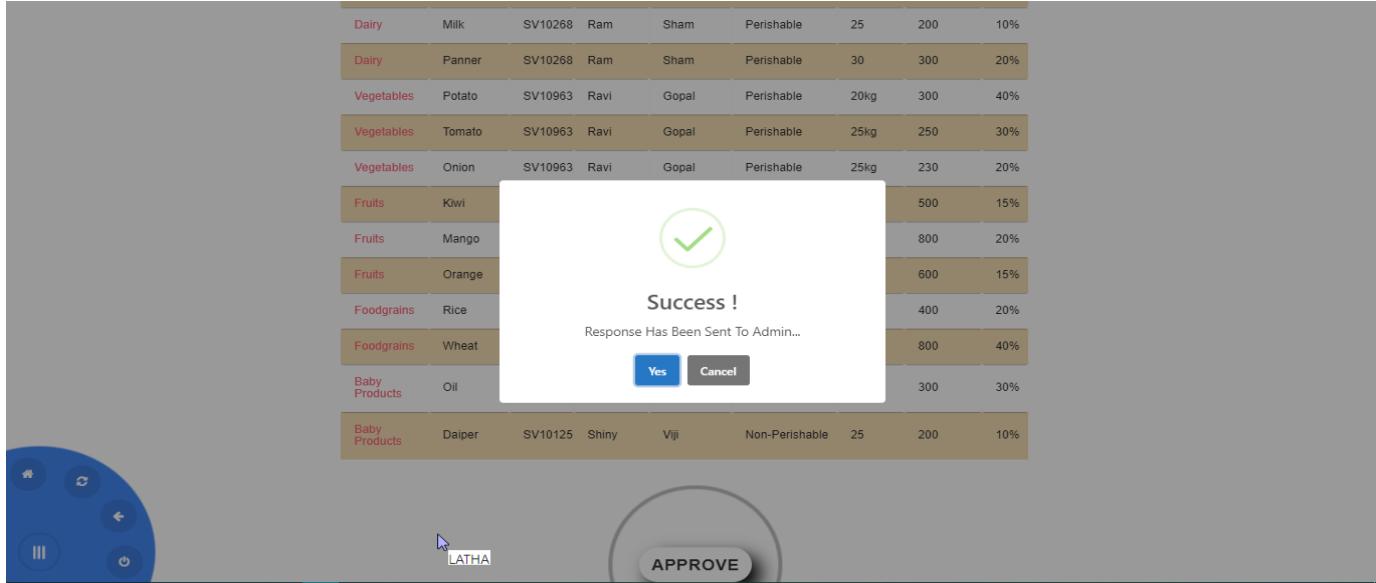


Figure 14: Vendor's Response Storing in database

```
[{"OrderId": "Lan321456",
"ProductId": "FA10234",
"ProductType": "Fruits",
"ProductName": "Kiwi",
"InDate": "12/3/20",
"OutDate": "16/3/20",
"VendorId": "SV10287",
"ProductQty": "10Kg"}, {"OrderId": "Lan753357",
"ProductId": "VE10234",
"ProductType": "Vegetables",
"ProductName": "Tomato",
"InDate": "15/3/20",
"OutDate": "20/3/20",
"VendorId": "SV10298",
"ProductQty": "15Kg"}, {"OrderId": "Lan753357",
"ProductId": "VE10234",
"ProductType": "Vegetables",
"ProductName": "Tomato",
"InDate": "15/3/20",
"OutDate": "20/3/20",
"VendorId": "SV10298",
"ProductQty": "15Kg"}, {"OrderId": "Lan954876",
"ProductId": "BA102742",
"ProductType": "Beverages",
"ProductName": "Coke",
"InDate": "21/3/20",
"OutDate": "25/3/20",
"VendorId": "SV10295",
"ProductQty": "20"}, {"OrderId": "Lan951247",
"ProductId": "DA10234",
"ProductType": "Dairy",
"ProductName": "Milk",
"InDate": "25/3/20",
"OutDate": "30/3/20",
"VendorId": "SV10265",
"ProductQty": "15"}]
```

Figure 15: Data stored in the database

DATABASE TESTING:

Database is one of the critical components of our web application and must be checked thoroughly. The response time of the queries are checked. The retrieved data from the database is shown accurately in our web application.

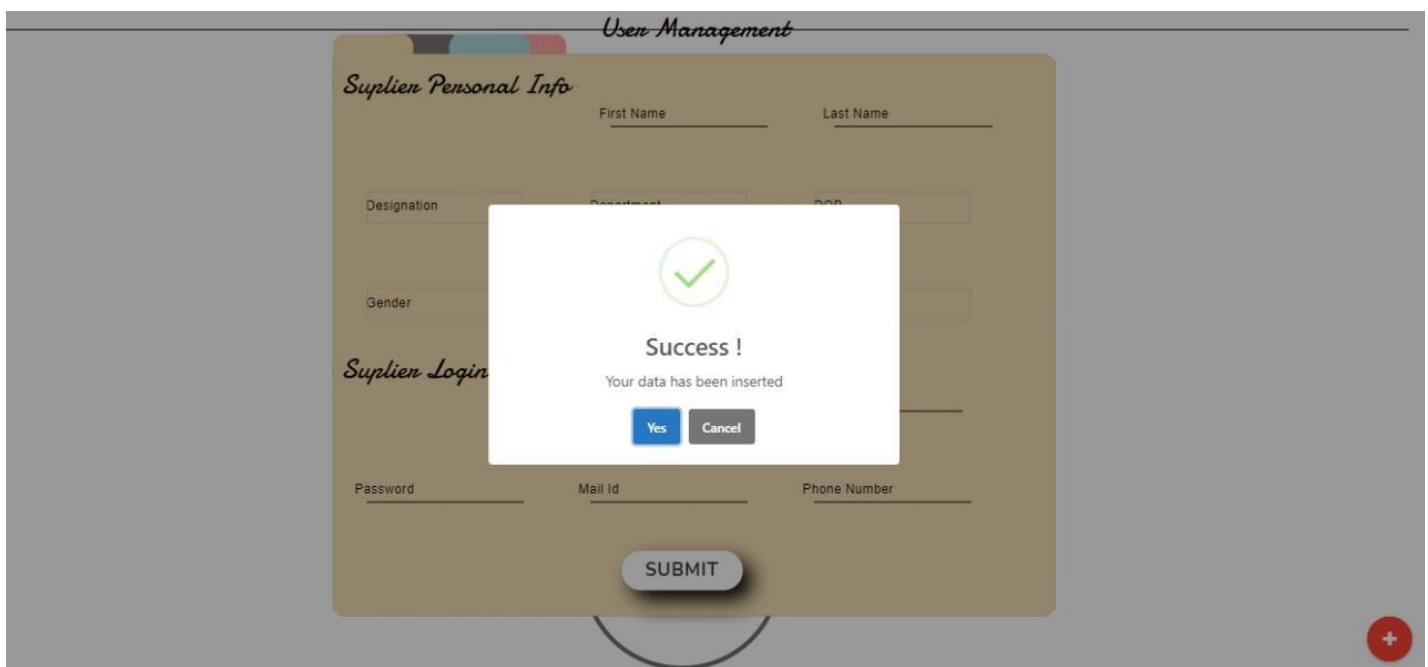
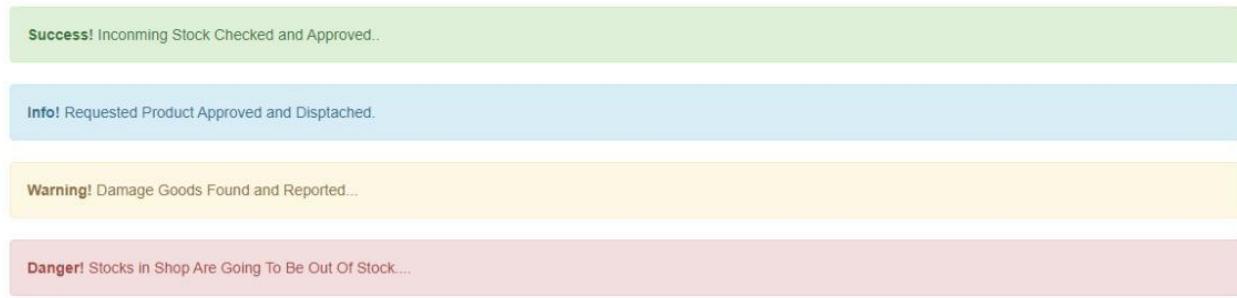


Figure 16: Data uploaded on user profile

Alerts



COMPATIBILITY TESTING:

Compatibility testing ensures that our web application displays correctly across different devices. The rendering of web elements like buttons, text fields etc. The website should work fine for various combination of operating systems such as Windows, Mac browsers such as Firefox, Internet Explorer.



Figure 17: Desktop view of Homepage - Footer



Figure 18: Mobile view of Homepage – Footer

8. REPORTS

8.1 SCREENSHOTS

INVENTORY MANAGEMENT FEATURES PAGE

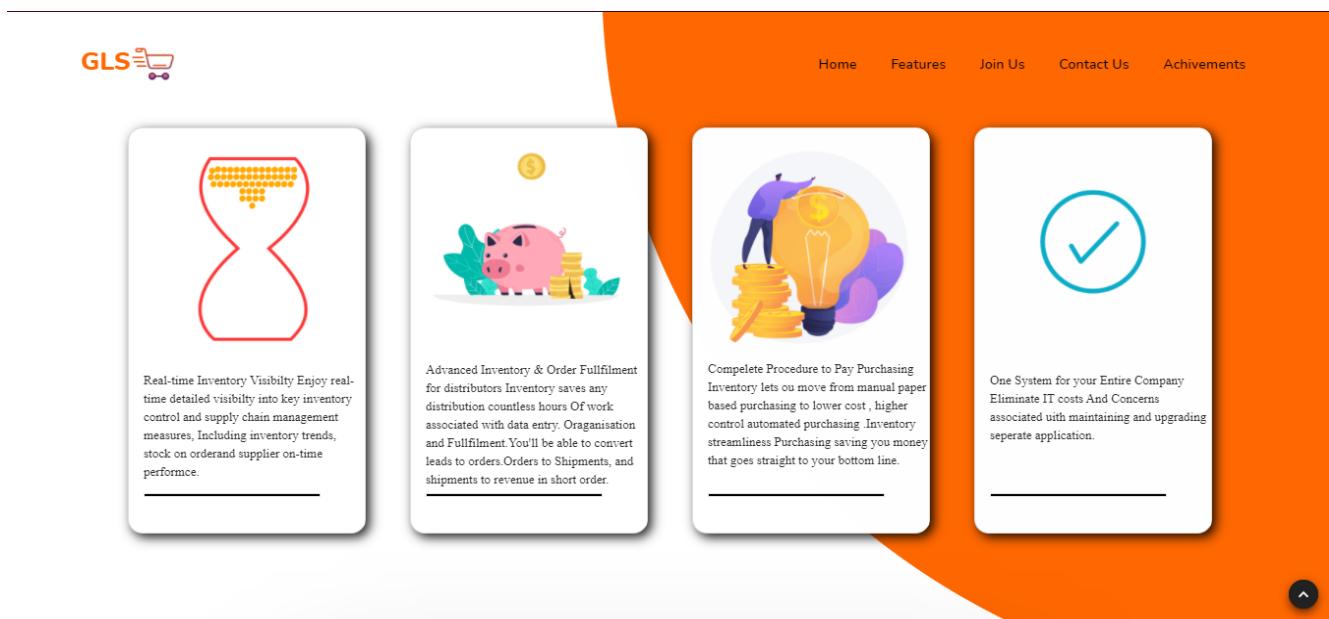


Figure 19: Feature's page

INVENTORY MANGEMENT JOIN US PAGE

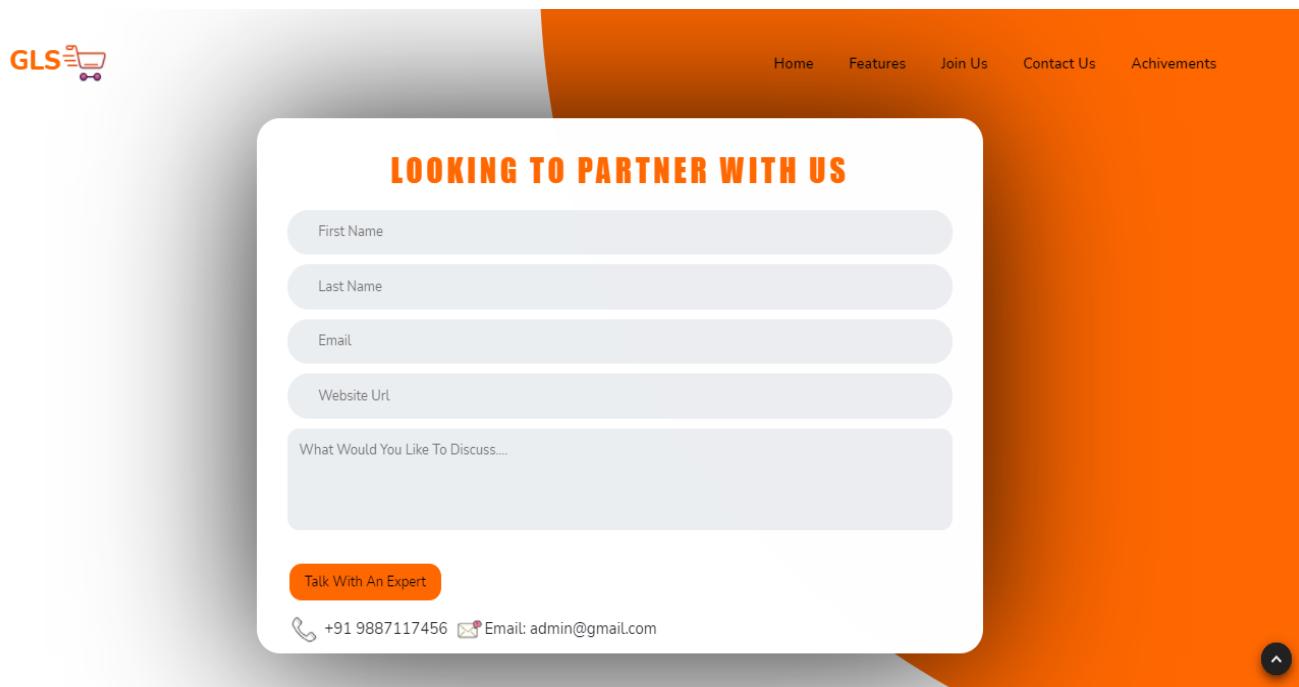


Figure 20: Join us page

INVENTORY MANGEMENT CONTACT US PAGE

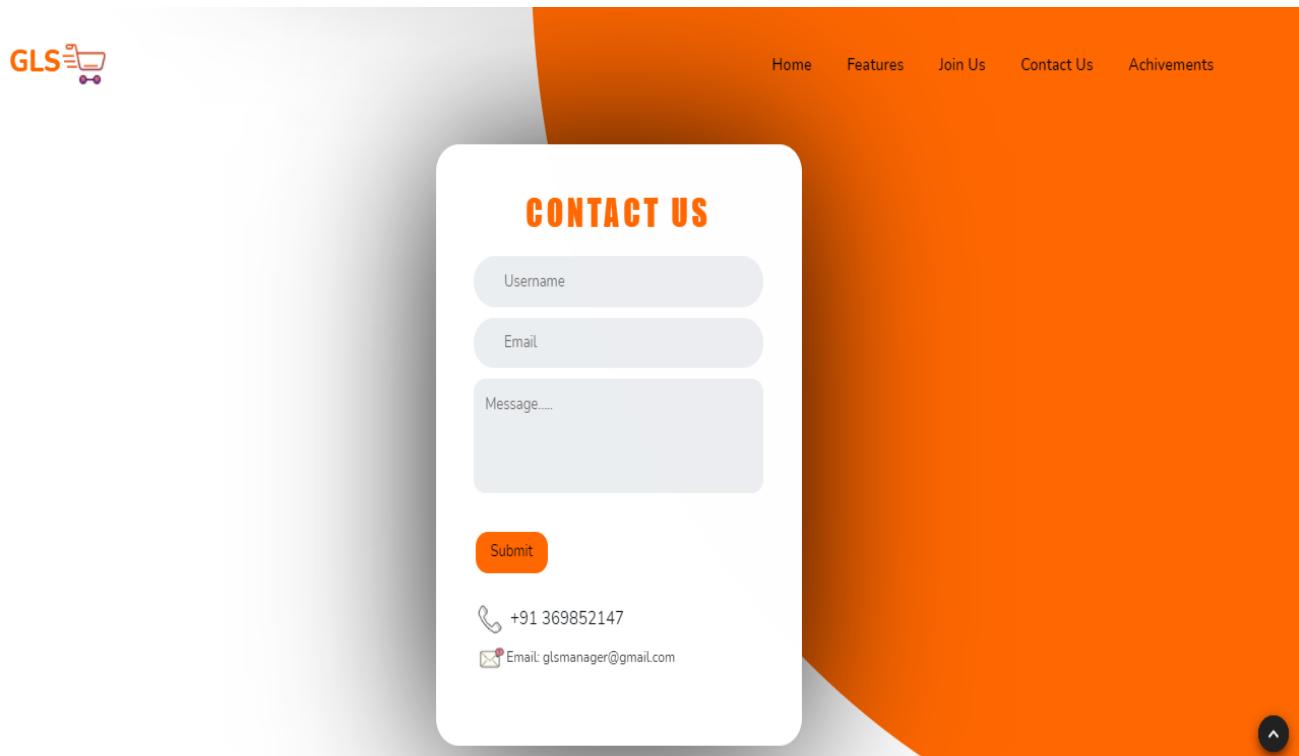


Figure 21: Contact us page

INVENTORY MANAGEMENT AWARDS PAGE



Figure 22: Award page

INVENTORY MANGEMENT FOOTER



Figure 23: Footer page

INVENTORY MANGEMENT ADMIN LOGIN FORM

A screenshot of an admin login form. The form has a light orange gradient background. It contains a title 'ADMIN LOGIN FORM' and two input fields: one for 'Latha' and another for a password represented by four dots ('....'). Below the password field is a 'Login' button. At the bottom of the form is a link 'Forgot Password ? Click here'. On the left side of the screen, there is a vertical navigation bar with a back arrow icon and the number '90'.

Figure 24: Admin Login Page

INVENTORY MANAGEMENT ADMIN DASHBOARD

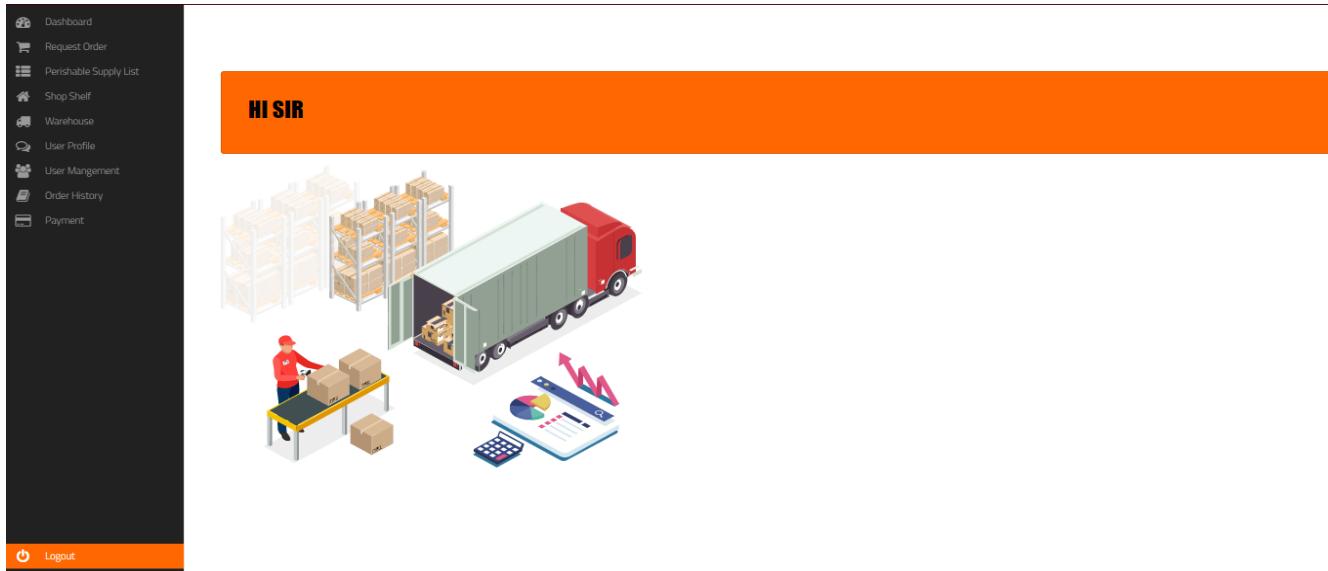


Figure 25: Admin Dashboard

INVENTORY MANAGEMENT PRODUCT REQUEST

A screenshot of the Product Request form. The title "Request Products" is displayed in a large orange header. The form includes dropdown menus for "Beverages" and "Soft Drinks", and input fields for "Vehicle No." (SV10232) and "Driver Name" (Kishore). A modal dialog box in the center asks "Do you want to save the changes?" with three buttons: "Save" (blue), "Don't save" (red), and "Cancel" (grey). At the bottom of the form are "Add" and "Submit" buttons, and a circular icon labeled "NEW". To the right, there is a red circular button with various icons: a magnifying glass, a person, a gear, a plus sign, a minus sign, and an X.

Figure 26: Product Request

INVENTORY MANAGEMENT AVAILABLE SHOP STOCK

Available Stock in Shop

OrderId	ProductId	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
Lan963214	FA10692	Fruits	Orange	09/6/20	10/6/20	SV10963	30kg
Lan158742	FG10234	Foodgrains	Wheat	30/5/20	05/6/20	SV10254	30Kg
Lan951234	DA10234	Dairy	Bread	21/5/20	22/5/20	SV10285	10

**Figure 27:
Available
shop stock**

Available Stock in Warehouse

OrderId	ProductId	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
Lan321456	FA10234	Fruits	Kiwi	12/3/20	16/3/20	SV10287	10Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan954876	BA102742	Beverages	Coke	21/3/20	25/3/20	SV10295	20
Lan951247	DA10234	Dairy	Milk	25/3/20	30/3/20	SV10265	15
Lan357894	CL10234	Cleaning	Floor Cleaner	30/3/20	02/4/20	SV10212	25
Lan963258	BP10234	Baby Products	Body Oil	09/4/20	12/4/20	SV10287	35
Lan852147	FG15987	FoodGrains	Rice	15/4/20	20/4/20	SV10285	25kg
Lan654789	FA10258	Fruits	Mango	21/4/20	24/4/20	SV10578	10kg



INVENTORY MANAGEMENT AVAILABLE STOCK IN WAREHOUSE

Figure 28 Available Stock in Warehouse

INVENTORY MANAGEMENT USER PROFILE

User Profile

FirstName	LastName	Designation	Department	DOB	Gender	PanId	AadharId	UserName	Id	Password	MailId	Phone
Srinath	narayan	Manager	Warehouse	12/12/1998	Male	WEA123456	3214 6542 6544	shri	MA12587469	Qwerty@123	waremanager@gamil.com	9632587412
Kishore	Kumar	Manager	Shop	22/10/1998	Male	JKA253456	3698 2587 1473	Kish	MA1236547892	Chennai@123	shopmanager@gamil.com	9512364785
Latta	Naidu	Supplier	Shop	14/08/1998	FeMale	MJA135789	3547 2587 1478	Latta	SP123654789	Zxy@1254	supplier@gamil.com	9965488745

Figure 29: User Profile

INVENTORY MANGEMENT PERISHABLE GOODS LIST

Perishable Goods List						
ProductId	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
FA10234	Fruits	Kiwi	12/3/20	16/3/20	SV10287	10Kg
FA36897	Fruits	Mango	18/3/20	12/3/20	SV12345	5kg
FA36987	Fruits	WaterMelon	16/8/20	20/4/20	SV12356	15kg
FA36856	Fruits	Orange	11/4/20	15/4/20	SV12369	50kg
VE36897	Vegetables	Brocoli	04/4/20	09/4/20	SV35321	5kg
VE96325	Vegetables	Potato	30/3/20	03/4/20	SV12856	9kg
VE98745	Vegetables	Onion	23/4/20	27/4/20	SV129886	8kg
VE98785	Vegetables	Tomato	28/4/20	01/5/20	SV129856	8kg
VE98795	Vegetables	Cabbages	05/5/20	09/5/20	SV129898	8kg
VE98963	Vegetables	Lady Finger	11/5/20	15/5/20	SV129874	20kg

VE36897	Vegetables	Brocoli	04/4/20	09/4/20	SV35321	5kg
VE96325	Vegetables	Potato	30/3/20	03/4/20	SV12856	9kg
VE98745	Vegetables	Onion	23/4/20	27/4/20	SV129886	8kg
VE98785	Vegetables				SV129856	8kg
VE98795	Vegetables				SV129898	8kg
VE98963	Vegetables				SV129874	20kg
FE98778	Fruits				SV129887	18kg
VE98745	Fruits				SV129875	28kg
VE987498	Vegetables	Carrot	24/5/20	230/5/20	SV129854	15kg
VE98752	Vegetables	Beans	03/6/20	10/6/20	SV129694	2kg
FE98796	Fruits	Apple	12/6/20	20/6/20	SV129354	18kg

Figure 30: Perishable goods list

INVENTORY MANGEMENT USER MANAGEMENT

User Management

Supplier Personal Info

First Name _____ Last Name _____

Designation _____ Department _____ DOB _____

Gender _____ Aadhar Number _____ Pan Id _____

Supplier Login Info

User Name _____ Supplier Id _____

Password _____ Mail Id _____ Phone Number _____

SUBMIT



Figure 31: Supplier's personal information

User Management

Vendor Personal Info

First Name _____ Last Name _____

Designation _____ Department _____ DOB _____

Gender _____ Aadhar Number _____ Pan Id _____

Vendor Login Info

User Name _____ Vendor Id _____

Password _____ Mail Id _____ Phone Number _____

SUBMIT



Figure 32: Vendor's personal information

User Management

Checker Personal Info

First Name	Last Name
Designation	Department
Gender	Aadhar Number
Pan Id	

Checker Login Info

User Name	Checker Id
Password	Mail Id
Phone Number	

SUBMIT



Figure 33: Checker's personal information

User Management

Manager Personal Info

First Name	Last Name
Designation	Department
Gender	Aadhar Number
Pan Id	

Manager Login Info

User Name	Manager Id
Password	Mail Id
Phone Number	

SUBMIT



Figure 34: Manager's personal information

INVENTORY MANAGEMENT PAYMENT DETAILS

Payment Details

VISA

Card Number	CVV
1234 1234 1234 1234	
Name	Complete Purchase
Your Name	
Expiration	
Mon ▾ Year ▾	

Figure 35: Payment details

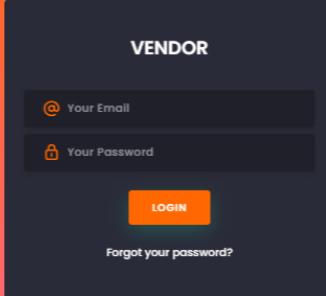
INVENTORY MANAGEMENT ORDER HISTORY

Order History						
Product Type	Product Name	Vendor Id	Vendor Name	Supplier Name	Product Description	Product Qty
Beverages	Beer	SV10287	Latta	Lokesh	Non-Perishable	10
Beverages	Pepsi	SV10287	Latta	Lokesh	Non-Perishable	20
Beverages	Coke	SV10287	Latta	Lokesh	Non-Perishable	30
Dairy	Bread	SV10268	Ram	Sham	Perishable	20
Dairy	Milk	SV10268	Ram	Sham	Perishable	25
Dairy	Panner	SV10268	Ram	Sham	Perishable	30
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg
Vegetables	Onion	SV10963	Ravi	Gopal	Perishable	25kg
Fruits	Kiwi	SV10963	Muthu	Shiva	Perishable	21kg
Fruits	Mango	SV10963	Muthu	Shiva	Perishable	30kg
Fruits	Orange	SV10963	Muthu	Shiva	Perishable	25kg



Figure 36: Order history

INVENTORY MANAGEMENT VENDOR LOGIN PAGE



The vendor login page features a dark-themed interface with orange accents. At the top, there are two tabs: "VENDOR" and "SUPPLIER". A central "LOG IN" button is flanked by "Forgot your password?" and "Create account" links. Below the button is a "Remember me" checkbox. The main area contains fields for "Your Email" and "Your Password", each with its respective icon.

Figure 37: Vendor Login Page

INVENTORY MANAGEMENT VENDOR DASHBOARD



Figure 38: Vendor Dashboard

INVENTORY MANAGEMENT VENDOR STOCK REQUEST

The image shows a "Stock Request" interface. At the top, a blue header bar displays the text "Stock Request". Below this is a white main area containing a table with data. The table has columns for Product Type, Product Name, Vendor Id, Vendor Name, Supplier Name, Product Description, and Product Qty. The data is as follows:

Product Type	Product Name	Vendor Id	Vendor Name	Supplier Name	Product Description	Product Qty
Beverages	Beer	SV10287	Latta	Lokesh	Non-Perishable	10
Beverages	Pepsi	SV10287	Latta	Lokesh	Non-Perishable	20
Beverages	Coke	SV10287	Latta	Lokesh	Non-Perishable	30
Dairy	Bread	SV10268	Ram	Sham	Perishable	20
Dairy	Milk	SV10268	Ram	Sham	Perishable	25
Dairy	Fanner	SV10268	Ram	Sham	Perishable	30
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg
Vegetables	Onion	SV10963	Ravi	Gopal	Perishable	25kg
Fruits	Kiwi	SV10963	Muthu	Shiva	Perishable	21kg
Fruits	Mango	SV10963	Muthu	Shiva	Perishable	30kg
Fruits	Orange	SV10963	Muthu	Shiva	Perishable	25kg

A blue circular navigation menu is located in the bottom left corner of the main area. It contains several icons, likely for navigating between different sections of the dashboard or performing specific actions like refresh or search.

Dairy	Bread	SV10268	Ram	Sham	Perishable	20
Dairy	Milk	SV10268	Ram	Sham	Perishable	25
Dairy	Panner	SV10268	Ram	Sham	Perishable	30
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg
Vegetables	Onion					25kg
Fruits	Kiwi					21kg
Fruits	Mango					30kg
Fruits	Orange					25kg
Foodgrains	Rice					30kg
Foodgrains	Wheat					35kg
Baby Products	Oil	SV10125	Shiny	Viji	Non-Perishable	25
Baby Products	Daiper	SV10125	Shiny	Viji	Non-Perishable	25

 Success !
Request Has Been Forwarded

[Yes](#) [Cancel](#)

FORWARD REQUEST TO SUPPLIER

Figure 39: Vendor Stock Request

5.3.4 INVENTORY MANGEMENT VENDOR DAMAGE STOCK REQUEST

Damage Stock Request

Product Type	Product Name	Vendor Id	Vendor Name	Supplier Name	Product Description	Product Qty	Damage Qty	Damage Qty
Beverages	Beer	SV10287	Latta	Lokesh	Non-Perishable	10	Yes	5
Beverages	Pepsi	SV10287	Latta	Lokesh	Non-Perishable	20	Yes	10
Beverages	Coke	SV10287	Latta	Lokesh	Non-Perishable	30	No	0
Dairy	Bread	SV10268	Ram	Sham	Perishable	20	Yes	15
Dairy	Milk	SV10268	Ram	Sham	Perishable	25	No	0
Dairy	Panner	SV10268	Ram	Sham	Perishable	30	Yes	20
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg	No	0
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg	Yes	5Kg

 Success !
Damage Stock Has Been Approved..

[Yes](#) [Cancel](#)

APPROVE

Figure 40: Vendor Stock Request

5.3.5 INVENTORY MANGEMENT VENDOR APPROVING REQUEST



Approved Stock									
Product Type	Product Name	Vendor Id	Vendor Name	Supplier Name	Product Description	Product Qty	Product Price	%	
Beverages	Beer	SV10287	Latta	Lokesh	Non-Perishable	10	360	20%	
Beverages	Pepsi	SV10287	Latta	Lokesh	Non-Perishable	20	300	50%	
Beverages	Coke	SV10287	Latta	Lokesh	Non-Perishable	30	250	250%	
Dairy	Bread	SV10268	Ram	Sham	Perishable	20	250	20%	
Dairy	Milk	SV10268	Ram	Sham	Perishable	25	200	10%	
Dairy	Panner	SV10268	Ram	Sham	Perishable	30	300	20%	
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg	300	40%	
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg	250	30%	
Vegetables	Onion	SV10963	Ravi	Gopal	Perishable	25kg	230	20%	
Fruits	Kiwi	SV10963	Muthu	Shiva	Perishable	21kg	500	15%	
Fruits	Mango	SV10963	Muthu	Shiva	Perishable	30kg	800	20%	
Fruits	Orange	SV10963	Muthu	Shiva	Perishable	25kg	600	15%	



Dairy	Milk	SV10268	Ram	Sham	Perishable	25	200	10%
Dairy	Panner	SV10268	Ram	Sham	Perishable	30	300	20%
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg	300	40%
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg	250	30%
Vegetables	Onion	SV10963	Ravi	Gopal	Perishable	25kg	230	20%
Fruits	Kiwi						500	15%
Fruits	Mango						800	20%
Fruits	Orange						600	15%
Foodgrains	Rice						400	20%
Foodgrains	Wheat						800	40%
Baby Products	Oil						300	30%
Baby Products	Daiper	SV10125	Shiny	Viji	Non-Perishable	25	200	10%


Success !
 Response Has Been Sent To Admin...

[Yes](#)
[Cancel](#)

LATHA

APPROVE

Figure 41: Vendor Approving Request

INVENTORY MANGEMENT SUPPLIER LOGIN PAGE

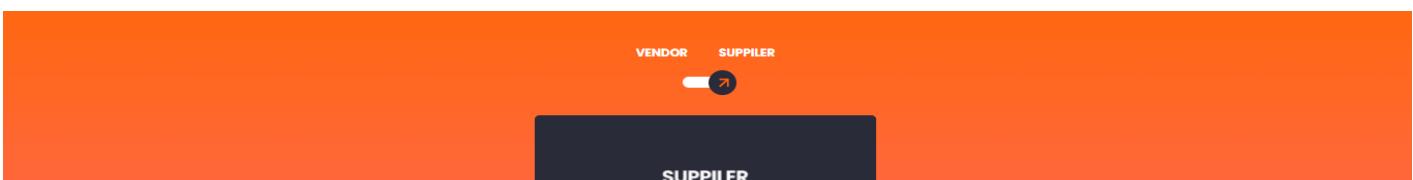


Figure 42: Supplier login page

INVENTORY MANGEMENT SUPPLIER LOGIN PAGE

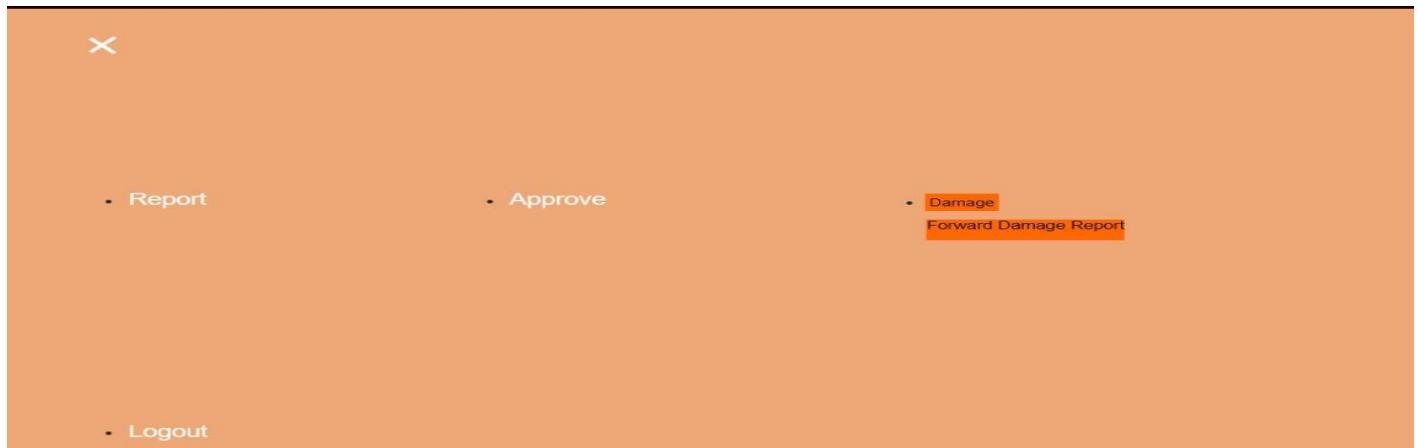


Figure 43: Supplier Dashboard page

INVENTORY MANGEMENT STOCK APPROVING

Approve Stock									
Product Type	Product Name	Vendor Id	Vendor Name	Supplier Name	Product Discription	Product Qty	Product Price	%	
Beverages	Beer	SV10287	Latta	Lokesh	Non-Perishable	10	360	20%	
Beverages	Pepsi	SV10287	Latta	Lokesh	Non-Perishable	20	300	50%	
Beverages	Coke	SV10287	Latta	Lokesh	Non-Perishable	30	250	250%	
Dairy	Bread	SV10268	Ram	Sham	Perishable	20	250	20%	
Dairy	Milk	SV10268	Ram	Sham	Perishable	25	200	10%	
Dairy	Fanner	SV10268	Ram	Sham	Perishable	30	300	20%	
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg	300	40%	
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg	250	30%	
Vegetables	Onion	SV10963	Ravi	Gopal	Perishable	25kg	230	20%	
Fruits	Kiwi	SV10963	Muthu	Shiva	Perishable	21kg	500	15%	
Fruits	Mango	SV10963	Muthu	Shiva	Perishable	30kg	800	20%	

Fruits	Kiwi	SV10963	Muthu	Shiva	Perishable	21kg	500	15%
Fruits	Mango	SV10963	Muthu	Shiva	Perishable	30kg	800	20%
Fruits	Orange	SV10963	Muthu	Shiva	Perishable	25kg	600	15%
Foodgrains	Rice	SV10963	Muthu	Shiva	Perishable	30kg	400	20%
Foodgrains	Wheat	SV10963	Muthu	Shiva	Perishable	35kg	800	40%
Baby Products	Oil	SV10963	Muthu	Shiva	Perishable	25	300	30%
Baby Products	Daiper	SV10963	Muthu	Shiva	Perishable	25	200	10%

Are you sure?

You won't be able to revert this!

No, Approve! Yes, Deny it!

Approve

Deny

INVENTORY MANAGEMENT DAMAGE REQUEST

Damage Stock Request

Product Type	Product Name	Vendor Id	Vendor Name	Supplier Name	Product Description	Product Qty	Damage Qty
Beverages	Beer	SV10287	Latta	Lokesh	Non-Perishable	10	Yes 5
Beverages	Pepsi	SV10287	Latta	Lokesh	Non-Perishable	20	Yes 10
Beverages	Coke	SV10287	Latta	Lokesh	Non-Perishable	30	No 0
Dairy	Bread	SV10268	Ram	Sham	Perishable	20	Yes 15
Dairy	Milk	SV10268	Ram	Sham	Perishable	25	No 0
Dairy	Panner	SV10268	Ram	Sham	Perishable	30	Yes 20

Are you sure?

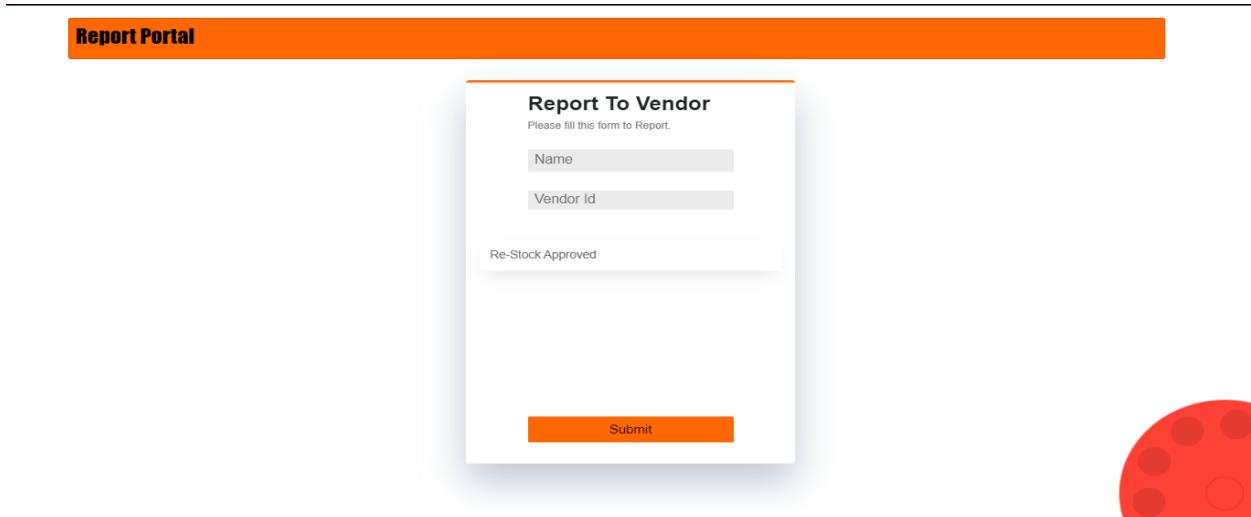
You won't be able to revert this!

Yes, Approve it! Cancel

Figure 44: Stock Approving

Figure 45: Damage Stock Request

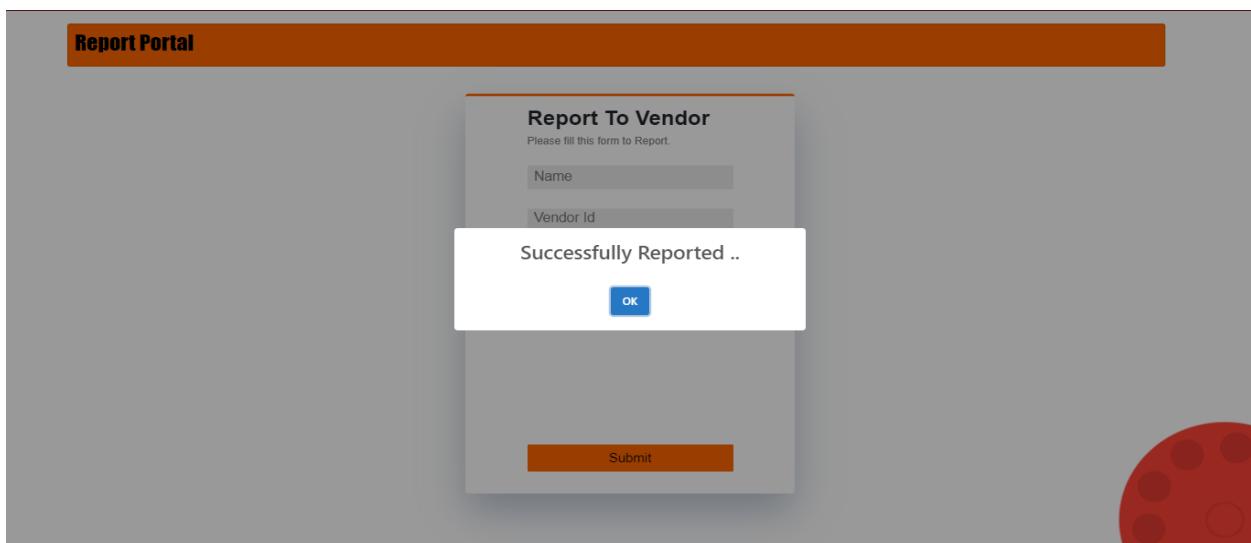
INVENTORY MANAGEMENT REPORT PORTAL



The screenshot shows a web-based reporting interface titled "Report To Vendor". At the top, a message says "Please fill this form to Report.". Below are two input fields: "Name" and "Vendor Id". A checkbox labeled "Re-Stock Approved" is present. At the bottom right is an orange "Submit" button.

Figure 46: Report Portal

INVENTORY MANAGEMENT REPORT PORTAL



This screenshot is similar to Figure 46, showing the "Report To Vendor" form. However, a modal dialog box is overlaid, displaying the message "Successfully Reported .." with an "OK" button. The original form elements like Name, Vendor Id, and Submit button are visible in the background.

Figure 47: Report Portal submitting

INVENTORY MANAGEMENT CHECKER LOGIN PAGE

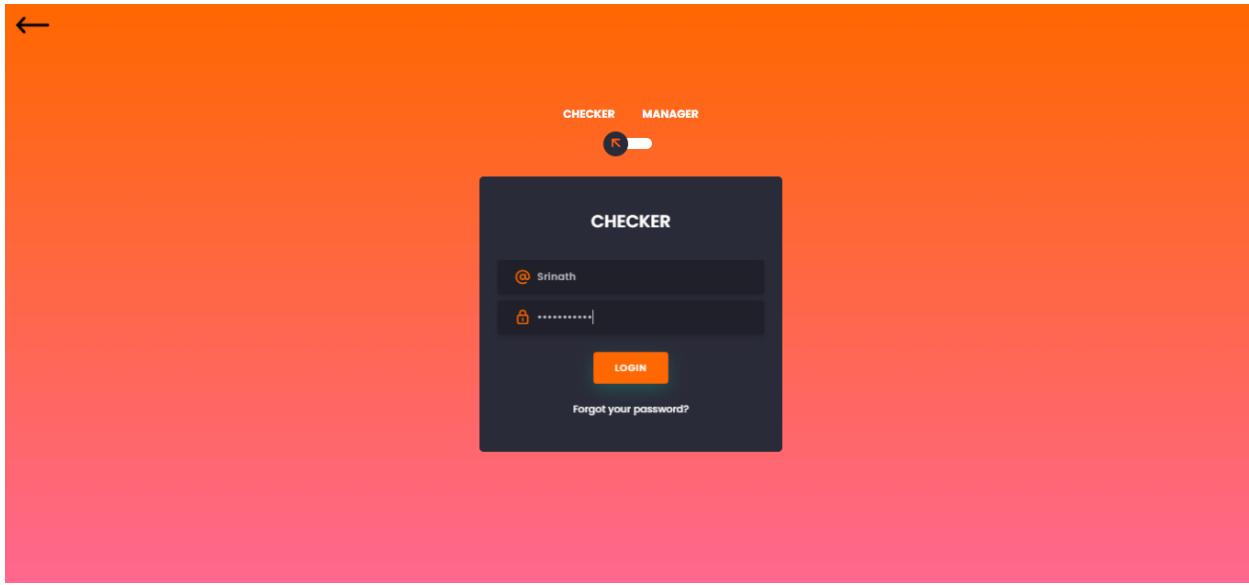


Figure 48: Checker Login

INVENTORY MANAGEMENT CHECKER DASHBOARD

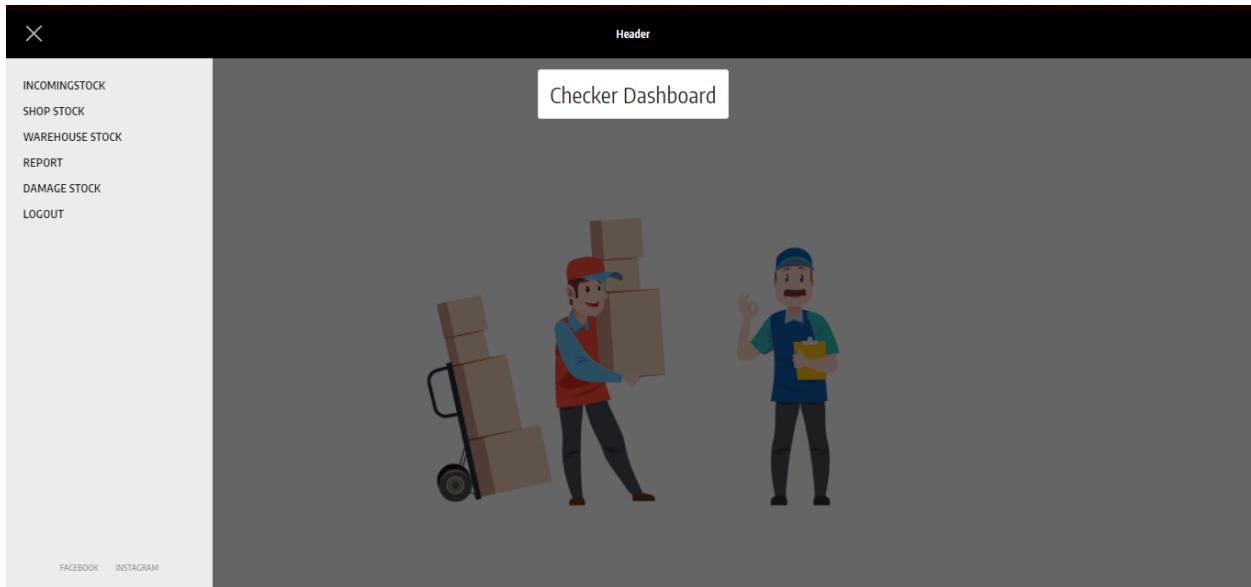


Figure 49: Checker Dashboard

INVENTORY MANAGEMENT CHECKER INCOMING STOCK

Incoming Stock

Product Type	Product Name	Vendor Id	Vendor Name	Supplier Name	Product Discription	Product Qty
Beverages	Beer	SV10287	Latta	Lokesh	Non-Perishable	10
Beverages	Pepsi	SV10287	Latta	Lokesh	Non-Perishable	20
Beverages	Coke	SV10287	Latta	Lokesh	Non-Perishable	30
Dairy	Bread	SV10268	Ram	Sham	Perishable	20
Dairy	Milk	SV10268	Ram	Sham	Perishable	25
Dairy	Panner	SV10268	Ram	Sham	Perishable	30
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg
Vegetables	Onion	SV10963	Ravi	Gopal	Perishable	25kg
Fruits	Kiwi	SV10963	Muthu	Shiva	Perishable	21kg
Fruits	Mango	SV10963	Muthu	Shiva	Perishable	30kg
Fruits	Orange	SV10963	Muthu	Shiva	Perishable	25kg
Foodgrains	Rice	SV10159	Kishore	Priya	Non-Perishable	30kg

The modal dialog box contains the following content:

- A green circular icon with a white checkmark.
- The text "Approved!" in bold.
- The message "Your Order has been Approved."
- A blue rectangular button labeled "OK".
- A grey rectangular button labeled "Cancel".

Dairy	Panner	SV10268	Ram	Sham	Perishable	30
Vegetables	Potato	SV10963	Ravi	Gopal	Perishable	20kg
Vegetables	Tomato	SV10963	Ravi	Gopal	Perishable	25kg
Vegetables	Onion	SV10963	Ravi	Gopal	Perishable	25kg
Fruits	Kiwi	SV10963	Muthu	Shiva	Perishable	21kg
Fruits	Mango					30kg
Fruits	Orange					25kg
Foodgrains	Rice					30kg
Foodgrains	Wheat					35kg
Baby Products	Oil					25
Baby Products	Daiper					25

Approve **Deny**

Figure 50: Checker's Incoming Stock

INVENTORY MANAGEMENT AVAILABLE STOCK IN SHOP

Available Stock in Shop

OrderId	ProductId	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
Lan963214	FA10692	Fruits	Orange	09/6/20	10/6/20	SV10963	30kg
Lan158742	FG10234	Foodgrains	Wheat	30/5/20	05/6/20	SV10254	30Kg
Lan951234	DA10234	Dairy	Bread	21/5/20	22/5/20	SV10285	10
Lan763159	CL10287	Cleaning	Handwash	18/5/20	20/5/20	SV10225	25
Lan456321	BP10234	Baby Products	Diapers	10/5/20	12/5/20	SV10963	40
Lan956874	BA10852	Beverages	Pepsi	02/5/20	05/5/20	SV10254	30
Lan123965	VE10234	Vegetables	Patato	26/4/20	30/4/20	SV10951	20Kg
Lan654789	FA10258	Fruits	Mango	21/4/20	24/4/20	SV10578	10kg
Lan852147	FG15987	FoodGrains	Rice	15/4/20	20/4/20	SV10285	25kg

Figure 51: Stock in Shop

INVENTORY MANGEMENT AVAILABLE STOCK IN SHOP

Available Stock in WareHouse

OrderId	ProductId	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
Lan321456	FA10234	Fruits	Kiwi	12/3/20	16/3/20	SV10287	10Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan954876	BA102742	Beverages	Coke	21/3/20	25/3/20	SV10295	20
Lan951247	DA10234	Dairy	Milk	25/3/20	30/3/20	SV10265	15
Lan357894	CL10234	Cleaning	Floor Cleaner	30/3/20	02/4/20	SV10212	25
Lan963258	BP10234	Baby Products	Body Oil	09/4/20	12/4/20	SV10287	35
Lan852147	FG15987	FoodGrains	Rice	15/4/20	20/4/20	SV10285	25kg
Lan654789	FA10258	Fruits	Mango	21/4/20	24/4/20	SV10578	10kg

Figure 52: Stock in Shop

INVENTORY MANAGEMENT AVAILABLE STOCK IN WAREHOUSE

Available Stock in WareHouse

OrderId	Productid	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
Lan321456	FA10234	Fruits	Kiwi	12/3/20	16/3/20	SV10287	10Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan954876	BA102742	Beverages	Coke	21/3/20	25/3/20	SV10295	20
Lan951247	DA10234	Dairy	Milk	25/3/20	30/3/20	SV10265	15
Lan357894	CL10234	Cleaning	Floor Cleaner	30/3/20	02/4/20	SV10212	25
Lan963258	BP10234	Baby Products	Body Oil	09/4/20	12/4/20	SV10287	35
Lan852147	FG15987	FoodGrains	Rice	15/4/20	20/4/20	SV10285	25kg
Lan654789	FA10258	Fruits	Mango	21/4/20	24/4/20	SV10578	10kg

Figure 53: Stock in Warehouse

INVENTORY MANAGEMENT REPORT PORTAL

Report Portal

Report To Manager
Please fill this form to Report.

Name

Checker Id

Re-Stock Stored

Submit

Figure 54: Report Portal

INVENTORY MANAGEMENT REPORT PORTAL

Report Portal

Report To Manager
Please fill this form to Report.

Name

Checker Id

Successfully Reported ..

OK

Figure 55: Submitting Report Portal

INVENTORY MANGEMENT MANAGER LOGIN PAGE

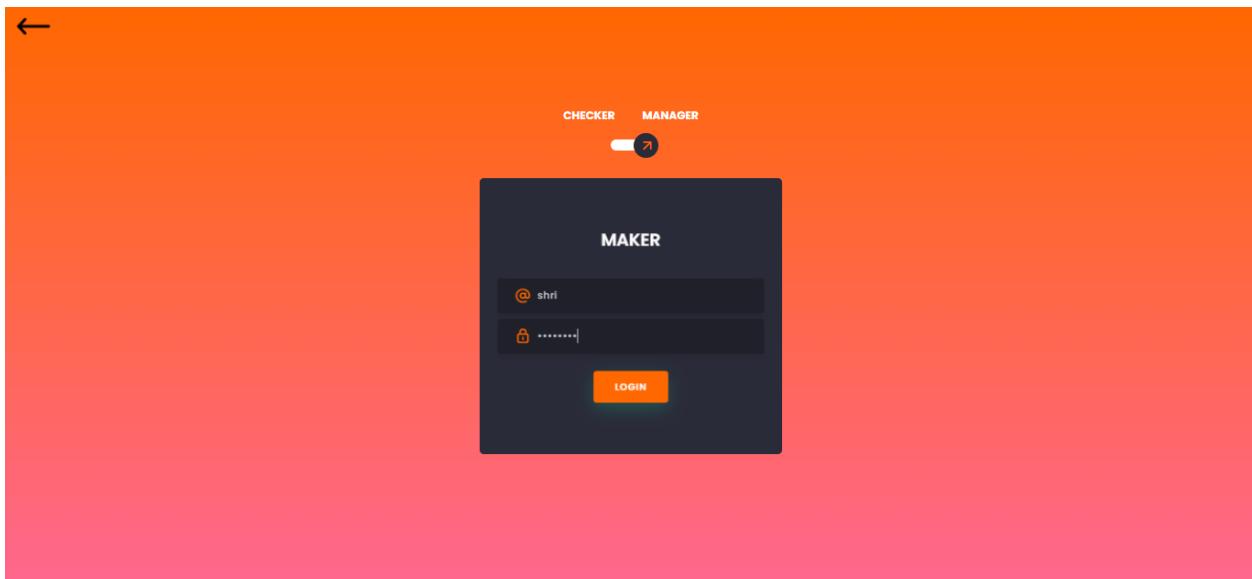


Figure 56: Submitting Report Portal

INVENTORY MANGEMENT MANAGER DASHBOARD PAGE

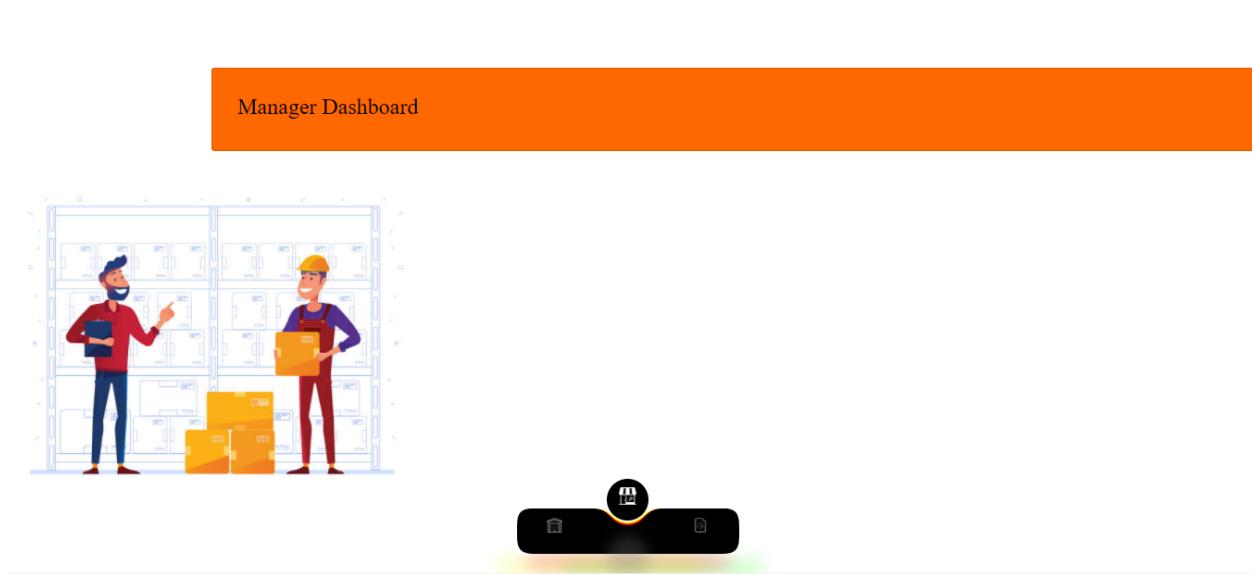


Figure 57: Manager Dashboard

INVENTORY MANGEMENT MANAGER -AVAILABILITY OF SHOP STOCK

Available Stock in Shop

OrderId	ProductId	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
Lan963214	FA10692	Fruits	Orange	09/6/20	10/6/20	SV10963	30kg
Lan150742	FG10234	Foodgrains	Wheat	30/5/20	05/6/20	SV10254	30Kg
Lan951234	DA10234	Dairy	Bread	21/5/20	22/5/20	SV10285	10
Lan753159	CL10287	Cleaning	Handwash	18/5/20	20/5/20	SV10225	25
Lan456321	BP10234	Baby Products	Diapers	10/5/20	12/5/20	SV10963	40
Lan999874	BA10852	Beverages	Pepsi	02/5/20	05/5/20	SV10254	30
Lan123965	VE10234	Vegetables	Patato	26/4/20	30/4/20	SV10951	20Kg
Lan654788	FA10258	Fruits	Mango	21/4/20	24/4/20	SV10578	10kg
Lan852147	FG15987	FoodGrains	Rice	15/4/20	20/4/20	SV10285	25kg



Figure 58: Shop Stock

INVENTORY MANGEMENT MANAGER -AVAILABILITY OF WAREHOUSE STOCK

Available Stock in WareHouse

OrderId	ProductId	ProductType	ProductName	InDate	OutDate	VendorId	ProductQty
Lan321456	FA10234	Fruits	Kiwi	12/3/20	16/3/20	SV10287	10Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan753357	VE10234	Vegetables	Tomato	15/3/20	20/3/20	SV10298	15Kg
Lan954876	BA102742	Beverages	Coke	21/3/20	25/3/20	SV10295	20
Lan951247	DA10234	Dairy	Milk	25/3/20	30/3/20	SV10265	15
Lan357894	CL10234	Cleaning	Floor Cleaner	30/3/20	02/4/20	SV10212	25
Lan963258	BP10234	Baby Products	Body Oil	09/4/20	12/4/20	SV10287	35
Lan852147	FG15987	FoodGrains	Rice	15/4/20	20/4/20	SV10285	25kg
Lan654789	FA10258	Fruits	Mango	21/4/20	24/4/20	SV10578	10kg



Figure 59: Warehouse Stock

INVENTORY MANGEMENT MANAGER REPORT PORTAL

Report To Admin
Please fill this form to Report.

Name

Manager Id

Re-Stock Stored

Submit

**Figure 60 Report portal**

Report Portal

Report To Admin
Please fill this form to Report.

Name

Manager Id

Successfully Reported ..

OK

Submit

**Figure 61: Reported Portal**

9. ENHANCEMENTS

The Inventory Management for a E-mart Grocery can be designed with much more upgraded features in the future. The website has been developed to overcome the manual entry of content or status. The loading speed of Webpages can be increased which will provide a clear visibility of the website. In future Bar code generation can be implemented (i.e.) The stickers that will be pasted on the items, Invoice would be generated as PDF files (signed) and sent through email

to the corresponding distributors or seller, Alerts to be send as Email as part of phase 1 and in phase 2 we will add SMS to it.

By providing these upgraded features our website would become more users friendly and compatible to all kind of devices.

10.CONCLUSION

This internship at the SRM Technologies was a very beneficial experience. I was extremely honored to get the opportunity to work as an intern for this organization. I have gotten used to the workspace and have enjoyed my time there. As my internship with SRM Technologies comes to an end, I am glad to say that those days were well spent. I have accomplished my own goals and the team at SRM Technologies has exceeded my expectations.

Going into this experience was something completely new for me, but with their help I have learned to work with real time application. I have learned multiple strategies that the company uses to promote a business or to make the service more appealing. During my time here, I have realized that a big part of designing a module is creating quick and easy ways to access the service.

The things that I have learned here will help me in the future IT career. I am very grateful for this experience and all the ways that it will beneficiary. This opportunity has given me great experience.

11. LIST OF TABLES

S.NO	TABLE NO	TABLE NAME	PAGE NO
1	3.2.1	admin_request	12
2	3.2.2	user_profile	12
3	3.2.3	Shop_stock	13
4	3.2.4	Perishablegoods	13
5	3.2.5	Warehouse_stock	14
6	3.2.6	Damage_stock	14
7	3.2.7	Approve Table	15

12. LIST OF FIGURES

S.NO	FIGURE NO	FIGURE NAME	PAGE NO
1	1	Admin - Use Case Diagram	28
2	2	Vendor- Use Case Diagram	29
3	3	Warehouse- Use Case Diagram	30
4	4	Overall Architecture	42
5	5	Home page	43
6	6	Admin Dashboard	44
7	7	Vendor Dashboard	44
8	8	Supplier Dashboard page	45
9	9	Checker Dashboard	45
10	10	Manager Dashboard	46
10	10	Inventory Management	74
11	11	Navigation menu on Vendor Dashboard	75
12	12	Product Type in Dropdown	76
13	13	Vendor's Response Storing in database	76
14	14	Data stored in the database	76

15	15	Data uploaded on user profile	77
----	----	-------------------------------	----

16	16	Notification alert	77
17	17	Desktop view of Homepage - Footer	78
18	18	Mobile view of Homepage – Footer	78
19	19	Feature's page	79
20	20	Join us page	79
21	21	Contact us page	80
22	22	Award page	80
23	23	Footer page	81
24	24	Admin Login Page	81
25	25	Admin Dashboard	81
26	26	Product Request	82
27	27	Available shop stock	82
28	28	Available Stock in Warehouse	83
29	29	User Profile	83
30	30	Perishable goods list	83
31	31	Supplier's personal information	84
32	32	Vendor's personal information	84
33	33	Checker's personal information	85

34	34	Manager's personal information	85
35	35	Payment details	85
36	36	Order history	86
37	37	Vendor Login Page	86
38	38	Vendor Dashboard	86
39	39	Vendor Stock Request	87
40	40	Vendor Stock Request	88
41	41	Vendor Approving Request	88
42	42	Supplier login page	89
43	43	Supplier Dashboard page	90
44	44	Stock Approving	90
45	45	Damage Stock Request	90
46	46	Report Portal	91
47	47	Report Portal submitting	91
48	48	Checker Login	92
49	49	Checker Dashboard	92
50	50	Checker's Incoming Stock	93
51	51	Stock in Shop	93

52	52	Stock in Warehouse	94
53	53	Report Portal	94
54	54	Submitting Report Portal	95
55	55	Manager Login Page	95
56	56	Manager Dashboard	96
57	57	Shop Stock	96
58	58	Warehouse Stock	97
59	59	Report portal	97
60	60	Reported Portal	97

13. REFERENCES

- [1] “animation,” *Mozilla.org*. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS/animation>. [Accessed: 08-May-2021].
- [2] “Angular Material,” *Angularjs.org*. [Online]. Available: <https://material.angularjs.org/1.1.1/layout/container>. [Accessed: 08-May-2021].
- [3] W. Le, “A simple way to animate scrolling effects in JavaScript,” *Alligator.io*. [Online]. Available: <https://alligator.io/js/smooth-scrolling/>. [Accessed: 08-May-2021].
- [4] K. Chinnathambi, “Creating a smooth sliding menu,” *Kirupa.com*. [Online]. Available: https://www.kirupa.com/html5/creating_a_smooth_sliding_menu.htm. [Accessed: 08-May-2021].
- [5] “Angular 4 pass data between 2 not related components,” *Stackoverflow.com*. [Online]. Available: <https://stackoverflow.com/questions/44414226/angular-4-pass-data-between-2-not-related-components>. [Accessed: 08-May-2021].
- [6] G. Karthik, “35 CSS card flip animations for communicating quick stories,” *Uicookies.com*, 27-Mar-2021. [Online]. Available: <https://uicookies.com/css-card-flip/>. [Accessed: 08-May-2021].
- [7] R. Karanam, “Connecting Spring Boot with databases - MySQL and Oracle,” *Springboottutorial.com*. [Online]. Available: <https://www.springboottutorial.com/spring-boot-with-mysql-and-oracle>. [Accessed: 08-May-2021].
- [8] S. Maletta, “Transfer large files using a rest API - the Startup - medium,” *The Startup*, 17-Oct-2020. [Online]. Available: <https://medium.com/swlh/transfer-large-files-using-a-rest-api-a0aa96983ebb>. [Accessed: 08-May-2021].
- [9] A. Jacobson, “Working with multiple databases in spring,” *InfoQ*, 30-Jul-2016. [Online]. Available: <https://www.infoq.com/articles/Multiple-Databases-with-Spring-Boot/>. [Accessed: 08-May-2021].
- [10] E. Paraschiv, “Introduction to Spring Data JPA,” *Baeldung.com*, 24-May-2014. [Online]. Available: <https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa>. [Accessed: 08-May-2021].
- [11] “Spring Boot,” *Geeksforgeeks.org*, 09-Jun-2020. [Online]. Available: <https://www.geeksforgeeks.org/spring-boot-how-to-access-database-using-spring-data-jpa/>. [Accessed: 08-May-2021].
- [12] Kevin, “Read data from database with spring data JPA in spring MVC,” *Learningprogramming.net*, 06-Jul-2018. [Online]. Available:

<https://learningprogramming.net/java/spring-mvc/read-data-from-database-with-spring-data-jpa-in-spring-mvc/>. [Accessed: 08-May-2021].

- [13] S. F. G. Contributor, “How to configure multiple data sources in a Spring Boot application - spring framework guru,” *Springframework.guru*, 31-May-2019. [Online]. Available: <https://springframework.guru/how-to-configure-multiple-data-sources-in-a-spring-boot-application/>. [Accessed: 08-May-2021].
- [14] J. Freeman, “What is JSON? A better format for data exchange,” *Infoworld.com*, 25-Oct-2019. [Online]. Available: <https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html>. [Accessed: 08-May-2021].
- [15] E. Elrom, “AngularJS,” in *Pro MEAN Stack Development*, Berkeley, CA: Apress, 2016, pp. 101–129.
- [16] mayankjtp, “Angular 8 Error Fixing,” *Tutorialandexample.com*, 14-Aug-2019. [Online]. Available: <https://www.tutorialandexample.com/angular-8-error-fixing/>. [Accessed: 08-May-2021].
- [17] “Angular,” *Angular.io*. [Online]. Available: <https://angular.io/guide/reactive-forms>. [Accessed: 08-May-2021].
- [18] “Template-driven forms,” *Codecraft.tv*. [Online]. Available: <https://codecraft.tv/courses/angular/forms/template-driven/>. [Accessed: 08-May-2021].
- [19] “Spring boot crud operations example with hibernate,” *Howtodoinjava.com*, 05-Jun-2019. [Online]. Available: <https://howtodoinjava.com/spring-boot2/spring-boot-crud-hibernate/>. [Accessed: 08-May-2021].
- [20] “Angular file upload with progress,” *Nils-mehlhorn.de*, 01-Feb-2021. [Online]. Available: <https://nils-mehlhorn.de/posts/angular-file-upload-progress>. [Accessed: 08-May-2021].
- [21] Y. Goudar, “Property binding in angular,” *Pluralsight.com*. [Online]. Available: <https://www.pluralsight.com/guides/property-binding-angular>. [Accessed: 08-May-2021].
- [22] C. Nwamba, “Understanding Angular property binding and interpolation,” *Telerik.com*, 03-Jun-2019. .