

Inventory Management and Business Intelligence in Yarn Trading

CSE4015 - Human Computer Interaction

Final Project Report

J Component

Submitted By:

Sanjit C K S - 18BCE0715

In partial fulfilment of the degree of B.tech

In

Computer Science and Engineering



Vellore - 632014, Tamil Nadu, India

School of Computer Science and Engineering

Fall Semester

October 2020

Problem Statement

Most small scale trading companies and manufacturing factories, especially in the textile industry lack the technological means for any kind of business intelligence.

Abstract

Textile trading is a million dollar industry in central Tamil Nadu. But it is composed of several small and medium scale organisations instead of huge corporates. At the same time, most small scale trading companies and manufacturing factories, in the textile industry lack the technological means for any kind of business intelligence. Generation of market insight with existing data is key for a booming business. The aim of the project is to create a simple web tool for a textile trading company that empowers them with business intelligence while also allowing them to reach more potential clients.

T Prevalent situation - trading companies of small/medium scale that cannot afford sophisticated customisable software that exists in the market. (Erode - Tiruppur - Coimbatore). Need is a relatively simpler application which can be easily built with the prevalent technology. This opens up a potential market for simpler web tools which provide business intelligence.

Keywords

Yarn Trading Company, Full Stack Web Application, Inventory Management, Business Intelligence, Visualisations, ReactJS, NodeJS, MongoDB

Introduction

Aim:

In small scale textile companies, Inventory details are mostly stored via generic accounting software or even worse, handwritten documents. Sending samples of yarn or fabric manually via post to client base wastes resources. It is also wasted potential data for business intelligence. Through this web application we aim to create a web application where potential customers can look at the catalog of available end products, and inquire the company owners about cost or ask for a quote via the website. The catalog of inventory items can be managed and updated constantly by employees with an employee login. Another important feature, would be the data will be visualised with appropriate graphs for an admin user for business intelligence purposes.

Scope:

This is a very prevalent situation in trading companies of small scale that cannot afford sophisticated customisable software that exists in the market. What these companies need is a relatively simpler application which can be easily built with the prevalent technology. This opens up a potential market for simpler web tools which provide business intelligence. These are companies that have *just* enough financial means to pay for a reasonable software but do not have the exposure or means to access such technology. Business insight is usually gained by information gathering and word of mouth. This is not always reliable. With proper marketing, by gaining a few clients it is possible to make this a successful business venture.

So the overall Objectives can be summarised as:

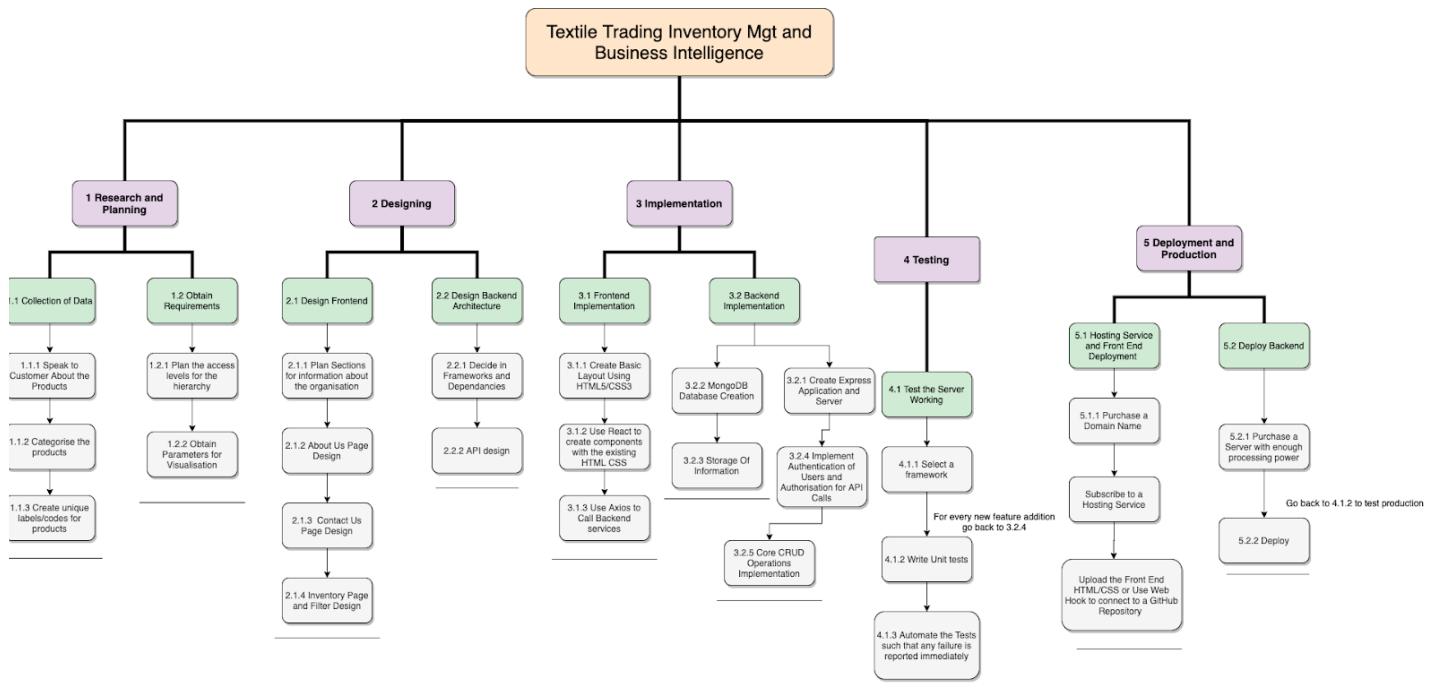
1. Product Inventory Maintenance - Creating, Viewing, Updating and Deleting Products by Employee Users
2. Inquiry Creation - Creating and sending Inquiries by potential customers on the public section
3. Business Intelligence - Creating graphs, charts and visualisations with Inquiry data (Demand Insights Generation) by the Admin User. Creation and Deletion of employee accounts

The process model chosen is **Evolutionary Prototype Model**. Here are the reasons why I think it could be a good choice for this particular project.

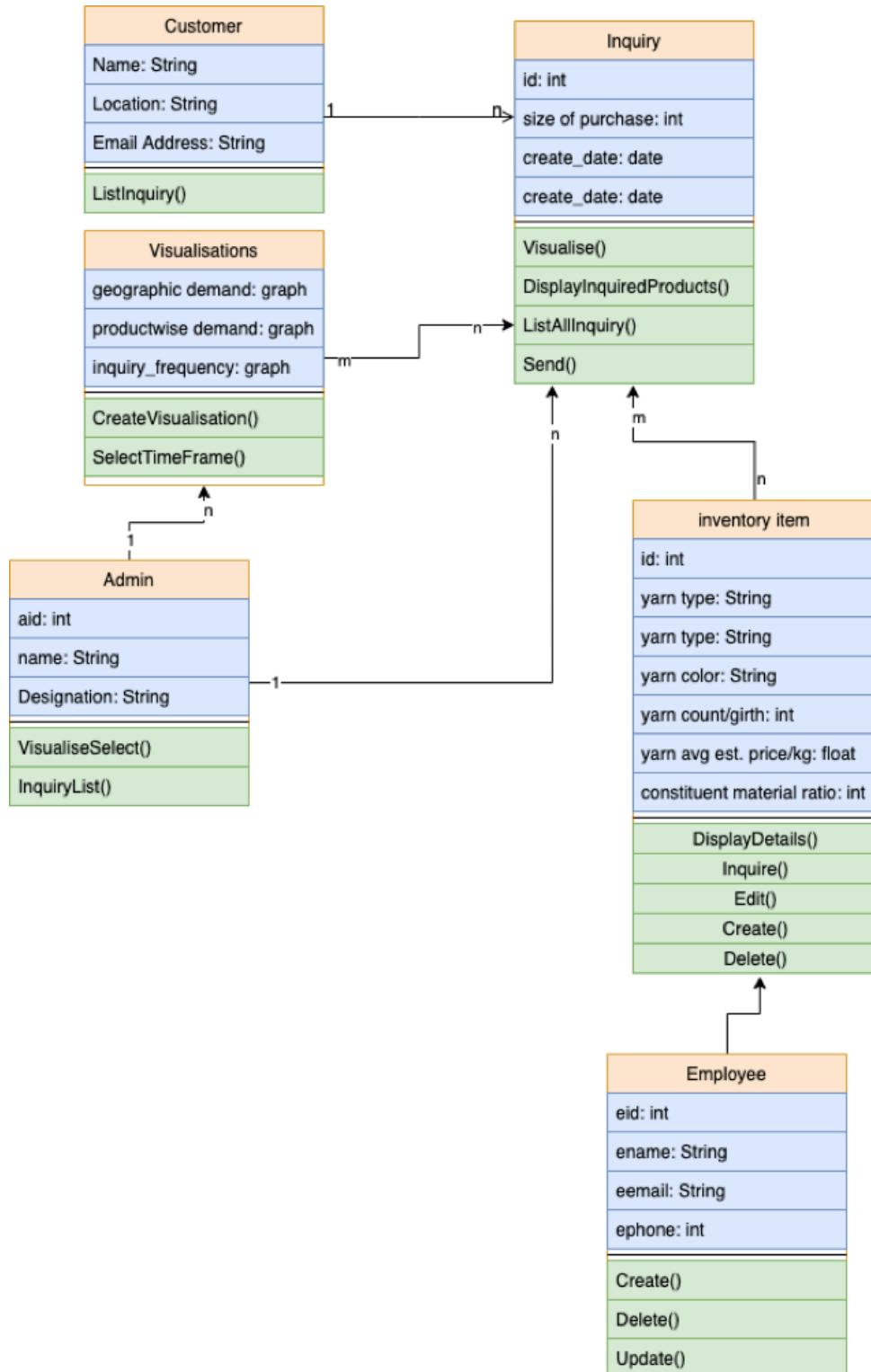
Prototype models are highly recommended for web interfaces that require customer feedback. Since the functionalities can also be modularised to some degree, features can be implemented quickly for a prototype.

Improvements to the design and UI-UX can be made on the go after positive criticism of the initial prototype. Although it might take some time to build the final version, this model allows ease of use and the end-users need minimal training. Prototyping ensures that the end users constantly work with the system and provide a feedback which is incorporated in the prototype to result in a useable system. They are excellent for designing good human computer interface systems. Also, it allows for modularisation to some degree that it can be build in chunks and tested before integration and deployment.

Hierarchical Technical Analysis



Class Diagram



Interaction Design Models

Primary Stakeholders

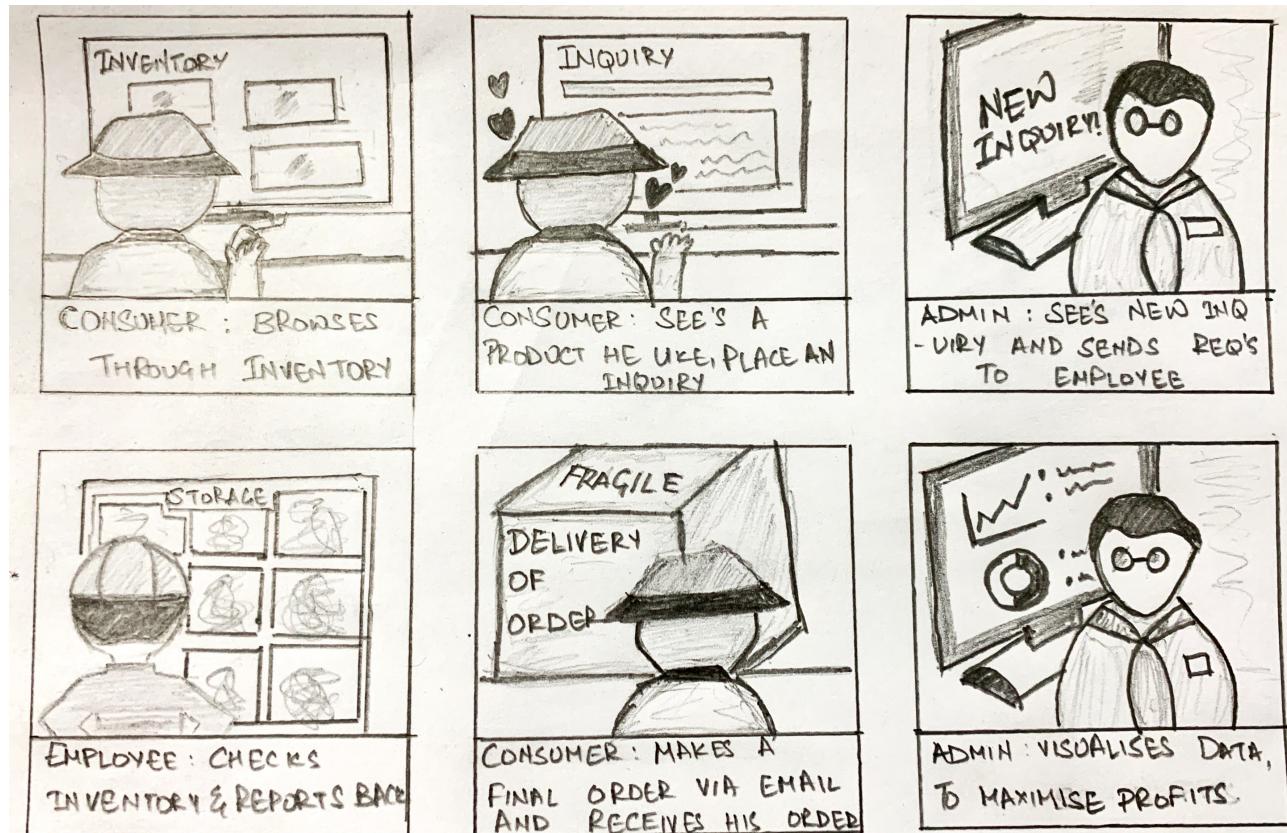
The stakeholders are doing to be small-scale textile trading companies. This could include Yarn Trading Companies, Fabric Traders (Bedsheets, Clothes etc) and weavers. The stakeholders studied for this particular project are yarn trading companies and small scale textile manufacturers in the Erode and Tiruppur region. This region quite well known for its textile manufacturers. *Kumar Yarns, Chennimalai* is one such yarn trading company studied for this particular project.

End Users: Potential Customers, Employees and Managers/Owners (Admins) in Small Scale Trading Companies. Employees use it for inventory maintenance. Admins use it for Inquiry response and marketing insight. (**Direct users**)

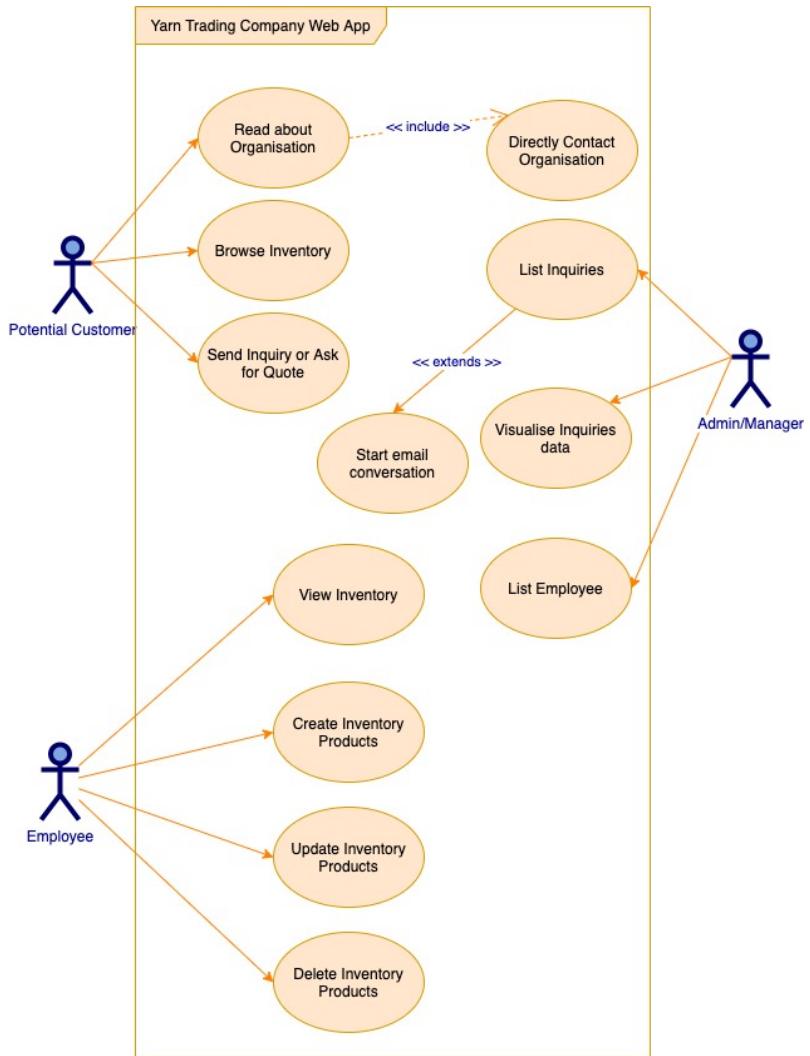
Transport agencies are secondary users who do not interact directly with the software but benefit from the publicly available inventory and status. (**Secondary Users**)

Beneficiaries General public and overall supply chain of the textile industry. The transparency encourages trust.

Story Boarding



Use Case Diagram



GOMS - Goals, Operations, Methods, Selections

GOAL: The Goal Is To Choose A Fabric Of The Users Selection And Make An Inquiry Regarding It

OPERATORS: Browse through inventory page and select A product

- Make inquiry for selected product
- Send purchase request on inquiry confirmation
- Send email to purchase order
- Send email to make inquiry

METHODS:

METHOD 1:

- Browse through inventory page and select A product
- Make inquiry for selected product
- Send purchase request on inquiry confirmation

METHOD 2:

- Send email to make inquiry
- Send email to purchase order

SELECTIONS:

SELECTION 1: User must navigate to the products page, where they can select a product, make an inquiry and click on send inquiry

SELECTION 2: User must navigate to the home page, where they can copy company Email ID, make an inquiry via E-mail.

Sample Code

Snippet of Product Operations and Management

```
const express = require('express');
const multer = require('multer');
const sharp = require('sharp');
const bodyParser = require('body-parser')
const Product = require('../models/product.js')
const {auth, adminAuth} = require("../middleware/auth")
const urlencodedParser = bodyParser.urlencoded({ extended:false})
const router = express.Router()
```

```
const upload = multer({
    limits: {
        fileSize: 5000000
    },
    fileFilter(req,file,cb) {
        if (!file.originalname.match(/\.(jpg|jpeg|png|JPG|PNG|JPEG)$/)) {
            return cb(new Error("Upload Proper File"))
        }
        cb(undefined,true)
    }
})
//CRUD Operations for Products by Employees
//Create Products
router.post("/",urlencodedParser, auth, upload.single("pPicture"), async (req, res) => {
    try {
        // const product = new Product(req.body)
        console.log(req.body)
        if (!req.file){
            const product = new Product({
                ...req.body
            })
            await product.save()
            return res.status(201).send(product)
        }
    }
})
```

```
        const buffer = await sharp(req.file.buffer).resize({ height: 250, width: 250}).png().toBuffer()
        const product = new Product({
            ...req.body,
            pPicture: buffer,
            pPictureURL: "http://127.0.0.1:4000/products/picture/" + req.body.pCode
        })
        await product.save()
        res.status(201).send(product)
    }
} catch (e) {
    console.log(e)
    res.status(400).send()
}
})
// GET /products?availability=true
// GET /products?limit=2&skip=2
// GET /products?sortBy=createdAt:asc
//View All Products
router.get("/summary", async (req,res) => {
    const match = {}
    const sort = {}
    if (req.query.availability) {
        match.pAvailability = req.query.availability === "true"
    }
    if (req.query.sortBy) {
        const parts = req.query.sortBy.split(":")
        sort[parts[0]] = parts[1] === "asc" ? 1 : -1
    }
    try {
        const allProducts = await Product.find(match)
        // if (!req.user){
        //     return res.status(404).send()
        // }
        res.send(allProducts)
    } catch (e) {
        console.log(e)
        res.status(500).send(e)
    }
})
```

```

//View Product with ProductID
router.get("/:code", async (req,res) => {
    const pid = req.params.code
    try {
        const foundProduct = await Product.findOne( { pCode:pid} )
        if (!foundProduct){
            return res.status(404).send()
        }
        res.send(foundProduct)
    } catch (e) {
        res.status(500).send()
    }
})
```

// Updating Existing Products

```

router.patch("/:code",urlencodedParser, auth, async (req,res) => {
    console.log("Update Route is HIT")
    const pid = req.params.code
    const updateFieldsReq = Object.keys(req.body)
    const validFields = ["pCount", "pAvailability", "pPriceEst", "pDesc"]
    const isValidateFields = updateFieldsReq.every( (field) =>
validFields.includes(field))
    if (!isValidateFields){
        console.log(req.body)
        console.log("valid fields are", validFields)
        return res.status(400).send({ "error":"Invalid Update Requested"})
    }
    try{
        console.log(req.body)
        const foundProduct = await Product.findOne({pCode: req.params.code})
        updateFieldsReq.forEach((updateField) => foundProduct[updateField] =
req.body[updateField])
        if (!foundProduct){
            return res.status(404).send()
        }
        await foundProduct.save()
        console.log("Updated "+foundProduct.pCode)
        console.log("Requested Update:", req.body)
        res.send(foundProduct)
    }
```

```
        } catch (e) {
            console.log(e)
            res.status(400).send(e)
        }
    })
}

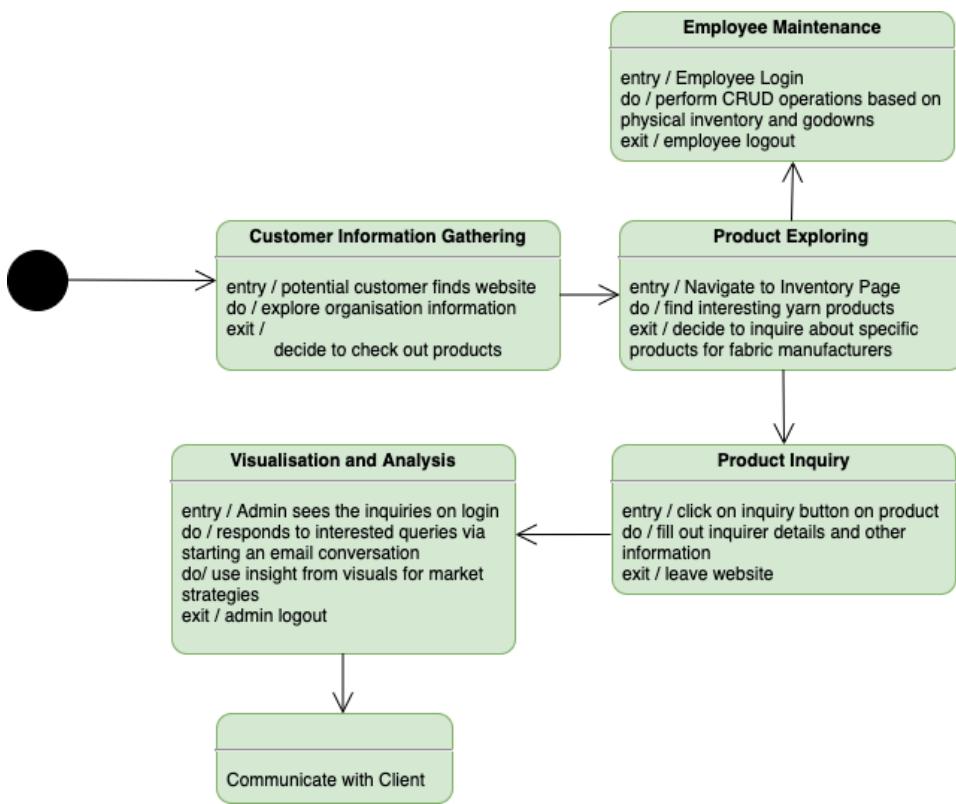
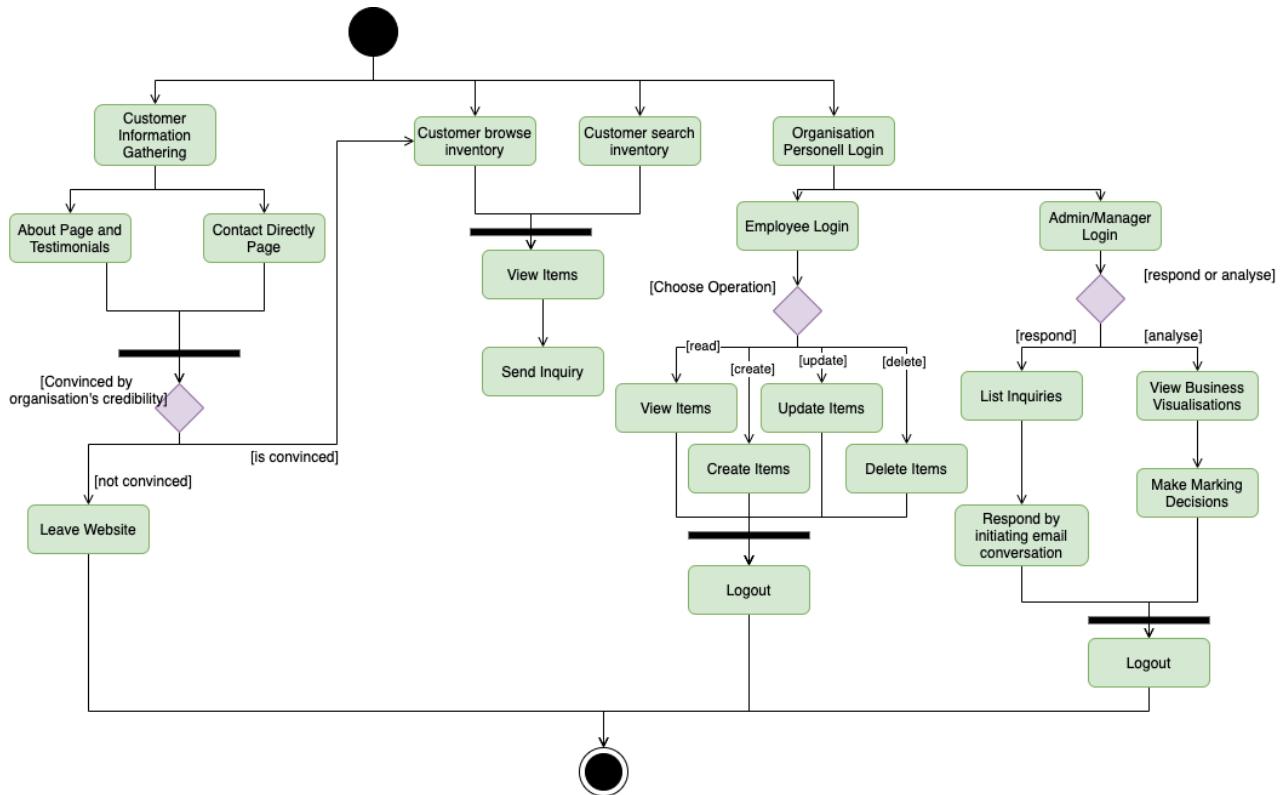
router.delete("/:code", auth, async (req,res) => {
    try {
        const deletedProduct = await Product.findOneAndDelete({pCode:req.params.code})
        if (!deletedProduct){
            return res.status(404).send()
        }
        res.send(deletedProduct)
    } catch (e) {
        res.status(500).send()
    }
})

// GET picture
router.get("/picture/:code", async (req,res) => {
    try{
        const product = await Product.findOne({pCode: req.params.code})
        if(!product || !product.pPicture) {
            throw new Error("Product or Picture doesn't exist")
        }
        res.set("Content-Type","image/png")
        res.send(product.pPicture)
    } catch (e) {
        console.log(e)
        res.status(404).send(e)
    }
})

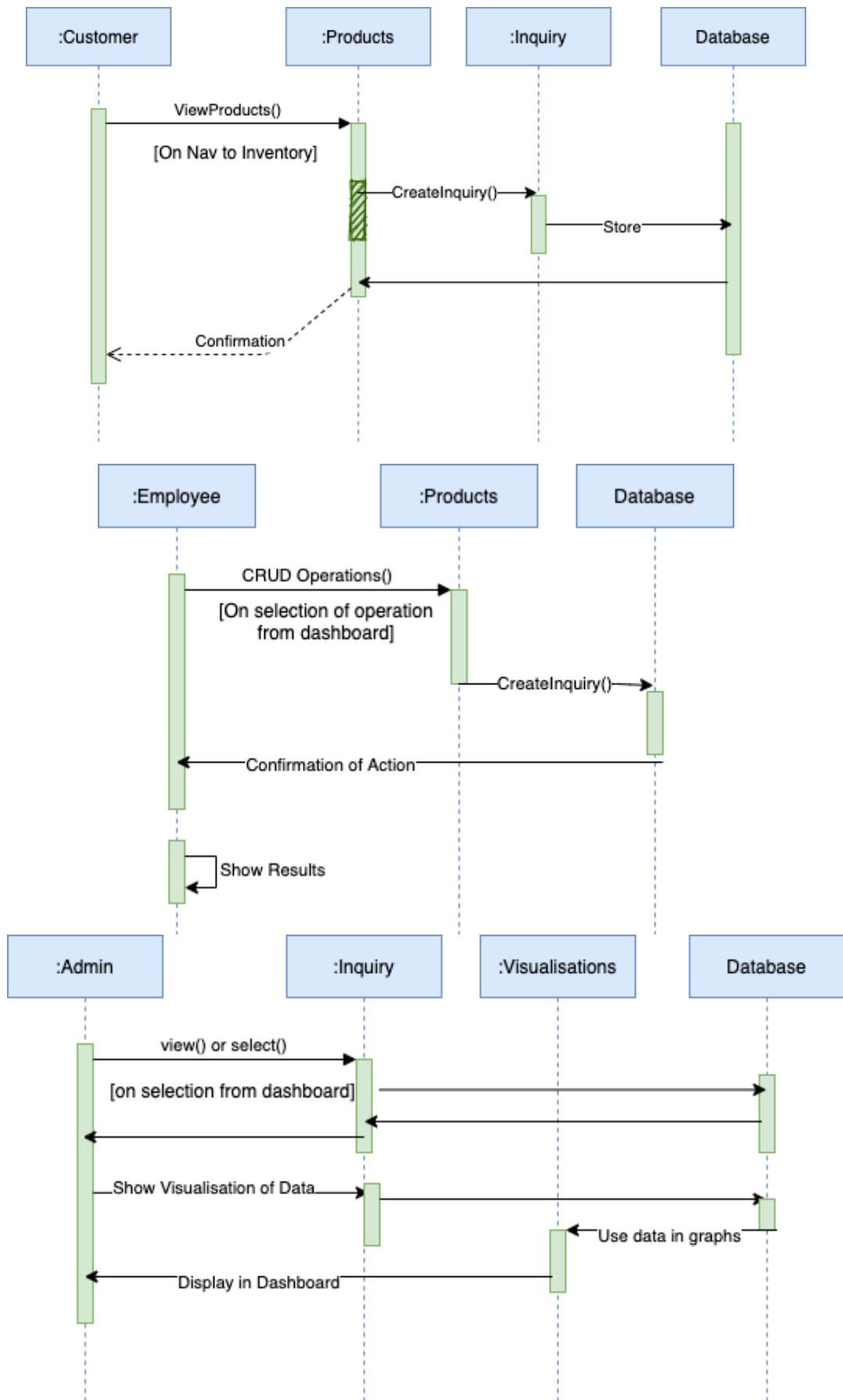
module.exports = router
```

Design Model

Activity Diagram and State Diagram (not STN)



Sequence Diagrams



Evaluation based on Schneiderman's 8 Rules

1. **Strive for Consistency:** The theme of the webpage has been maintained uniformly, thereby ensuring consistency in the overall design.
2. **Enable frequent users to use shortcuts:** The website showcases the newest collection always on the top, thereby allowing frequent users to always see the newest products first
3. **Offer Informative Feedback:** The website has links and auto redirects that allow a user to always stay in control of all his actions
4. **Design Dialogs to Yield Closure:** The website has inbuilt dialogues that respond on every action of the user, including error messages
5. **Offer error prevention and simple error handling:** The website has inbuilt error handling functions and incase the user performs an action that results in an error, he can retry that action
6. **Permit Easy Reversal of Actions:** The website allows a user to delete his actions and retry the action, this function is integrated with the backend API's.
7. **Support Internal Locus of Control:** The website will ask for a confirmation before the user finalises an action, thereby allowing the user to stay in absolute control.
8. **Reduce Short Term Memory Load:** The user will not have to remember an information, all the information is automatically rendered by the webpage, thereby reducing the user's short term memory load

Literature Survey

A. **Research on B2B2C E-commerce Website Design Based on User Experience**

*Method:*The B2B2C model is a new network communication sales method

*Findings:*The focus issue that needs attention in website design is how to help consumers quickly find the target product.

B. **An Empirical Study of Website Personalization Effect on Users Intention to Revisit E-commerce Website Through Cognitive and Hedonic Experience**

*Method:*Author collected data from web users using both e-commerce websites having personalization features on their web portal.

*Findings:*Presentation personalization adjusts the layout of user interface and provides content with good look and feel in the form of personalized themes, font, and background color generating ease of use and enjoy while browsing personalized e-commerce

C. **The value of comparative usability and UX evaluation for e-commerce organisations**

*Method:*This paper investigates a method that uses factual user behaviour data that are collected from interaction with the website to be evaluated, as well as data from user interaction with closely competing websites that offer directly comparable services.

*Findings:*Telecom A's results revealed clear usability and navigational issues, especially in comparison to the results of Telecom B and Telecom C. Telecom B users had more fixations on parts of the website indicating that there are more screen elements than on the other two sites. Although TelecomB fared well when compared to TelecomA, its results show that TelecomC supports the top-up functionality much more effectively

D. **The adoption of eCommerce communications and applications technologies in small businesses in New Zealand**

*Method:*This paper investigates the impact of 10 different factors, extended from the technological innovation literature. They focus on the adoption of different eCommerce communications and applications technologies (EC) in small businesses (SMEs) in New Zealand (NZ).

*Findings:*The extensive testing of the research model can lead to more generalizations across the different countries. Electronic commerce possesses many facets and this research has introduced three categories of adoption to capture the different aspects involved in the criteria for E-commerce adoption in Small to Medium-Sized Enterprises in New Zealand Professionals could focus and prioritise their strategies in marketing/selling their E-commerce products and solutions to Small to Medium-Sized Enterprises by focusing on certain E-commerce technology or by capitalising on certain motivators and avoiding certain hindrances.

E. **User Attitudes Regarding a User-Adaptive eCommerce Web Site**

*Method:*In this paper, the researchers executed an iterative series of three studies,carried out in multiple laboratory settings. The studies were carried out with a mixed group sessions in which participants were walked through the use of a prototype by an experimenter, and individual user evaluation sessions in which subjects interacted with a working prototype.

*Findings:*This research showed that,overall the users wanted to feel as if they are in control. The feeling of a user's sense of control is in whether they can readily make sense of the

interaction with a site. Not being able understanding why the site is displaying particular content engenders for users a sense of loss of control

F. E-Commerce adoption in developing countries: a model and instrument

*Method:*The purpose of this paper was to define the E-Readiness concept, to suggest an underlying model of E-Commerce adoption to identify the relevant managerial, organizational, technological, and environmental factors that affect decisions to open or develop eCommerce systems in developing countries; and to develop sufficiently validated measures to show the utility of the model.

*Findings:*They constructed and empirically tested a model of E-Commerce adoption: the PERM. The underlying theoretical perspectives allowed them to identify the relevant eCommerce, managerial, organizational, and contextual factors that could explain E-Commerce adoption and subsequent development.The study served to highlight contextual limitations that often are taken for granted in other markets.

G. The rise of ecommerce as an epidemic in the small world of venture capital

*Method:*This chapter examines the rise of an industry as a diffusion process across Venture Capitalists . Following a venerable tradition of research on diffusion processes, the percolation mechanism studied here is a network. The network is constituted of syndicated deals among Venture Capitalists and is analyzed as a small world

*Findings:*The epidemic model works well statistically in the presence of key controls, especially contagion effects represented by lagged syndications. Whether E-commerce is the only industry to which the model applies is a question that can be answered in a broader analysis

H. E-Commerce in the Textile and Apparel Supply chain management: Framework and Case Study

*Method:*The paper aims to understand the current state of ecommerce technologies applications in the Textile and Apparel supply chain and provide recommendations for future improvement. The paper puts forward that apparel supply chain management system should be designed integrated into the e-commerce system. A thorough analysis of the mechanism of e-commerce has been made and framework of e-supply chain analyzed accordingly.

*Findings:*Think about integrating the Internet with the existing supply chain network, rather than setting up a separate e-commerce. Structure e-commerce logistics to accommodate packages instead of pallets. Devise shipping pricing strategies that reflect the costs of activities.

I. Progressive SWOT Analysis of “E” – Textile Industries

*Method:*This paper mainly focuses on a SWOT analysis for the integration of the already existing textile industry with E-commerce and concludes if is an advantage or disadvantage

*Findings:*E commerce has made its special identity in the textile industry. Nowadays it has become an essential part of the textile industry. With the help of Information & Communication Technology (ICT), people are able to communicate rapidly with each other. More knowledge about EDI will support the quick exchange of products. Establishing & maintaining brand name is the biggest challenge for all the Electronic based textile industries. E commerce is the easiest way to enter in to the global market

Methodology

Technology Used

The solution developed is a full stack web application with MERN (MongoDB, ExpressJS, ReactJS and NodeJS). Since the application is aimed at small and medium scale companies there is not necessity for high level scalability, so NoSQL databases like MongoDB work perfectly well. The React and Node combination provides the fluidity and speed in dynamic updatations and rendering of data. All the API routes are tested using JEST.

Stakeholders

The stakeholders are doing to be small-scale textile trading companies. This could include Yarn Trading Companies, Fabric Traders (Bedsheets, Clothes etc) and weavers. The stakeholders studied for this particular project are yarn trading companies and small scale textile manufacturers in the Erode and Tiruppur region. This region quite well known for its textile manufacturers. *Kumar Yarns, Chennimalai* is one such yarn trading company studied for this particular project.

Functionalities For Public Users:

1. Information about the organisation, history, credibility and administration will be available.
2. Inventory details and availability of products availability in real time. Any update in the data is *dynamically* updated here.
3. Filtering the products based on availability, Yarn Count, Yarn Colour and other metrics of yarn quality.
4. Search feature to lookup any item directly
5. Inquiry submission or asking for a quote - Data will be stored for business intelligence.

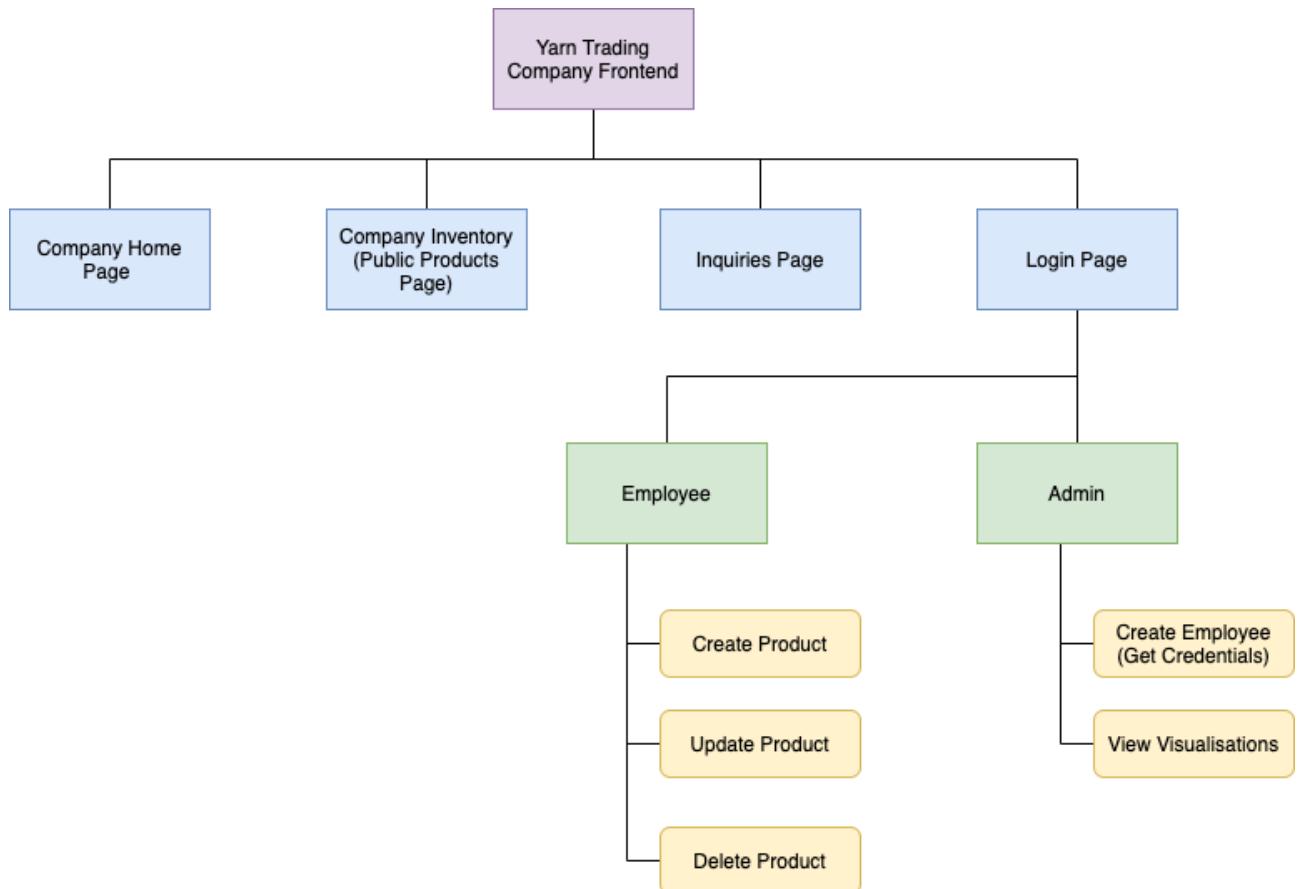
For Employees with Inventory Access:

1. *Exclusive login* for additional privileges.
2. *Keep track of the inventory details* for decisions on purchase and sales.
3. *Creating, Reading, Deleting, Updating* the details of products in the inventory (CRUD Operations).

For Administration/Managers:

1. View list of submitted inquiries
2. Visualisations of inquiry data - to gain insight on products showing more interest and business intelligence. As more inquiries flow in the visualisations *dynamically* change.
3. Delete employee users or update information if needed.
4. Response to Inquiries - Task Completion

Architecture



The architecture of the interface is designed to be minimal and simple. This is because the primary stakeholders are not extremely tech-savvy. The application will work as a website plus special portal for the employees and manager.

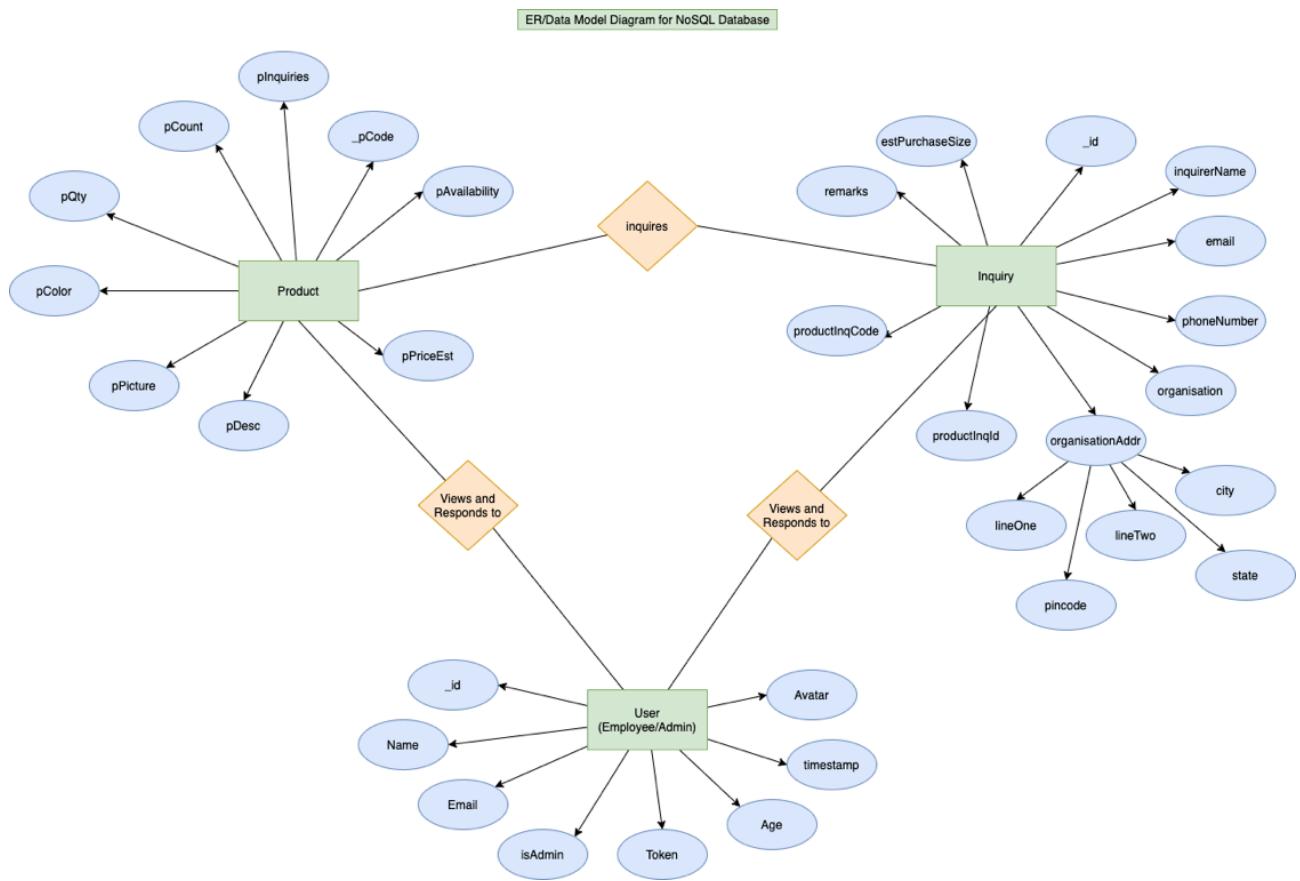
The Company home page is the landing page of the web application. It is a webpage with impressive UI elements, company vision, mission and values. It also contains testimonials and history of the company to create trust in a casual user.

The product inventory page is publicly available and is updated real time by employees with employee accounts. All updations occur without the page having to re-load because of the reactJS frontend. The inquiry page is currently a separate form that takes in details of the inquirer and uses stores them in the backend.

The login pages serves as a single login for both the *employees* and the *managers*. They login with the credentials provided to them by the company. An employee has access to create, view, update and delete products. The admin has capabilities to create employee accounts. The admin also has the key role of viewing visualisations of product and inquiry data for market insights.

Visualisations

The data model used is as follows. Note that the visualisations are created via the relationship between different product and inquiry collections.



The following are the visualisations currently implemented

1. Total Inquiries against Products
2. Average purchase size (generated from inquiries) against products
3. Total Inquiries reviewed against Yarn Count
4. Total Inquiries against States of Inquirer

Note: Yarn Count here means the girth to the thickness of the yarn. The actual purpose of the yarn changes with change in yarn count (even if its of the sample standard/quality). Lower Counts (10s, 15s) mean that the yarn is coarser, more rough and used in thicker material. Higher Counts (40s, 60s) are used for delicate fabrics like napkins t-shirts.

Testing and Results

Testing is done in 2 levels in the application - Usability Testing and API Testing.

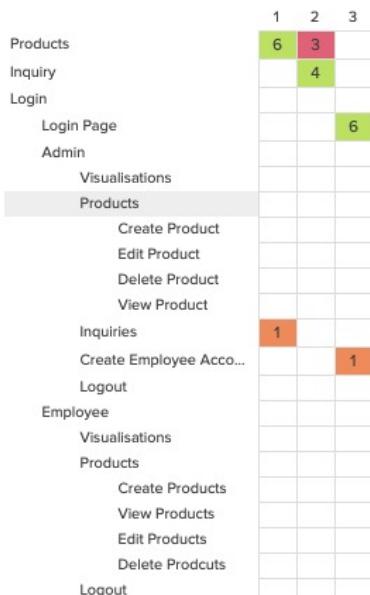
Usability Testing

First usability testing is done with the help of tree jack. Tree Jack is a free online usability testing software that tests navigation and generates graphs for the same. It does so with the help of test taker surveys. We tested our project with it and the results we obtained are as follows

The tests show the success ratio and comfort of the users in reaching the correct destinations.

Participant destinations

Correct Incorrect (< 10% of responses) Incorrect (10% - 20% of responses) Incorrect (> 20% of responses)

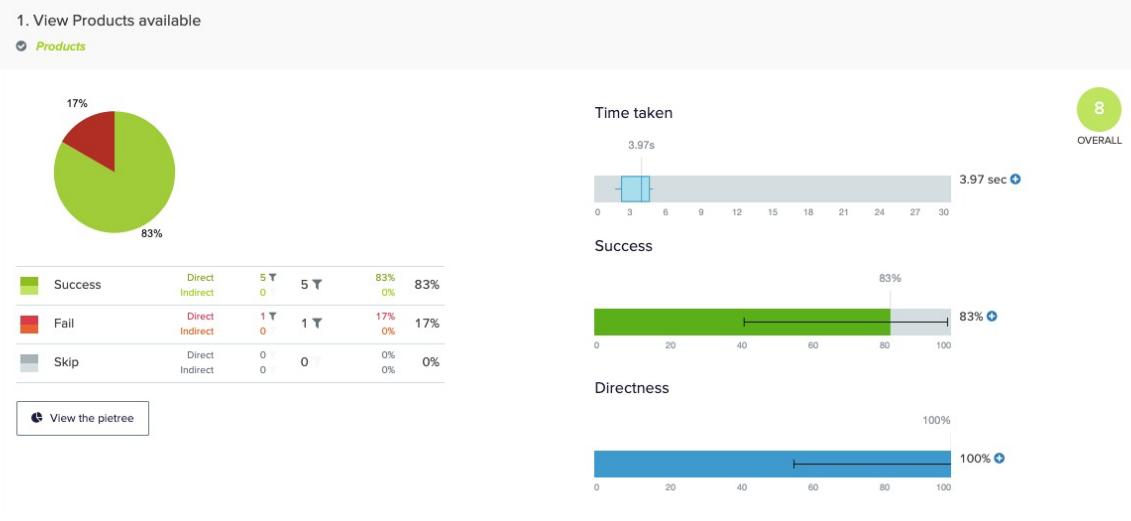


Participant paths

1. View Products available

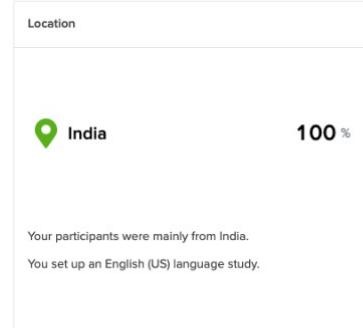
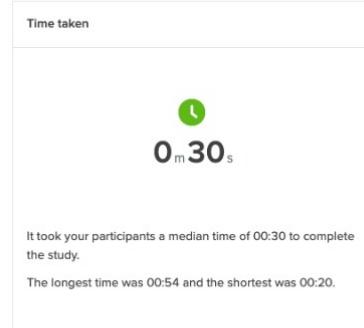
Filter: Direct success Indirect success Direct failure Indirect failure Direct skip Indirect skip

Success	Participant	Path
Green	2	> Products
Green	3	> Products
Green	4	> Products
Green	5	> Products
Red	6	> Login > Admin > Inquiries
Green	7	> Products
Green	8	> Products

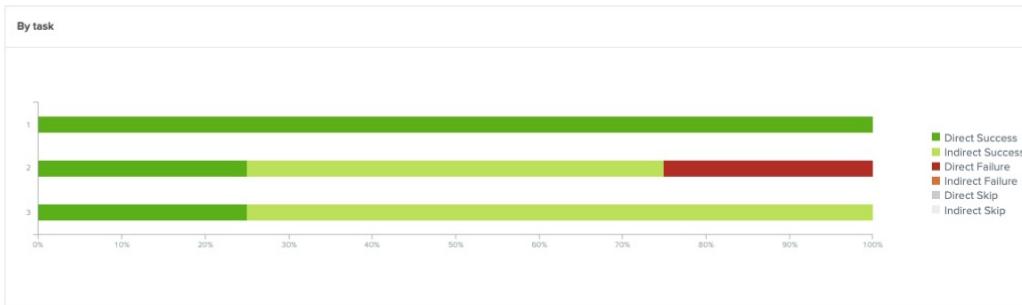
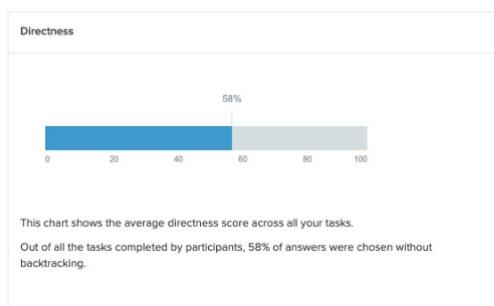
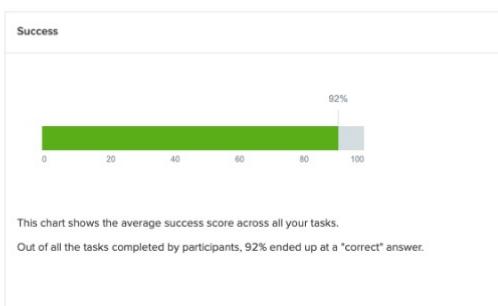


Participants

There is still 1 active participant



Tasks



[Click to go forward, hold to see history](#)

First click [?](#)

1. View Products available

Label	Visited first	Visited during
Products	86%	86%
Inquiry	0%	0%
Login	14%	14%

Skipped by 0% of participants

2. Inquire about a product

Label	Visited first	Visited during
Products	57%	57%
Inquiry	29%	57%
Login	14%	14%

Skipped by 0% of participants

3. If you are an employee of the company, and you are given your credentials, how will you login?

Label	Visited first	Visited during
Products	43%	43%
Inquiry	0%	14%
Login	57%	100%

Skipped by 0% of participants

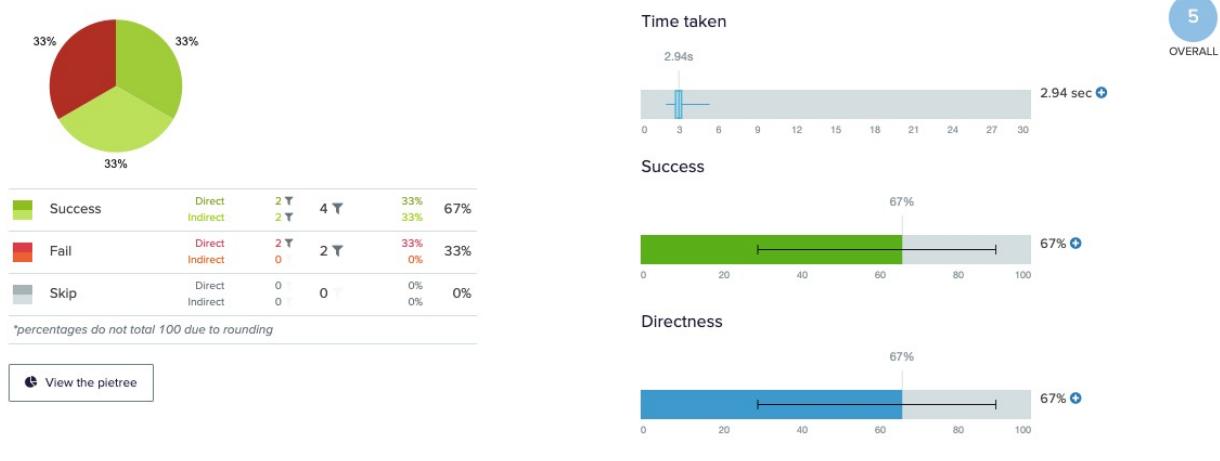
3. If you are an employee of the company, and you are given your credentials, how will you login?

[Login > Login Page](#)



2. Inquire about a product

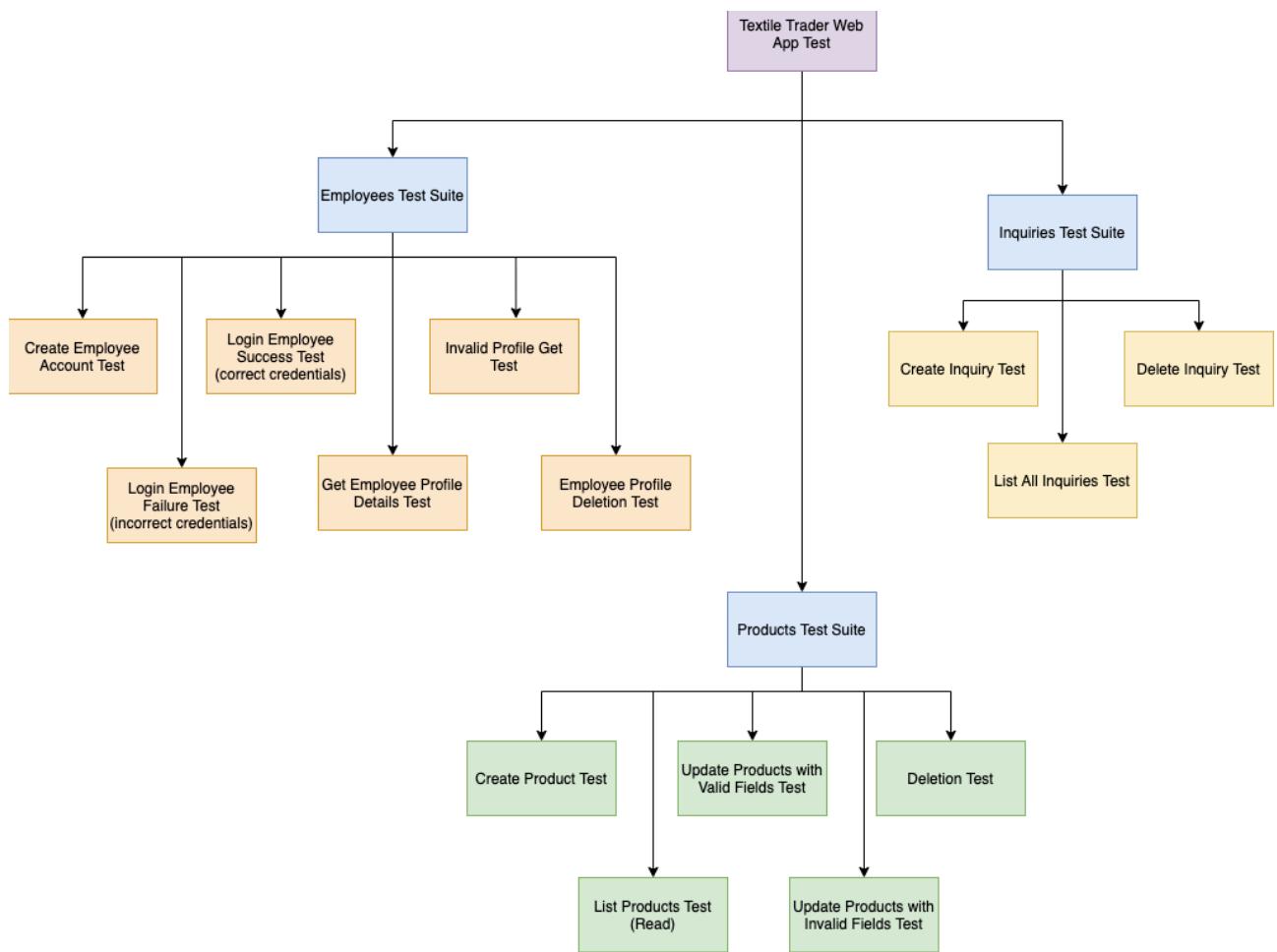
[Inquiry](#)



API Testing

We have also performed our API testing with JEST (A JavaScript testing framework). The architecture for the test suites for the same are as follows.

*Test Command: npm test (or) env-cmd -f ./config/test.env jest --watch –runInBand
Dependencies*



Dependencies of Test Module:

1. JEST JAVASCRIPT TESTING FRAMEWORK
2. SUPERTEST (From Express For Asynchronous Api Testing)
3. ENV-CMD (Npm Module For Configuring Test Environment And Variables)

DB is set up and turndown for testing purposes with 1-2 sample entities.

```
tests
  fixtures
    db.js
    employee.test.js
    inquiry.test.js
    products.test.js
```

Successful test results

```
PASS tests/employee.test.js (6.88 s)
```

- Console

```
  console.log  
    MONGO DB CONNECTED!
```

```
      at src/db/mongo.js:10:13  
        at runMicrotasks (<anonymous>)
```

```
PASS tests/inquiry.test.js
```

- Console

```
  console.log  
    MONGO DB CONNECTED!
```

```
      at src/db/mongo.js:10:13  
        at runMicrotasks (<anonymous>)
```

```
PASS tests/products.test.js
```

- Console

```
  console.log  
    MONGO DB CONNECTED!
```

```
      at src/db/mongo.js:10:13  
        at runMicrotasks (<anonymous>)
```

```
Test Suites: 3 passed, 3 total
```

```
Tests:       13 passed, 13 total
```

```
Snapshots:  0 total
```

```
Time:        10.317 s
```

```
Ran all test suites related to changed files.
```

```
Watch Usage: Press w to show more. □
```

For the above tests, manually designed test cases are tabulated as follows

Login Module

S. no	Class	Input Conditions	Expected Results	Actual Results
1	Correct Credentials	Valid Input Email and Correct Corresponding Password	Successful redirect to homepage of Employee	Successful redirect to home page of employee
2	Incorrect Credentials	Invalid Email (doesn't exist) and password Or Correct Email and incorrect corresponding password	Stay in Login page and ask to retry	Error is thrown and user is notified to retry
3	Empty Input Credentials	Blank Email or Blank Password Fields	Stay in Login page and ask to retry	Highlight blank fields and display that they are mandatory

Product Creation

S. no	Class	Input Conditions	Expected Results	Actual Results
1	Successful	Product attributes: 1. Color 2. Quantity (Non-negative) 3. Availability 4. Price (Non-Negative) 5. Count ($0 < \text{Count} < 100$) 6. Code (length=4) 7. Picture Are provided.	Database updation and success message	Database updated with new product and success message
2	Invalid Range of Input	Product attributes with range: 1. Color 2. Quantity (negative) 3. Availability 4. Price (Negative) 5. Count ($0 < \text{Count} < 100$) 6. Code (length=4) 7. Picture Are provided.	Throw Error and Stay in Creation Page	Throw Error "Invalid Input Fields" and stay in page
3	Mandatory inputs missed	Any of the following mandatory fields are missed: 1. Code 2. Color 3. Quantity 4. Count	Throw Error and Stay in Creation Page	Throw Error "Empty Input Fields" and stay in page

Product Updation

S.no	Class	Input Conditions	Expected Results	Actual Results
1	Successful	Updation attributes selected within correct fields: 1. pCount 2. pAvailability 3. pPriceEst 4. pDesc	Database updation and success message	Database updated with new value and success message
2	Invalid Fields	Updation attributes selected within incorrect fields: 1. pQty 2. pCode	Throw Error and Stay in Creation Page	Throw Error "Fields are not allowed for updation" and stay in page

Product Deletion

S.no	Class	Input Conditions	Expected Results	Actual Results
1	Successful	Correct ID of the product is specified (product with the provided ID exists)	Database updation and success message	Database updated (deleted the product) and success message
2	Invalid Fields	Correct ID of the product is specified (product with the provided ID does not exist)	Throw Error and Stay in Creation Page	Error message "Invalid Product ID Please try again"

Product CRUD Operation Access (Authorisation Middleware)

S.no	Class	Input Conditions	Expected Results	Actual Results
1	Successful for Employee level operation	The Bearer Token in the 'Auth header' of API call is valid and verified.	Access to API end point granted	CRUD operation is performed
2	Successful for Admin level operation	The Bearer Token in the 'Auth header' of API call is valid the isAdmin field is true on decrypting the JWT token.	Access to Admin only API end point granted	Operation is performed
3	Invalid Auth Header of User (Employee or Admin)	The Bearer Token in the 'Auth header' of API call is invalid.	Access to API end point denied	CRUD operation is not performed

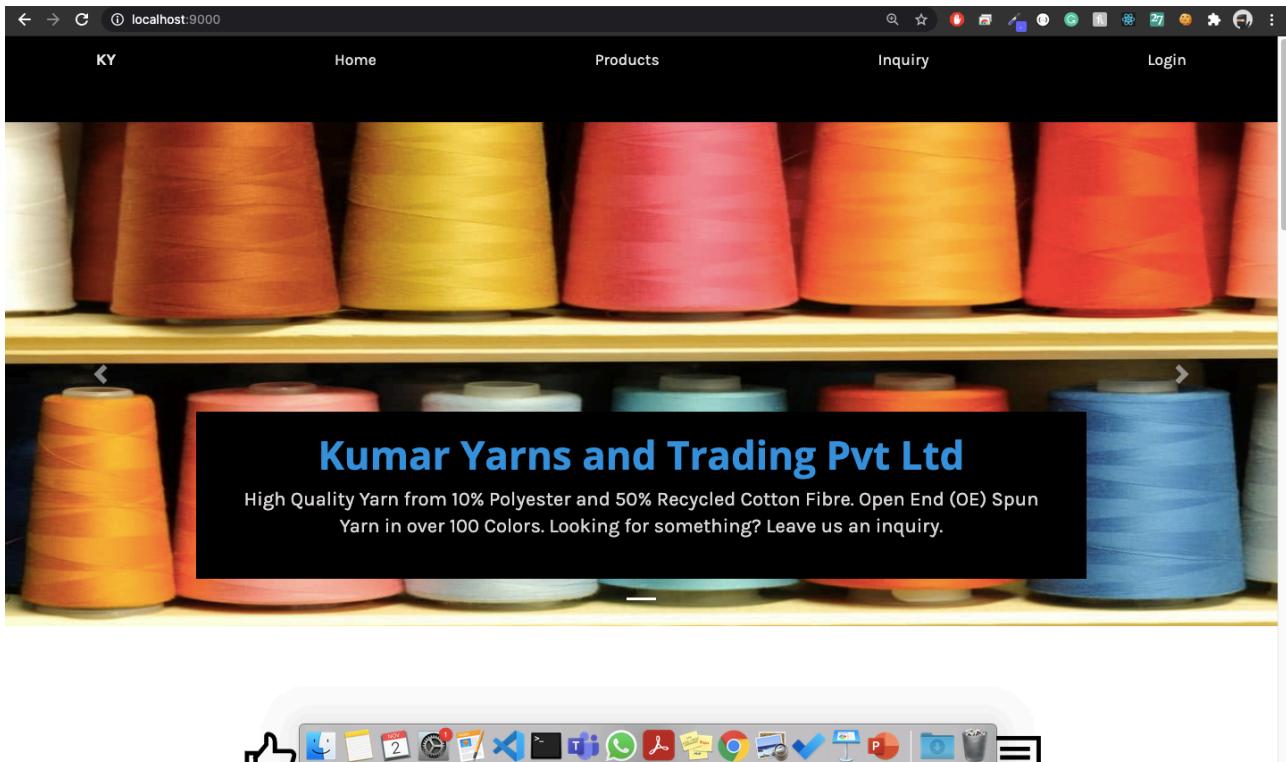
S.no	Class	Input Conditions	Expected Results	Actual Results
4	Correct Auth Header of Employee for Admin Only Operation	The Bearer Token in the 'Auth header' of API call is valid but isAdmin is false on decrypting the JWT token.	Access to API end point denied	Display Error "Admin Privilege Required"

Inquiry Creation

S.no	Class	Input Conditions	Expected Results	Actual Results
1	Successful	Inquiry attributes: 1. inquirerName 2. email (valid email range) 3. phoneNumber (valid ph no) 4. Organisation 5. Organisation Address 6. Estimated Purchase Size 7. Remark Are provided.	Database updation and success message	Database updated with new inquiry and success message
2	Invalid Range of Input	Inquiry attributes: 1. inquirerName 2. email (invalid email range) 3. phoneNumber (invalid ph no) 4. Organisation 5. Organisation Address 6. Estimated Purchase Size 7. Remark Are provided.	Throw Error and Stay in Creation Page	Throw Error "Invalid Input Fields" and stay in page
3	Mandatory inputs missed	Any of the following mandatory fields are missed: 1. Inquirer Name 2. Email 3. Phone Number	Throw Error and Stay in Creation Page	Throw Error "Empty Input Fields" and stay in page

Experimental Results

The following are screenshots of the working/interface of the application with RectJS.



Quality Guaranteed
For over 20 years we have guaranteed our Yarn Quality. Return goods within 3 business days if not satisfactory**

On-time Delivery
We deliver in one or two business days from the day of invoice given that we are stocked on your purchase. For details visit our inventory section.

Pay Online/via Cheque
Owing to the current pandemic we only take online payments either via Net Banking or via a Cheque posted to our address.

Quality Focused. Customer First.
For us Quality is everything. For great quality fabric and clothes it

For us Quality is everything. For great quality fabric and clothes it starts from great quality fibre. We obtain and stock first quality yarn obtained spun from high grade fibre. For other purposed yarn of lesser quality from recycled cotton is also available.



Values. Employee Centered

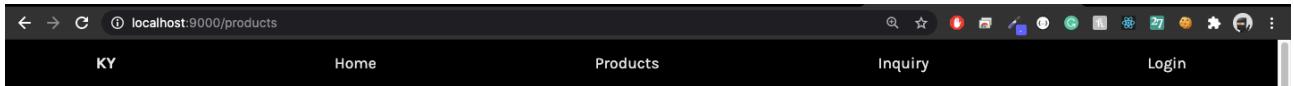
We promote a healthy and robust work environment and culture. Our employees come from all spheres of life. The textile market is ever growing and flourishing. Come check us out if you are looking for a job!

History and Reputation. Trust.

Since we have been in the yarn market for longer than most others, we have a good understanding of customer expectation. We build trust and are always available to feedback.

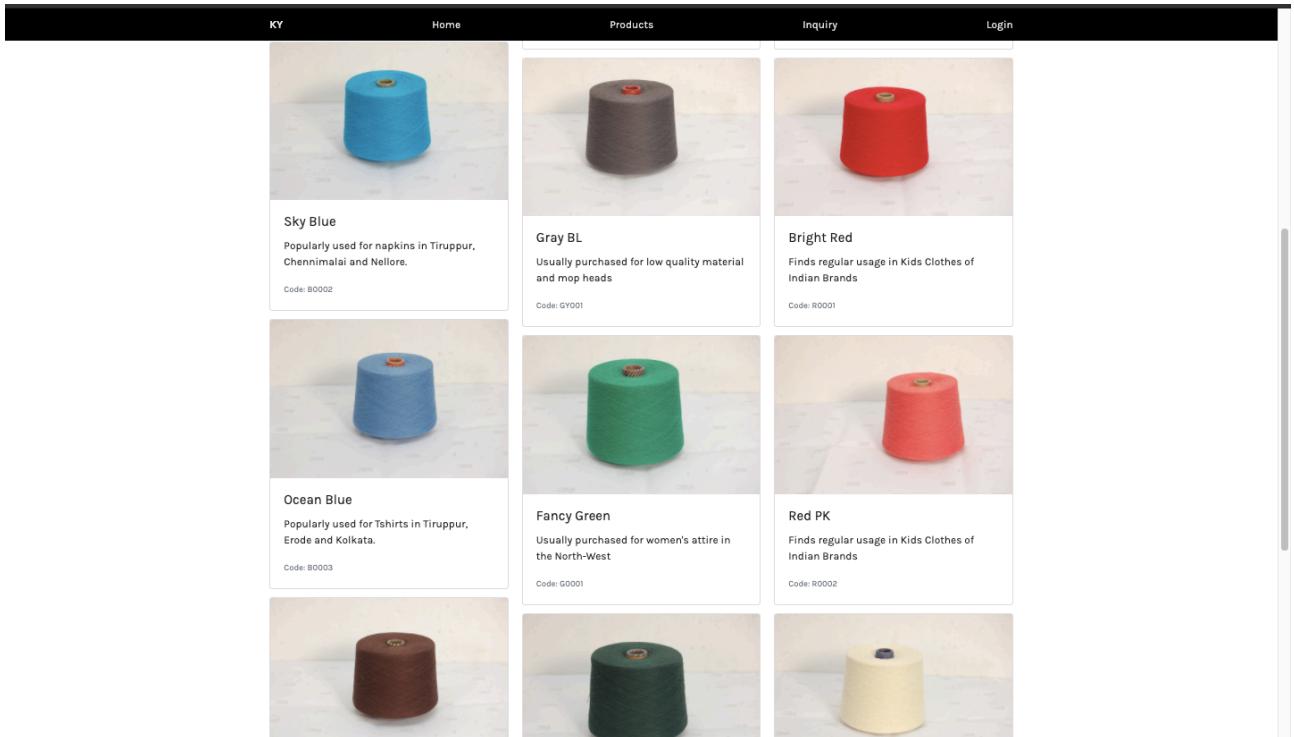


Products Page (Dynamically rendered, updated)



Products

Currently available Products. Note that this page is constantly updated and you get real time data. Please leave us an inquiry if you are interested in a product.



Inquiry Page

KY Home Products Inquiry Login

Inquiry form

You may inquire for upto 1(one) product at a time. Please fill in the details below.

Billing address

Your name	Organisation name	
Sanjit	XYZ	
Product Code	Phone Number	
B0001	9842314733	
Email		
ex@ex.com		
Address		
1234, Sanjit		
Address 2		
Chennimalai		
City	State	Zip
Erode	Tamil Nadu	638051
Estimated purchase size	Remark	
140	ASAP	

Successful!

Login Page (Common for Employee and Admin)

KY Home Products Inquiry Login



Kumar Yarns

Sign into your account

Login

Crud Operations on products is allowed for Employee. Other Functionality blocked for Employee.

Dashboard- employeeTest1							
Products		Inquiry					
New Product							
Code	Available	Color	Count	Quantity	Price	Delete	Edit
B0001	true	Royal Blue	20	400	120	<button>Delete</button>	<button>Edit</button>
B0002	true	Sky Blue	10	200	70	<button>Delete</button>	<button>Edit</button>
B0003	true	Ocean Blue	30	700	89	<button>Delete</button>	<button>Edit</button>
BR003	true	Brown AEW	40	700	120	<button>Delete</button>	<button>Edit</button>
BU001	true	Biscuit TE	20	150	90	<button>Delete</button>	<button>Edit</button>
GY001	true	Gray BL	10	10	40	<button>Delete</button>	<button>Edit</button>
G0001	true	Fancy Green	10	70	58	<button>Delete</button>	<button>Edit</button>
G0002	true	Camo Green	20	140	73	<button>Delete</button>	<button>Edit</button>
G0005	true	Green GR	30	500	90	<button>Delete</button>	<button>Edit</button>
P0001	true	Pink BW	30	200	100	<button>Delete</button>	<button>Edit</button>
R0001	true	Bright Red	10	70	120	<button>Delete</button>	<button>Edit</button>
R0002	true	Red PK	40	20	130	<button>Delete</button>	<button>Edit</button>
L0002	true	Lemon Yellow	40	100	150	<button>Delete</button>	<button>Edit</button>

On Clicking Create Product Button - New Modal - Success - Change is reflected in product page immediately.

Create a new product ×

B0011
15
false
Peacock Blue
100
120
Sample v1
<input type="button" value="Choose file"/> 7
<input type="button" value="Close"/> <input type="button" value="Create Product"/>

Create a new product ×

B0011
15
false
Peacock Blue
100
120
Sample v1
<input type="button" value="Choose file"/> 7
<input type="button" value="Close"/> <input type="button" value="Done"/>

Create a new product ×

B0011
15
false
Color
100
120
Sample v1
<input type="button" value="Choose file"/> 7

Close
Try Again

On Failure

On clicking Edit. Only some fields are permitted for editing.

The screenshot shows a web-based application interface for managing products. On the left, there's a sidebar with links for Dashboard, Inquiries, and Products. The main area has tabs for Home, Products, and Inquiry, with the Products tab currently active. A sub-menu under Products shows a 'New Product' button. The main content area displays a table of products with columns for Code, Available, Name, Price, Delete, and Edit. One row in the table is selected, and a modal dialog is overlaid on the page. This modal is titled 'Update the following product fields'. It contains four input fields: 'Price' with the value '10', 'Available' with the value 'true', 'Color' with the value '100', and a larger input field for 'Sample v2' which is currently selected and highlighted with a blue border. At the bottom of the modal are two buttons: 'Close' and 'Update Product'.

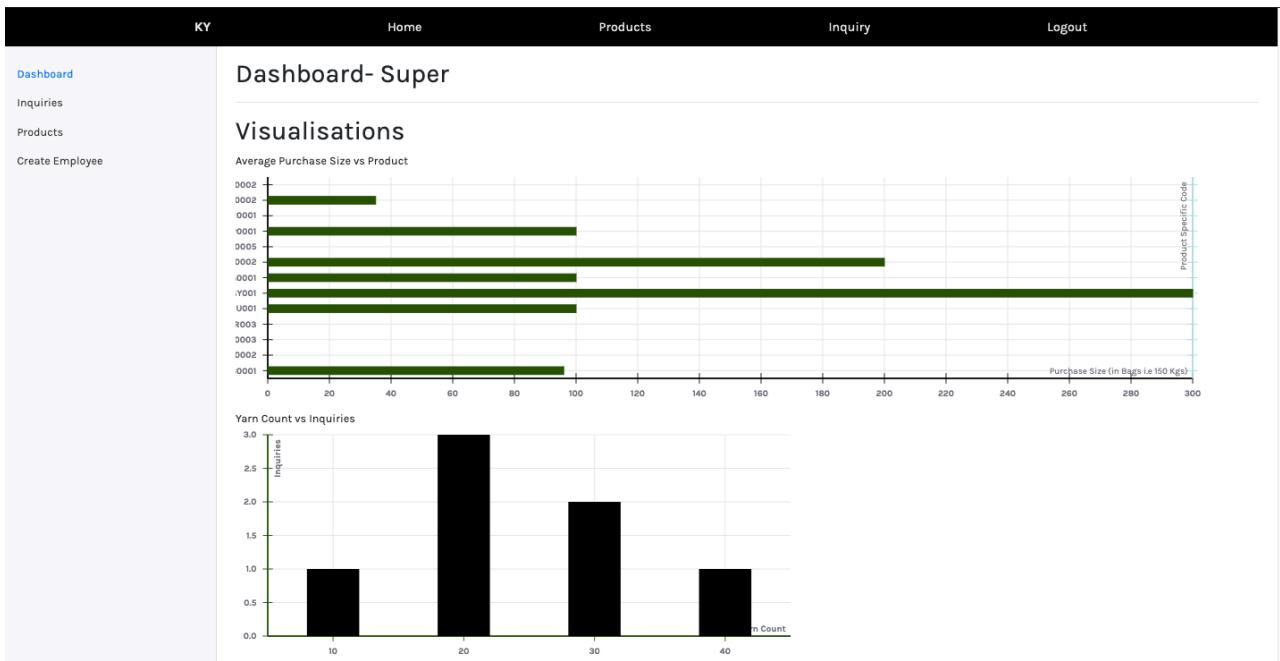
On hitting delete the product is deleted and disappears from the products page.

KY	Home	Products	Inquiry	Logout				
Dashboard	Dashboard- employeeTest1							
Inquiries								
Products								
Products								
New Product								
Code	Available	Color	Count	Quantity	Price	Delete	Edit	
B0001	true	Royal Blue	20	400	120	Delete	Edit	
B0002	true	Sky Blue	10	200	70	Delete	Edit	
B0003	true	Ocean Blue	30	700	89	Delete	Edit	
BR003	true	Brown AEW	40	700	120	Delete	Edit	
BU001	true	Biscuit TE	20	150	90	Delete	Edit	
GY001	true	Gray BL	10	10	40	Delete	Edit	
G0001	true	Fancy Green	10	70	58	Delete	Edit	
G0002	true	Camo Green	20	140	73	Delete	Edit	
G0005	true	Green GR	30	500	90	Delete	Edit	
P0001	true	Pink BW	30	200	100	Delete	Edit	
R0001	true	Bright Red	10	70	120	Delete	Edit	
R0002	true	Red PK	40	20	130	Delete	Edit	
L0002	true	Lemon Yellow	40	100	150	Delete	Edit	

Admin Functions - Visualisations

Graph 1: Product (Code) vs Average Product Purchase Size (know how much to purchase accordingly). Horizontal Bar graph.

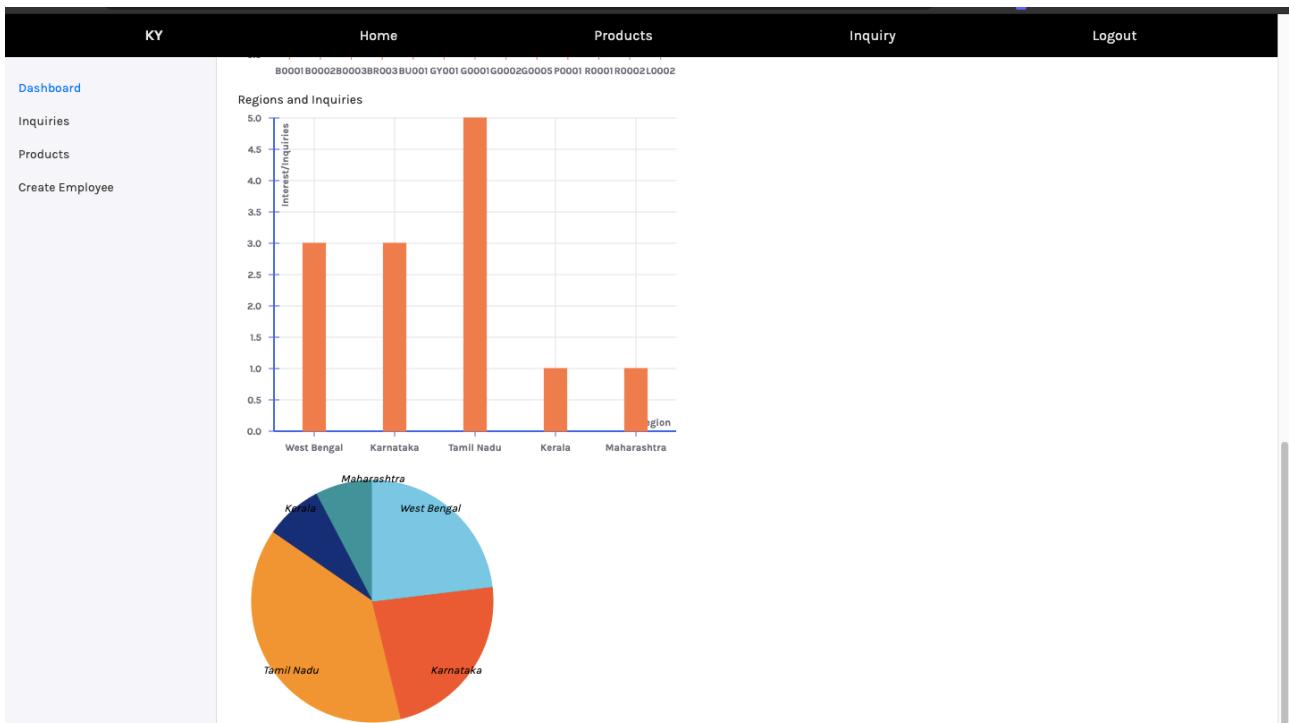
Graph 2: Yarn Count vs Inquiries (know which kind of counts receive major interest). Vertical Bar graph.



Graph 3: Overall trend of inquiry reception. Line Graph.



Graphs 4 and 5: State-wise interest shown in products. Both as Bar graph and Pie Chart.



Create employee account or manager account for existing admin

KY	Home	Products	Inquiry	Logout
Dashboard Inquiries Products Create Employee	<h3>Dashboard- Super</h3> <h4>Create</h4> <p>Full Name</p> <p>Email address</p> <p>*****</p> <p>Admin? 'true' or 'false'</p> <p>Age</p> <p style="text-align: center;">Login</p>			

Conclusion and Future Scope

Though this project we have created a minimal and simplistic prototype for a full scale web application that can act as an inventory tracking and inquiry generation system while at the same time creating visualisations that provide meaningful insights. We have done so while keeping the interface minimal and simplistic to make it available to anybody who can handle a normal website since our target audience are not exactly tech savvy. We have made sure to test all the functionality and the usability too.

The scope for this project is really interesting. This is a very prevalent situation in trading companies of small scale that cannot afford sophisticated customisable software that exists in the market. What these companies need is a relatively simpler application which can be easily built with the prevalent technology. This opens up a potential market for simpler web tools which provide business intelligence. These are companies that have *just enough* financial means to pay for a reasonable software but do not have the exposure or means to access such technology. Business insight is usually gained by information gathering and word of mouth. This is not always reliable. With proper marketing, by gaining a few clients it is possible to make this a successful business venture.

That said, there is room for improvement in the sense that, with feedback from the end users we should be able to improve the visualisation standard. Although the UI is clean and minimal, it is still possible to make it fancier.

References

- [1] Cai, L., He, X., Dai, Y., & Zhu, K. (2018). Research on B2B2C E-commerce Website Design Based on User Experience. *Journal of Physics: Conference Series*, 1087, 062043. doi:10.1088/1742-6596/1087/6/062043
- [2] Desai, D. (2018). An Empirical Study of Website Personalization Effect on Users Intention to Revisit E-commerce Website Through Cognitive and Hedonic Experience. *Advances in Intelligent Systems and Computing*, 3–19. doi:10.1007/978-981-13-1274-8_1
- [3] <https://aisel.aisnet.org/confirm2016/9>
- [4] Al-Qirim, N. (2007). The adoption of eCommerce communications and applications technologies in small businesses in New Zealand. *Electronic Commerce Research and Applications*, 6(4), 462–473. doi:10.1016/j.elerap.2007.02.012
- [5] Alpert, S. R., Karat, J., Karat, C.-M., Brodie, C., & Vergo, J. G. (2003). User Modeling and User-Adapted Interaction, 13(4), 373–396. doi:10.1023/a:1026201108015
- [6] Molla, A., & Licker, P. S. (2005). eCommerce adoption in developing countries: a model and instrument. *Information & Management*, 42(6), 877–899.
- [7] Walker, G. (n.d.). The rise of ecommerce as an epidemic in the small world of venture capital. *Advances in Strategic Management*, 3–29. doi:10.1016/s0742-3322(08)25001-1
- [8] Jinfu, W., & Aixiang, Z. (2009). E-Commerce in the Textile and Apparel Supply Chain Management: Framework and Case Study. 2009 Second International Symposium on Electronic Commerce and Security. doi:10.1109/isecs.2009.84
- [9] Udwant, Prashant & Shivankar, Vijay. (2016). Progressive SWOT Analysis of " E " – Textile Industries. *International Journal on Textile Engineering & Processes*. 2. 54-58.
- [10] Deepita Chakravarty. (2002). Work Organisation and Employment Contracts: Technological Modernisation in Textile Firms. *Economic and Political Weekly*, 37(8), 743-749. Retrieved November 3, 2020, from <http://www.jstor.org/stable/4411777>