



An-Najah National University

Faculty of Engineering & Information Technology

Computer Engineering Department

Graduation Project I

Smart Quality Inspection

A Web/Mobile Application to management and detection assembly line

Authors

Omar Dere

Ibrahim Arda

Supervisor

Dr. Anas Toma

**Presented in partial fulfillment of the requirements for Bachelor
degree in (Computer Engineer).**

May, 2022

Acknowledgements

“First, we’d want to express my gratitude to Dr. Anas Toma Who does not hesitate for a moment to provide assistance and instructions for us to complete the project. Furthermore, we would like to express our gratitude to all of the professors in Computer engineering department who assisted us by offering advice.

We’d also want to express my gratitude to our friends and families for their support. we’d want to express our gratitude to everyone who assisted and encouraged us to work on this project . We would not have been able to finish this project without their help.”

Disclaimer

This report was written by students (Omar Dere and Ibrahim Arda) at the Computer Engineering Department, Faculty of Engineering, An-Najah National University.

Therefore, has not been amended or modified as a consequence of assessment, with the exception of editorial adjustments, and it may include grammatical and content problems.

The opinions stated, as well as any conclusions or suggestions, are purely those of the students. An-Najah National University accepts no responsibility or liability for the effects of this report being used for something other than what it was intended for.

Contents

Abstract	6
1 Introduction	7
1.1 Background	7
1.2 Problem Statement	7
1.3 Significance	7
1.4 Objectives and Scope	7
2 Constraints and Earlier Coursework	9
2.1 Constraints Limitations	9
2.1.1 Time Limit	9
2.2 Earlier Coursework	9
3 Literature Review	10
3.1 A Quality Control Application on a Smart Factory Prototype	10
3.2 Quality inspection and maintenance: the framework of interaction	10
3.3 How is Our project Better?	11
4 Methodology	12
4.1 Architecture & Technologies Utilized	12
4.1.1 ReactJS	12
4.1.2 React-Bootstrap	12
4.1.3 Charts	13
4.1.4 PostgreSQL	13
4.1.5 OpenCV	13
4.1.6 Firebase	13
4.1.7 Java Spring Boot	14
4.1.8 Others	14
4.2 Features & Implementation	14
4.2.1 Customer page	15
4.2.2 Admin Page	21
4.2.3 Employee page	30
4.2.3 Image processing	33
4.2.3 Mobile application	36
5 Results & Discussion	40
5.1 Final Application	40
5.2 Project Outcomes	40
6 Conclusion & Future Work	4\
6.1 Summary	4\
6.2 Future Work	4\

List of Figures

Figure 1:Home page	15
Figure 2: about page	15
Figure 3:portfolio	16
Figure 4: our products.....	16
Figure 5: product details	17
Figure 6: press add to card without login	17
Figure 7: log in but not have account	18
Figure 8: sign up without data	18
Figure 9: sign up with data.....	19
Figure 10: sign in with wrong data.....	19
Figure 11: card	20
Figure 12: customer pressed on place order	20
Figure 13:dashboard page	21
Figure 14: last order.....	21
Figure 15: header	21
Figure 16: sidebar	22
Figure 17:CRUD employee	22
Figure 18: add new product.....	23
Figure 19: material enrolled.....	23
Figure 20: enrolled products.....	24
Figure 21: table	25
Figure 22: Employee Table.....	26
Figure 23: material table.....	26
Figure 24: profile details	27
Figure 25: edit profile.....	27
Figure 26: manager chatting 1	28
Figure 27: manager chatting 2	28
Figure 28: first chart.....	29
Figure 29: second chart.....	29
Figure 30: detection	30
Figure 31: product view	30
Figure 32: product details	31
Figure 33: sidebar employee.....	31
Figure 34: employee chat1	32
Figure 35: employee chat2	32
Figure 36: product ok.....	33

Figure 37: mask ok	34
Figure 38: defect area	34
Figure 39: mask defect area.....	35
Figure 40: color defects	35
Figure 41: sign in mobile	36
Figure 42: home page mobile and menu	37
Figure 43: charts mobile	38
Figure 44: report page mobile	39

Abstract

Defect detection was executed manually by human inspectors, naturally prone to fatigue, inattentiveness and biases. Later, manual inspection was augmented by rule-based machine vision technologies. Over the past decade defect detection has become increasingly technology-driven, building on advancements in artificial intelligence and image processing and big data. The use of smart cameras and related image processing-enabled systems is already helping manufacturers deliver high quality inspection in shorter cycles, reducing latency and costs, and setting new standards that are far beyond the capabilities of even the most experienced human inspectors.

In this project we design and implement software for process management on a production line to improve quality of the production process and detection if an error occurs. The software includes Customer page and Admin page, On the customer page, we market the products, allow the customer to purchase them, and display general information about the product line to the customer, in the admin page we divide it to three roles: Manager, Head and Employee each of them has specific powers to management production line and dealing with image processing. Because this out of our scope we be working with students from industrial engineering department.

Our project is built with ReactJS library for web that based on JavaScript programming language and Flutter for mobile that's based on Dart language. Furthermore, it includes application programming interface (API) made with Java spring boot to support the main application features and manage the database.

It uses 2 types of databases, a SQL database (PostgreSQL) is used for storing the main data of the application while real-time features like the chat messages is implemented using firebase messaging service.

We built image processing part using python.

Chapter 1

Introduction

1.1 Background

The presence of image processing on the production line helps to increase accuracy in detecting defects in products that may not be noticed by the worker standing in front of the production line. The worker also needs to know more about the product in which defects were discovered and to stop production at a certain percentage.

1.2 Problem Statement

Every employee whether he is worker or manager faces two main problems:

1. Predictive Maintenance: how to Improving the overall effectiveness of the equipment by predicting system failures and failures before they occur and continuously monitoring performance and deviations to optimize the scheduling of downtime.
2. Defect Detection: How to Improve Accuracy and Sensitivity of Fault Detection in Difficult, High Variation Environments.

Managers also face problems in managing production lines and how to present their products to customers. Employees also have problems communicating with each other.

1.3 Significance

Monitoring the product when it leaves the production line helps to discover defects, and in the event of their recurrence, production is stopped and the production line is maintained, which leads to reducing waste and costs, ensuring quality and increasing productivity

1.4 Objectives and Scope

This application does not support the maintenance of production lines and does not make predictions using artificial intelligence. The goal of this application is to discover defects in the products and inform the employee of this to take the necessary actions. In addition, the manager can manage the production line and market his products.

1. Chat Messages Feature: This feature enables employees to communicate with each other as well as enables the manager to communicate with them.
2. Card Feature: This feature is located on the customer's page and enables him to buy the products he needs.
3. Financial Feature: This feature enables the manager to know the amount of profit he earned during the month, the number of new customers who bought his product, and other financial matters.
4. Charts Feature: Through the charts, the employee can track the production process and the defects that are discovered.

Chapter 2

Constraints and Earlier Coursework

2.1 Constraints Limitations

2.1.1 Time Limit.

It took a long time to construct the entire program, which included learning technologies, Searching for a topic, creating the user interface, and implementing it on both sides. A Time wasn't enough to perform search of best libraries to handle some features.

2.2 Earlier Coursework

- Object-Oriented Programming (JAVA):

We used Java Spring Boot to build the backend as we used many of the features that we learned in this course.

- Database Design Management:

The database for this app was built completely using PostgreSQL. This course also presented database design principles and important concepts including dependencies, keys, and indexes.

- Web Development:

HTML, CSS and JavaScript were used as part of the front-end, this course taught me all of these languages. This course, on the other hand, covered (APIs) and how and Why they're constructed.

- Software Engineering:

All of the concepts, including as software requirements, architecture, user acceptability, were thoroughly addressed and proved to be highly helpful in the development of this application

- Critical Thinking Scientific Research:

It took a lot of study across websites to prepare for constructing this application, Which was practiced in this course. On the other hand, it was the first place where The talent of producing scientific papers was presented.

Chapter 3

Literature Review

Our application aims to discover defects in products that come out of the production line to make it easier for employees to work. However, it is not the first application that was Created to achieve the same purpose, there are some other different applications that are Already live. Some of these applications are presented below:

3.1 A Quality Control Application on a Smart Factory Prototype Using Deep Learning Methods

In this work, the implementation of visual quality control automation using a camera placed above the assembly line in an intelligent worker model is proposed. The product is detected in an image obtained from an assembly line and then rated as “good” or “unacceptable” using deep learning methods [1].

Advantages

Using a camera placed above the assembly line and informing the worker of any defect in the product as soon as it is produced, which leads to improving the quality of production.

Disadvantages

Using deep learning requires time and a large processor. The same goal can be achieved using image processing

3.2 Quality inspection and maintenance: the framework of interaction

This paper discusses that, under certain inspection procedures, we may determine whether a product's quality conforms to specifications or not [2].

3.3 How is Our project Better?

Our project was built with what other applications offer and what they don't offer in mind.

Our project is designed to have most of the important features offered by other applications in addition to designing a management system so that the manager can simply manage the plant and production line and we also designed a page for customers so that they can see and buy our products. Finally, we did our best to get rid of any issues

Featured with other apps.

Chapter 4

Methodology

4.1 Architecture & Technologies Utilized

4.1.1 ReactJS

React is considered a flexible free open-source front-end JavaScript library that is used to Develop fast and interactive web applications [3]. The library was made by Meta (Facebook Formerly) and has a vast community and a very large collection of libraries and reusable Components.

We decided to use React for many different reasons:

1. React offers a clean and easy to understand and utilize syntax that's called JSX Syntax. It is very similar to HTML but the main idea is that it allows the developer To inject JavaScript code inside the front-end design, which allows for more freedom Over controlling the component.
2. Reusable components is one of React's strongest features. You don't need to recreate A layout you have made earlier, instead, you only need to create it once and import It everywhere else where it is needed.
3. A vast collection of strictly free libraries. This one other important feature of React that allows the developers to use the fruits of other experienced developers By utilizing their libraries. This doesn't necessarily mean it is a good thing, but it Serves the purpose of building projects in the fastest interval.

4.1.2 React-Bootstrap

React-Bootstrap replaces the Bootstrap JavaScript. Each component has been built from scratch as a true React component, without unneeded dependencies like jQuery.

As one of the oldest React libraries, React-Bootstrap has evolved and evolved alongside React, making it an excellent choice as the basis for our user interface.

Built with compatibility in mind, we embrace our bootstrap core and strive to be compatible with the world's largest UI ecosystem.

By relying entirely on the Bootstrap stylesheet, React-Bootstrap just works with the thousands of Bootstrap themes you already love [4].

4.1.3 Chart.js

Chart.js is a community maintained open-source library (it's available on GitHub) that helps you easily visualize data using JavaScript. It's similar to Chartist and Google Charts. It supports 8 different chart types (including bars, lines, & pies), and they're all responsive [5].

We used chart.js for many different purposes:

- 1- We used chart.js because we use it easily in react by using react-chartjs-2 library
- 2- Because it provides us with several types of charts.

4.1.4 PostgreSQL

PostgreSQL is the leading open source relational database. it's an incredibly reliable, easy to use, and multi-talented database [6].

We used PostgreSQL for many different purposes:

1. PostgreSQL can be easily interfaced with Java Spring Boot for easier data handling
2. PostgreSQL makes returning updated documents or query results significantly easier.

4.1.5 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products [7].

We decided to use OpenCV because this library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and identify objects.

4.1.6 Firebase

Firebase, according to their website, is a platform developed by Google for creating mobile And web applications. It was originally an independent company. [8] Firebase is a platform That allows developers to create apps for iOS, Android, and the web, as well as tools for Tracking analytics, reporting, and correcting app errors, as well as marketing and product Experimentation.

We used Firebase for many different reasons, including:

1. Manages all data real-time in the database. As a result, data is easily and quickly Transferred to and from the database. As a result, it's what we're using to create Chat messaging.
2. With only a few lines of code, you can give any social networking login.

5.1.7 Java Spring Boot

Spring is one of the most widely used frameworks for developing enterprise applications, providing a robust programming and configuration model. The creation of this framework was facilitated by the desire to simplify the development of applications on the popular Java EE technology stack from Oracle, which was very complex and difficult to use at the time.

Unlike other frameworks, spring focuses on several areas of application functionality and provides a wide range of additional functions for them [9].

One of the main advantages of Spring Framework is its use of the Dependency Injection pattern. DI makes it much easier to implement the functionality that applications need, and allows to develop loosely coupled, more general classes.

Some advantages of Spring Framework:

1. Spring Framework can be employed on all architectural layers used in the development of web applications
2. Uses the very lightweight POJO model when writing classes.
3. Allows you to freely link modules and easily test them.
4. Supports declarative programming.
5. Eliminates the need to independently create factory and singleton classes.
6. Supports various configuration methods.
7. Provides middleware-level service

5.1.8 Others

There are multiple other technologies used, including:

1. React Router which is used to build a single-page web application with navigation. [10]
2. JWT (JSON Web Token) which is primarily used for user login authentication. [11]
3. by crypt which is used to hash user passwords. [12]
4. axios which is used to do API calls from the client side. [13]

4.2 Features & Implementation

The application provides many features and a lot to talk about. This section will focus on the application features and the role of each library in the implementation.

The user can enter the site as a customer, manager (administrator) or employee. When the user logs in, the page is shown to him according to his role. By defining the role of the user we can promote the employee or change his role by modifying the role only without the need to make major modifications.

4.2.1 Customer Page

When a customer enters our site for the first time, he can see several pages about the product line but he can only interact with them after logging in

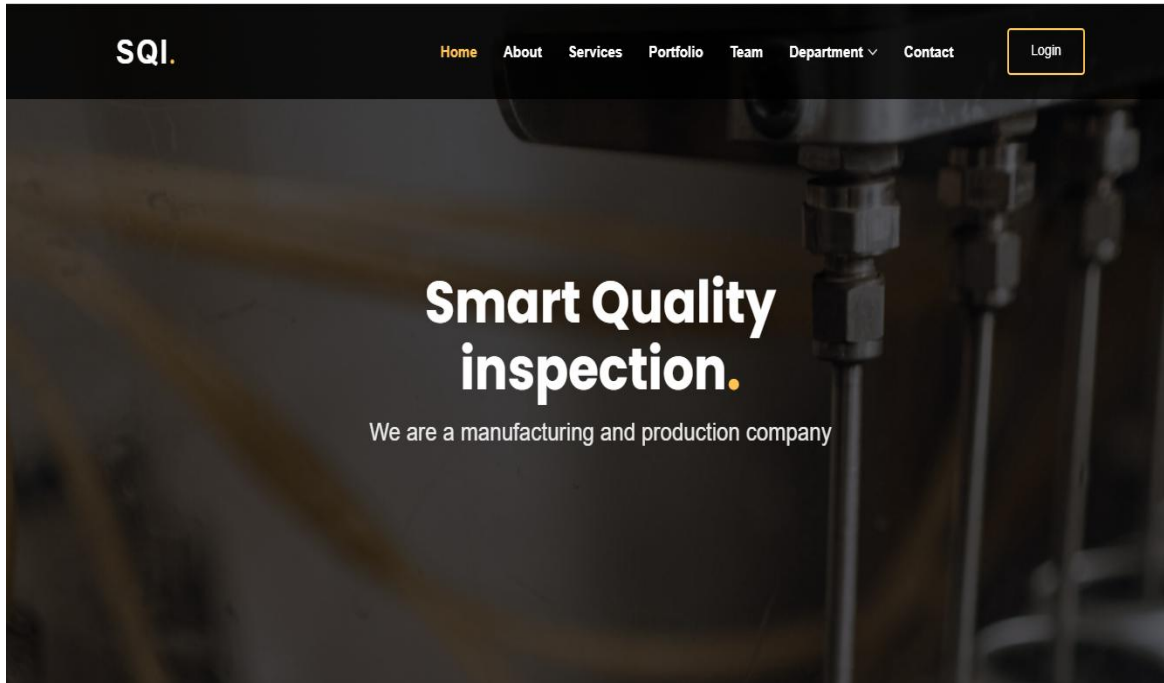


Figure 1: Home page

In about page the customer can see some information about factory

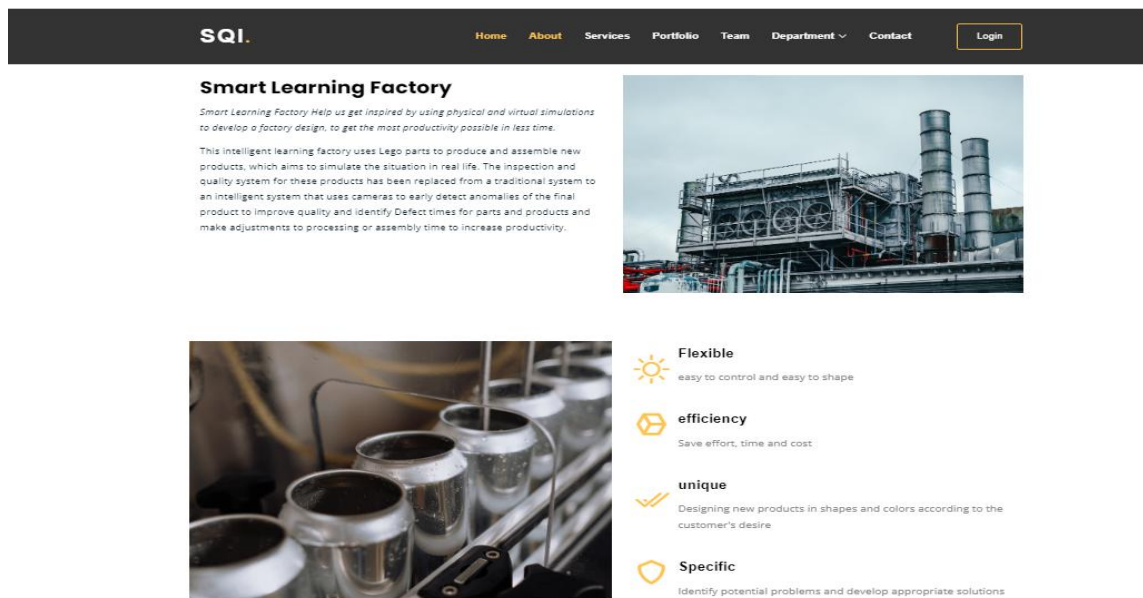


Figure 2: about page

In portfolio page the customer can see the products without details

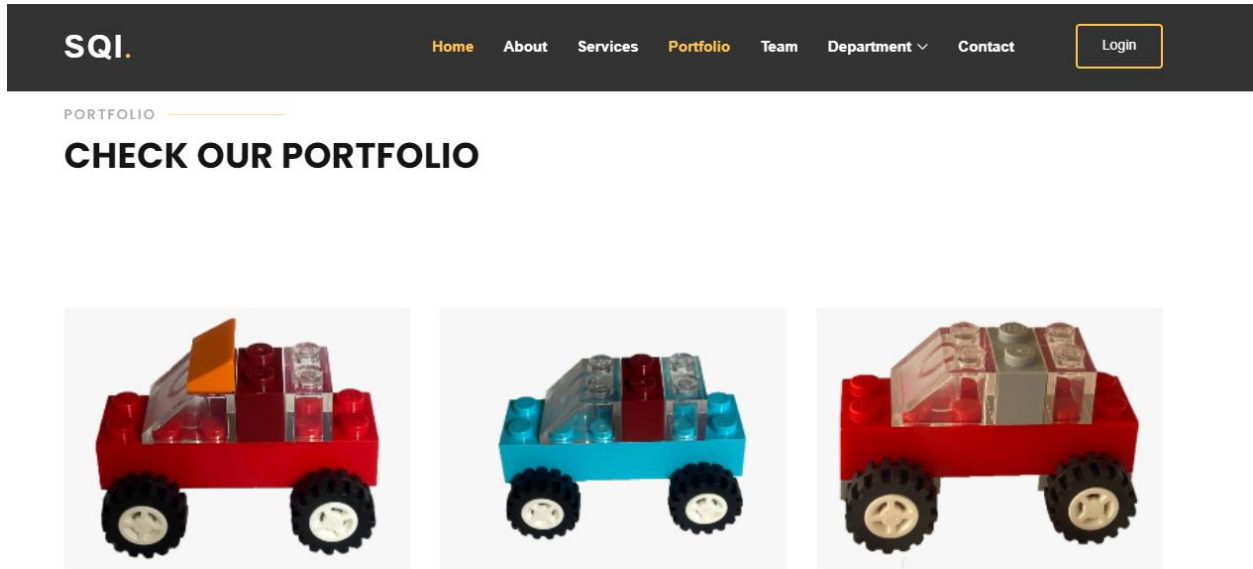


Figure 3: portfolio

In our product page the user can see the products with some details

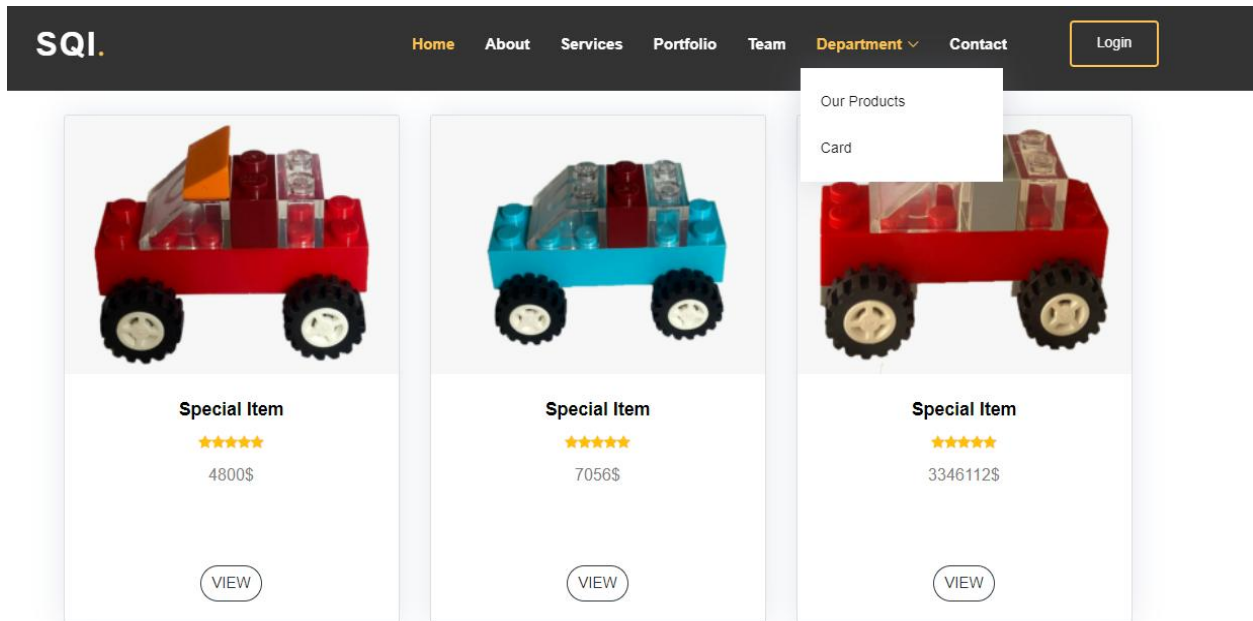


Figure 4: our products

If the customer click on VIEW button then the site display page contain all details about this product

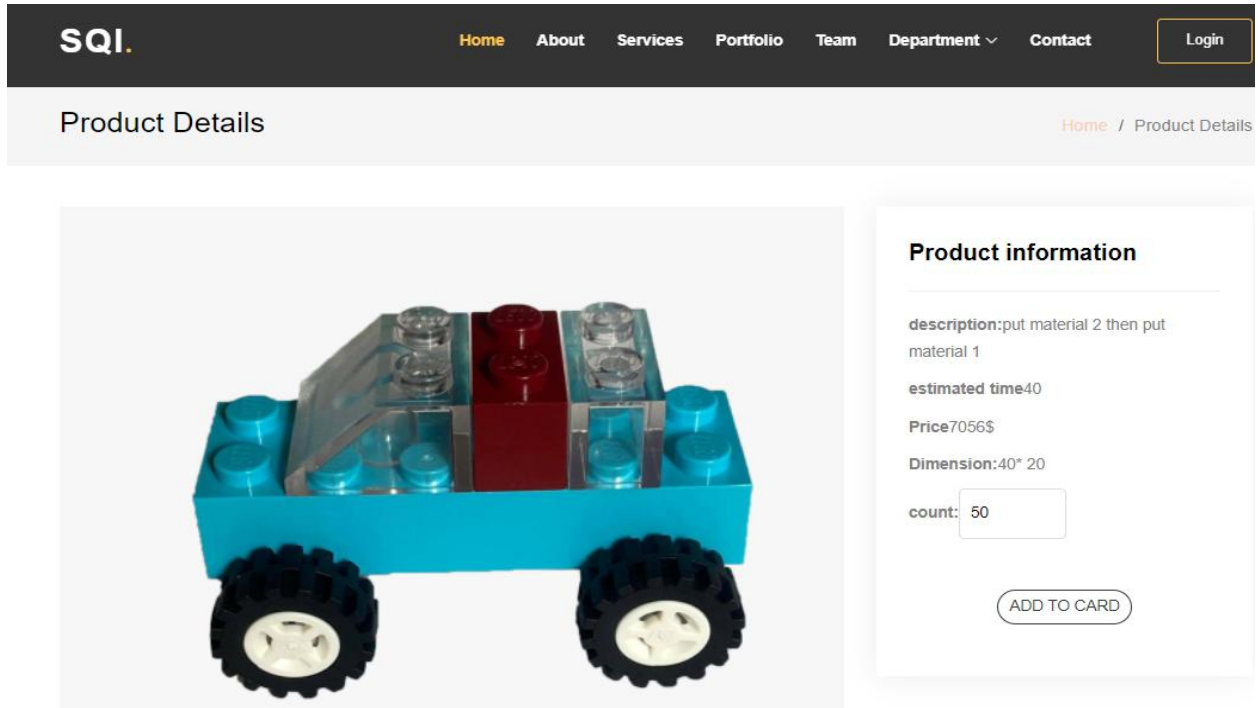


Figure 5: product details

After selecting the count that the customer wants to buy and then pressing add to card, the site asks the customer to log in to the site and take it to the login page

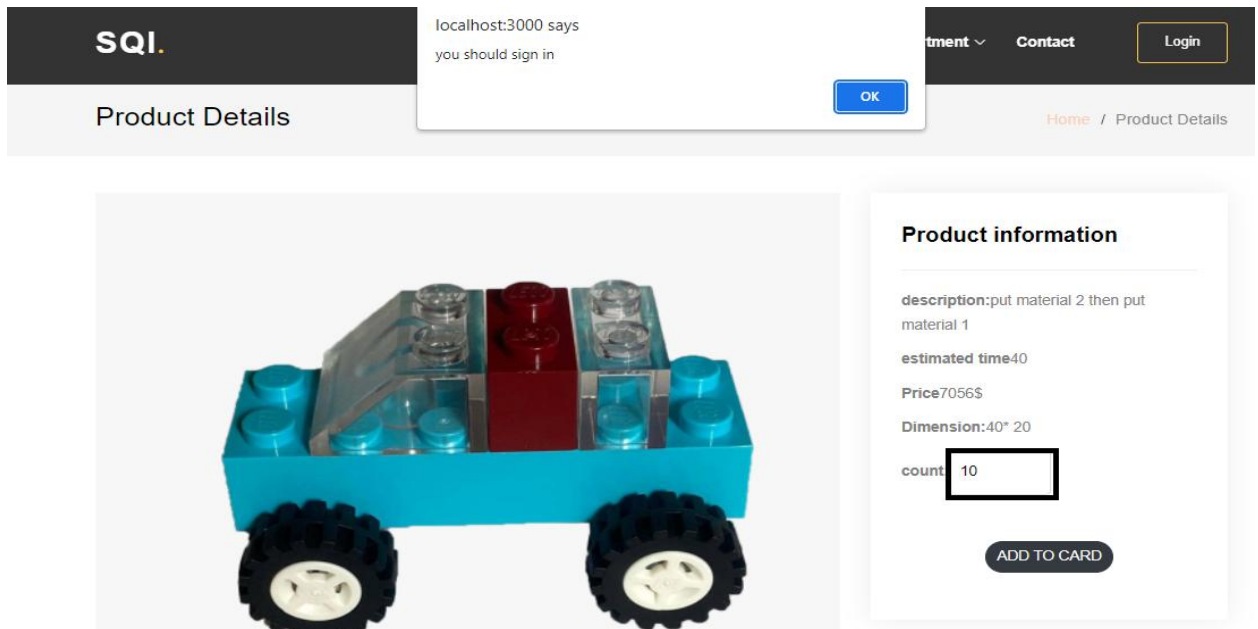


Figure 6: press add to card without login

But if the customer don't have account. He can signup

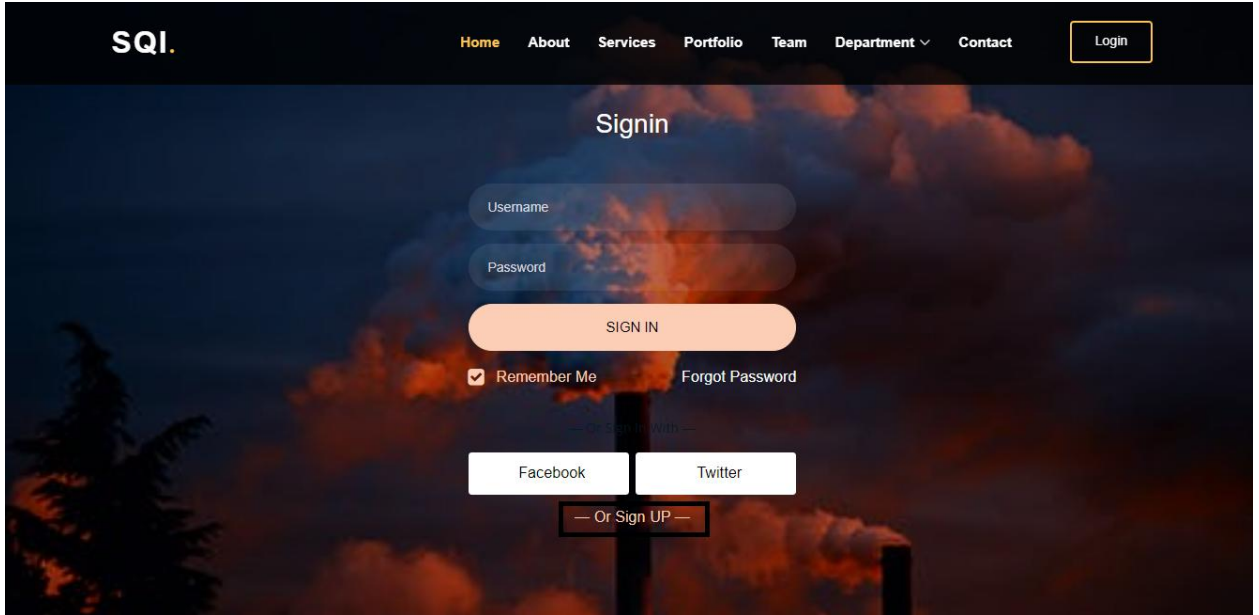


Figure 7: log in but not have account

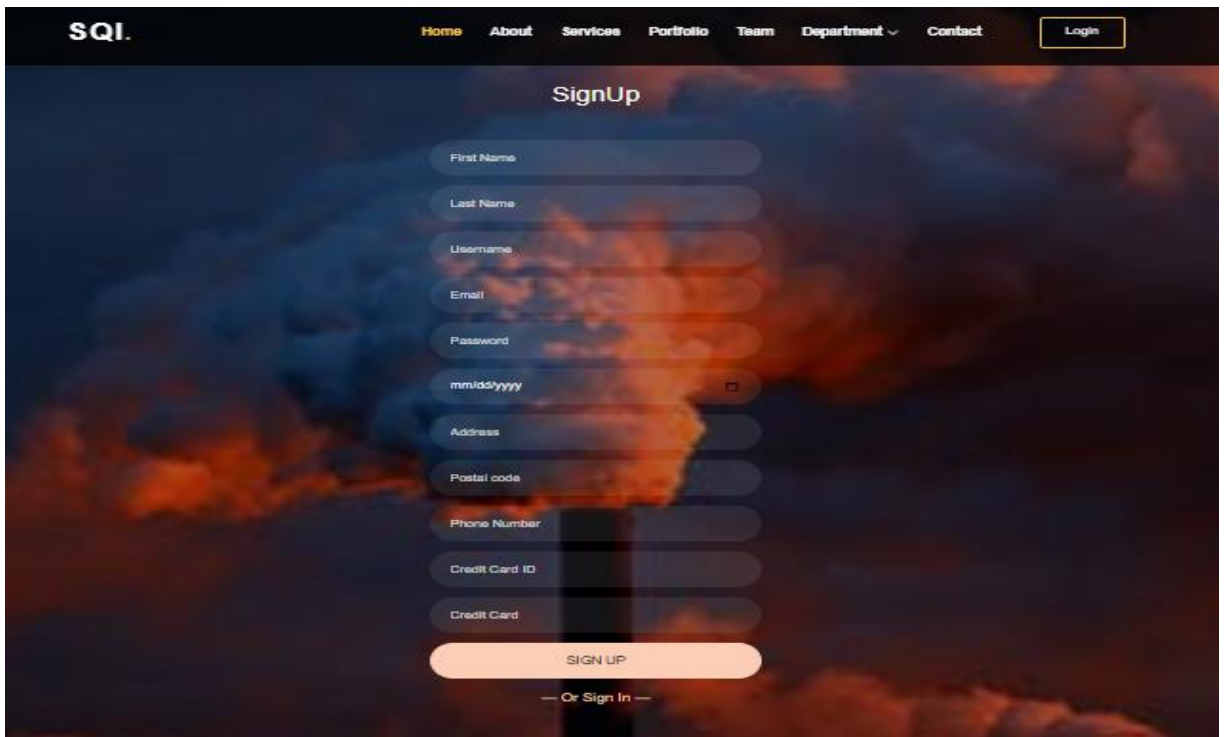


Figure 8: sign up without data

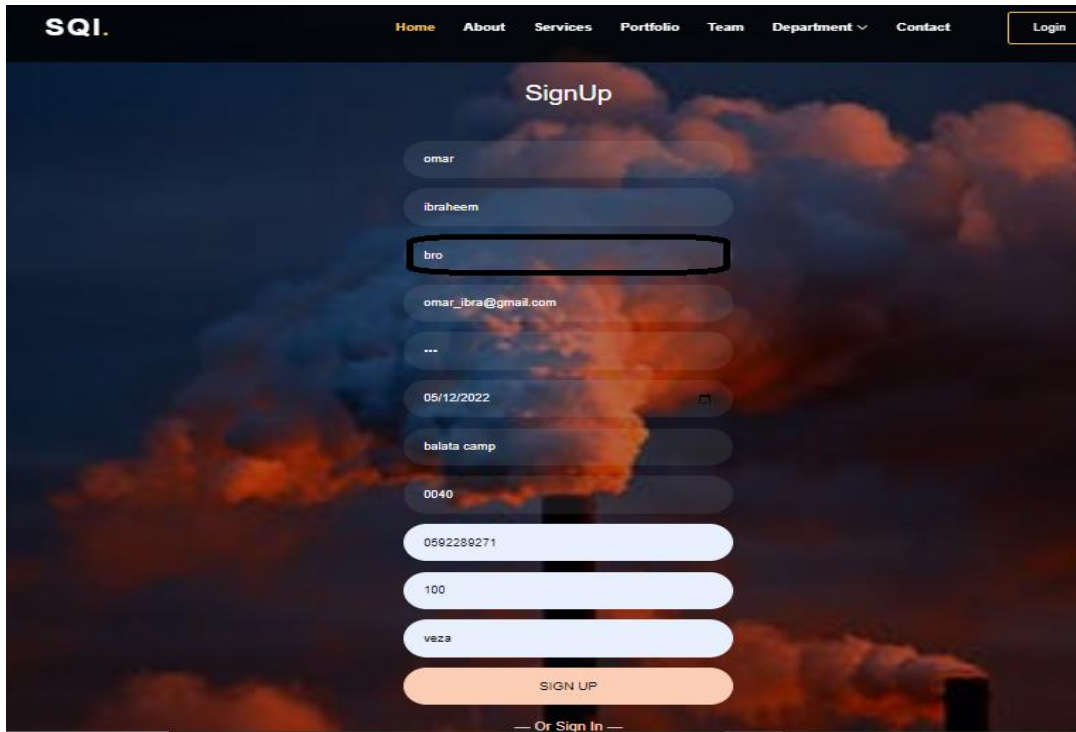


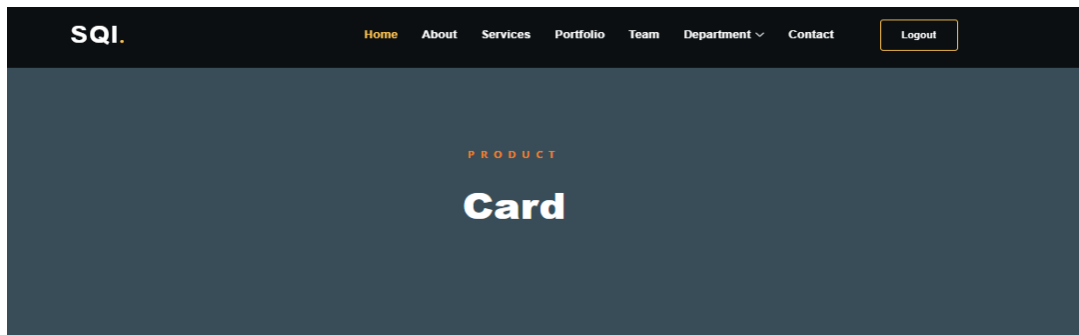
Figure 9: sign up with data



After sign up the customer can sign in to our site. if customer enter wrong username or password then warning message appear .



Figure 10: sign in with wrong data

Now, customer can add product to card. he can add many product and select The quantity of the product he wants to buy, also he can update or delete products from card . after adding his products, he presses on place order to create his order .

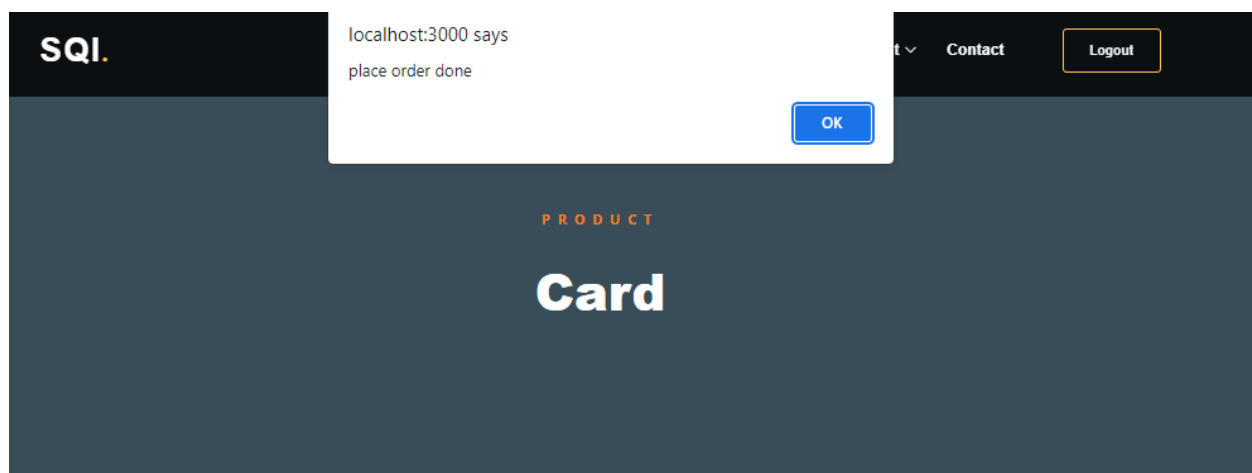




Product Image	Name	Price	Quantity	Total	Delete
	store1	\$97	<input type="text" value="2"/>	195	

Total	Price
Total:	\$195

Update Cart Place Order

Figure 11: card



Product Image	Name	Price	Quantity	Total	Delete
	store1	\$97	<input type="text" value="2"/>	195	

Total	Price
Total:	\$195

Update Cart Place Order

Figure 12: customer pressed on place order

4.2.2 Admin Page

The manager can login to site as admin because he has full responsibility to make any modification to the site. After admin logging in, the dashboard page will appear. This page contains general information about company as amount of profit, last sales and customer count in this month.

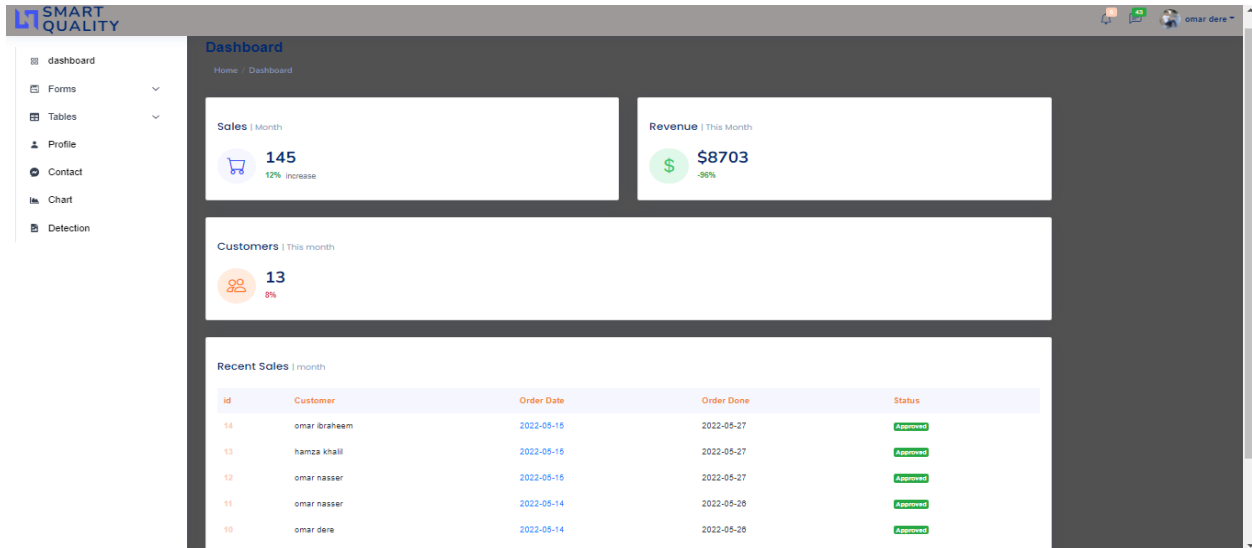


Figure 13: dashboard page

In last order we can see the order that we added a little while ago

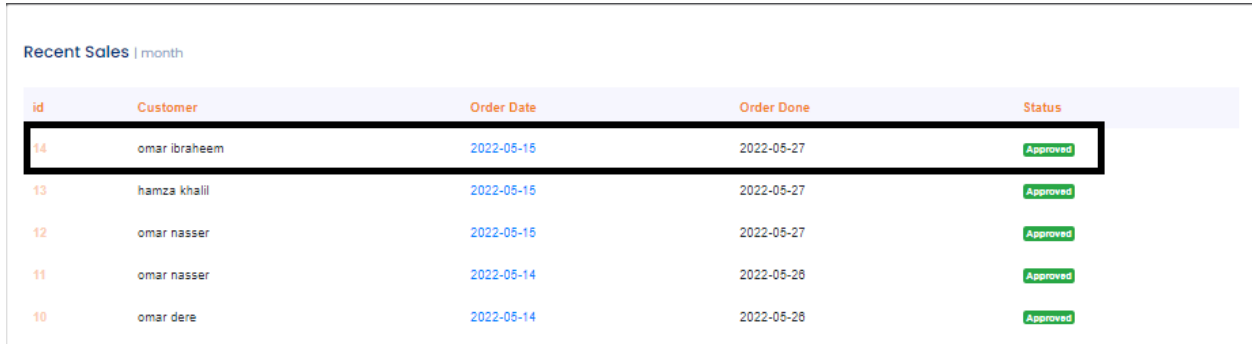


Figure 14: last order

All pages contain header which contain last notification, last messaging and profile details



Figure 15: header

In sidebars you can see forms, it's containing many pages Employee, customer, sublayer, store, product, material and department. From this page the Admin can perform CRUD operation on database

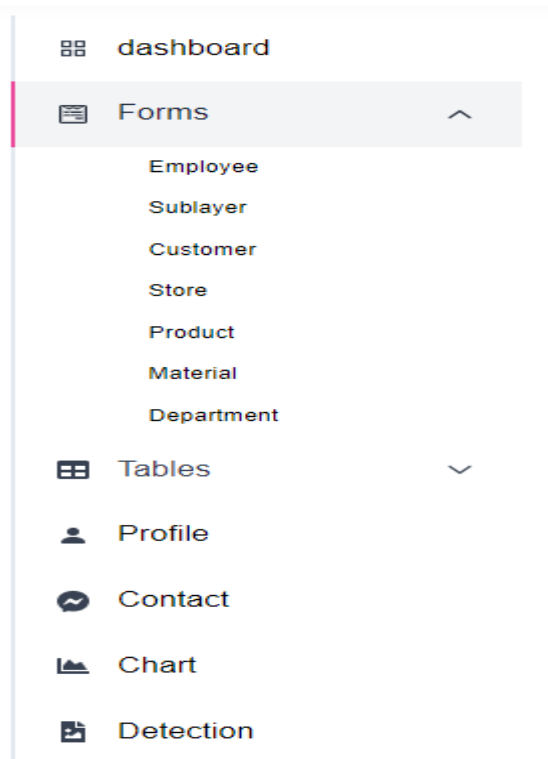


Figure 16: sidebar

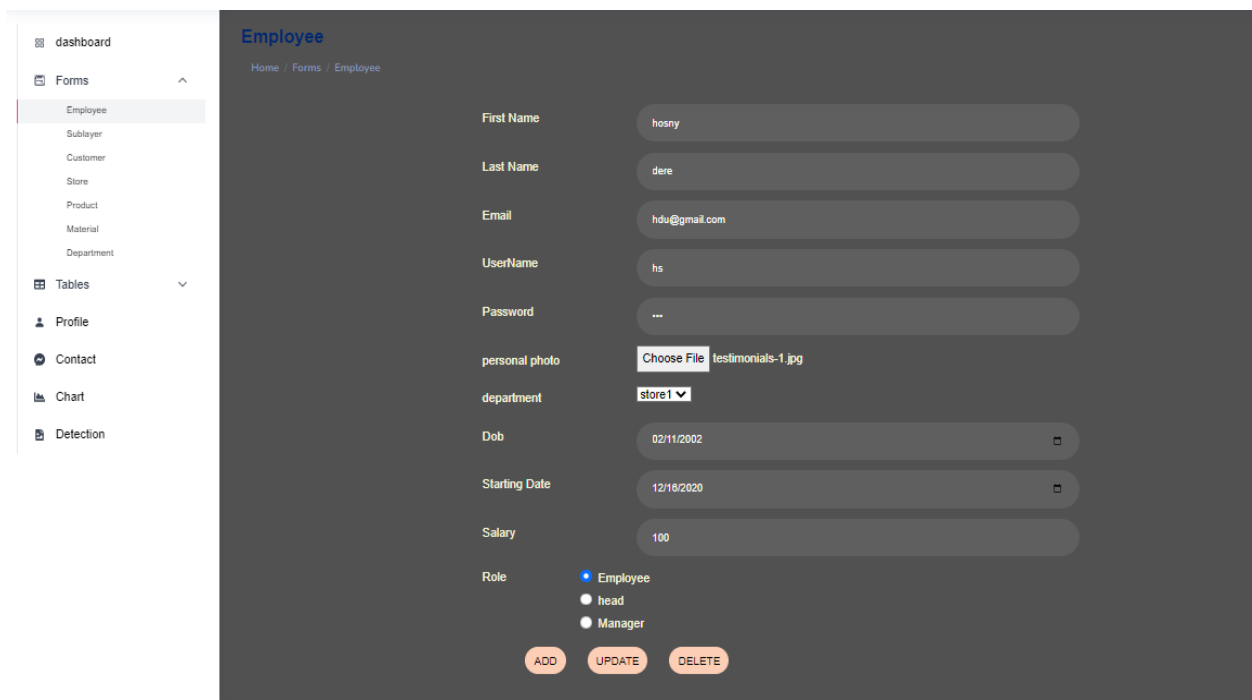


Figure 17: CRUD employee

If the manager need add new products, firstly fill all details about product

The screenshot shows a 'Products' form with the following fields and values:

- height: 100
- width: 100
- Time to end: 50
- percent: 0.2
- Quantity: 100
- product photo: Choose File p1.jpeg
- Description: put material 1 and material 2 then ...
- warehouse: store1

A 'NEXT' button is located at the bottom of the form.

Figure 18: add new product

If the administrator presses the Next button, the enrolled Materials page will appear. This page contains all the materials to allow the administrator to select the materials needed to build the product.






The screenshot shows a list of enrolled materials. Each material entry includes an image, a name, and several details:

- Material 1: image of a red toy car, name 'material 1', count: 100, availability: 10, description: this material can used to produce many pr...
- Material 2: image of a blue toy car, name 'material 2', count: 200, availability: 10, description: this material can used to produce many pr...
- Material 3: image of a red toy car, name 'material 3', product id: 1, estimatedTime: 50, height*width: 100*100, description: put material 1 and material 2 then ...
- Material 4: image of a red toy car, name 'material 4', product id: 5, estimatedTime: 50, height*width: 100*100, description: put material 1 and material 2 then ...
- Material 5: image of a blue toy car, name 'material 5', product id: 5, estimatedTime: 50, height*width: 100*100, description: put material 1 and material 2 then ...
- Material 6: image of a red toy car, name 'material 6', product id: 7, estimatedTime: 50, height*width: 100*100, description: put material 1 and material 2 then ...
- Material 7: image of a red toy car, name 'material 7', product id: 8, estimatedTime: 50, height*width: 100*100, description: put material 1 and material 2 then ...

At the bottom of the page, there are three input fields and an 'ADD' button.

Figure 19: material enrolled

If the administrator presses the Add button, the enrolled product page will appear. This page contains all the products to allow the administrator to select the product needed to build the new product.

	store1 product id :4 estimatedTime :50 height*with :100* 100 description :put material 1 and material 2 then ..
	store1 product id :5 estimatedTime :50 height*with :100* 100 description :put material 1 and material 2 then ..
	store1 product id :7 estimatedTime :50 height*with :100* 100 description :put material 1 and material 2 then ..
	store1 product id :8 estimatedTime :50 height*with :100* 100 description :put material 1 and material 2 then ..
	store1 product id :6 estimatedTime :50 height*with :100* 100 description :put material 1 and material 2 then ..

8 6 2 [ADD](#)

Figure 20: enrolled products

In sidebars also you can see tables, it's containing many pages Employee, store, sublayer, product and material. Table this page the Admin can see all information about this.

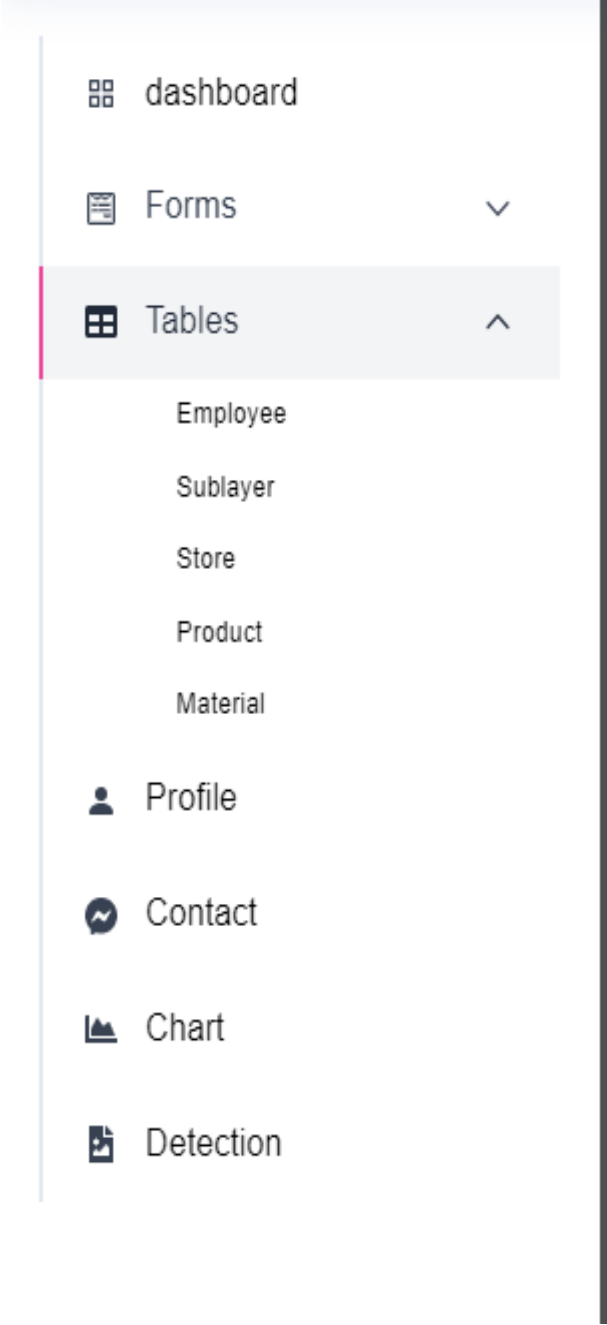


Figure 21: table

Employee

Home / Tables / Employee

Employee

this page contain information about Employees

search ...

Id	Name	Role	Email	Start Date	Salary
1	omar dere	Admin	omar@gmail.com	2010-10-09	100
3	ibraheem arda	employee	ibraheem@gmail.com	2022-05-20	100
4	omar dere	employee	omarder80@gmail.com	2022-05-17	100
2	omar dere	Admin	omarder2000@gmail.com	2022-05-17	100
5	hosny dere	employee	hdu@gmail.com	2020-12-15	100

Figure 22: Employee Table

Materials

Home / Tables / Materials



	<p>id:2</p> <p>availability: 10</p> <p>count: count</p> <p>ware house: store1</p>	<p>Description</p> <p>this matareial can used to produce many product</p>
	<p>id:1</p> <p>availability: 10</p> <p>count: count</p> <p>ware house: store1</p>	<p>Description</p> <p>this matareial can used to produce many product</p>

Figure 23: material table

In profile page the admin or employee can see all his information and you can edit it.

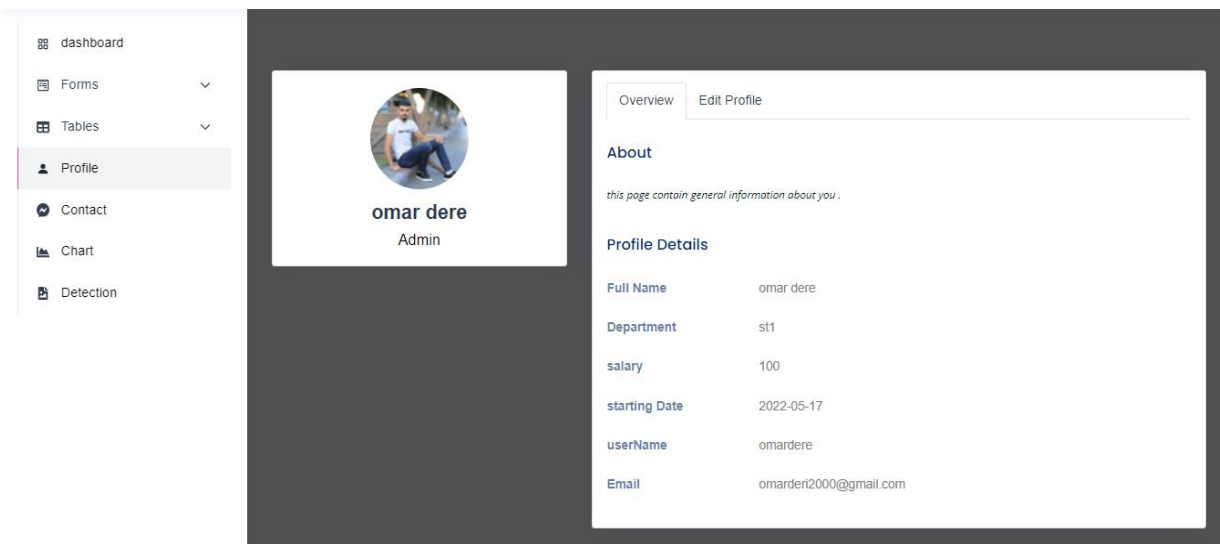


Figure 24: profile details

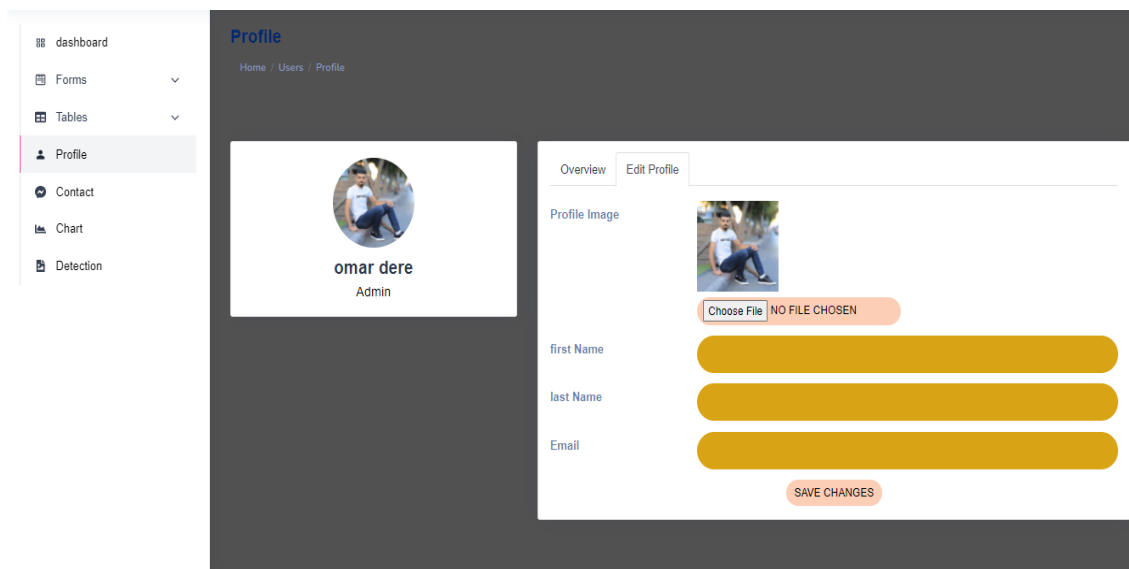


Figure 25: edit profile

Contact page contain chatting between all employees if the employee sent message all other employee received this message

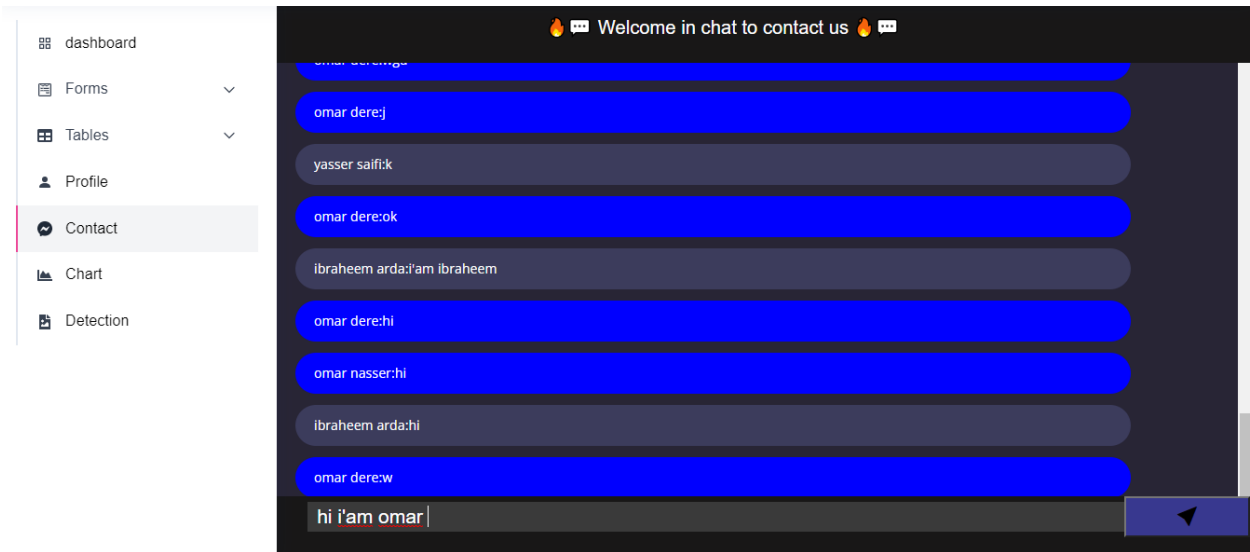


Figure 26: manager chatting 1

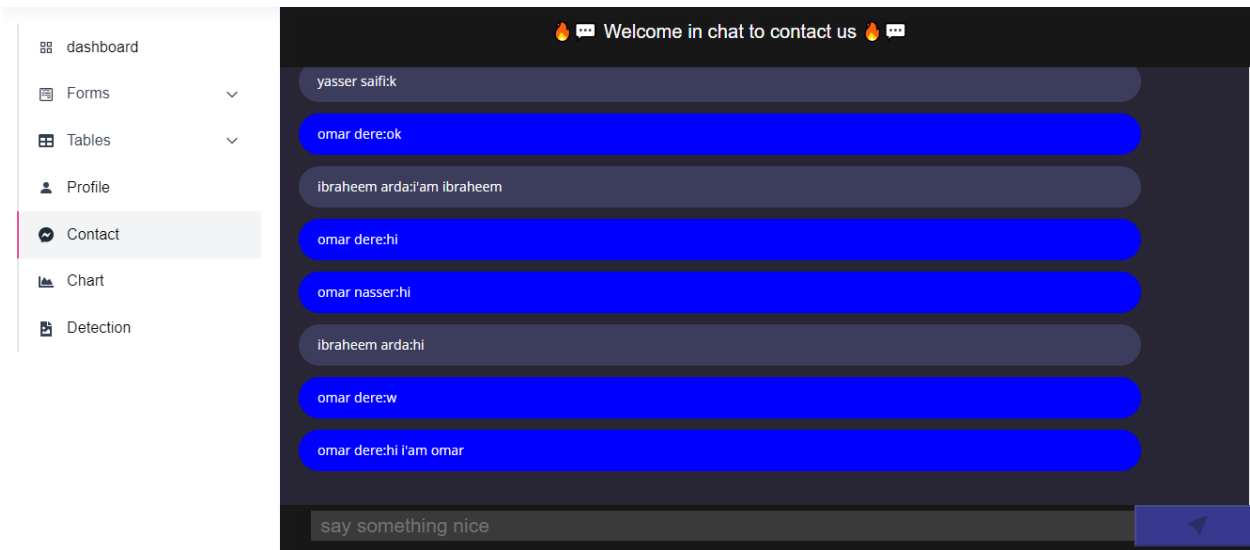


Figure 27: manager chatting 2

Charts page contain two charts: the first chart displays the number of products each 20 minutes and the second chart display the number of profits in last months.

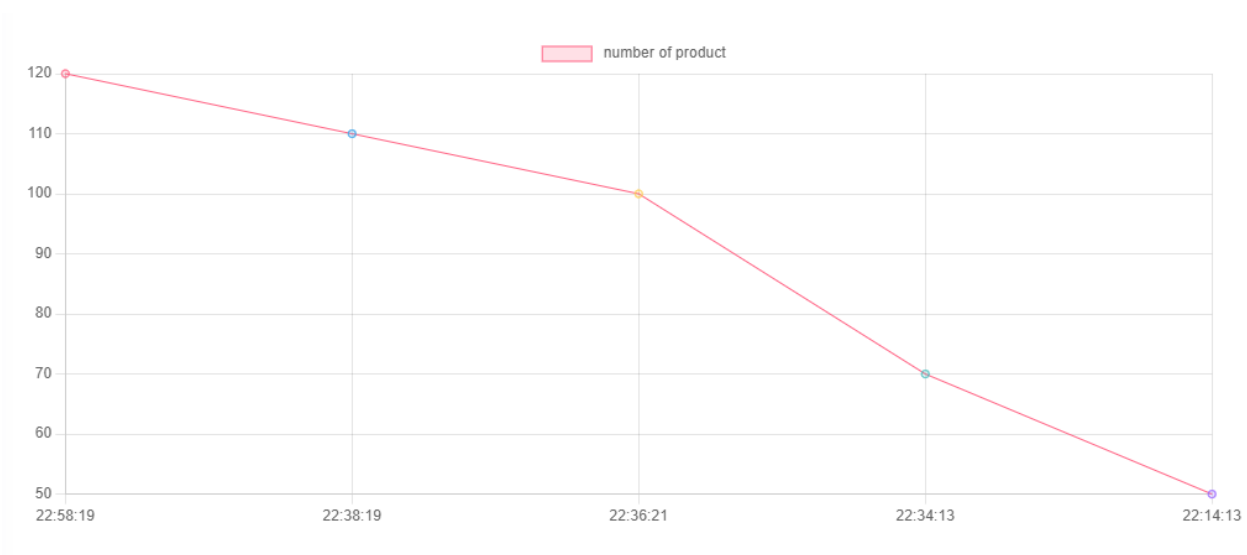


Figure 28: first chart

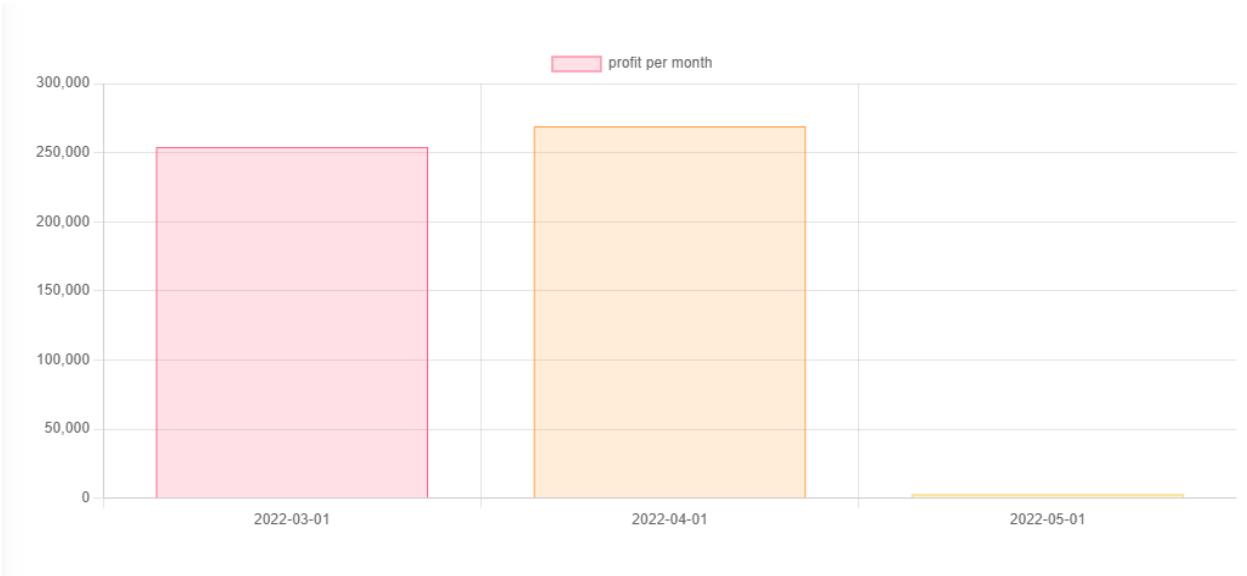


Figure 29: second chart

Finally detection page, in this page the employee can select the product to be produced on the assembly line. After select it, the product will activate and other products will deactivate. In image processing part we choose the active product from database and detection it.

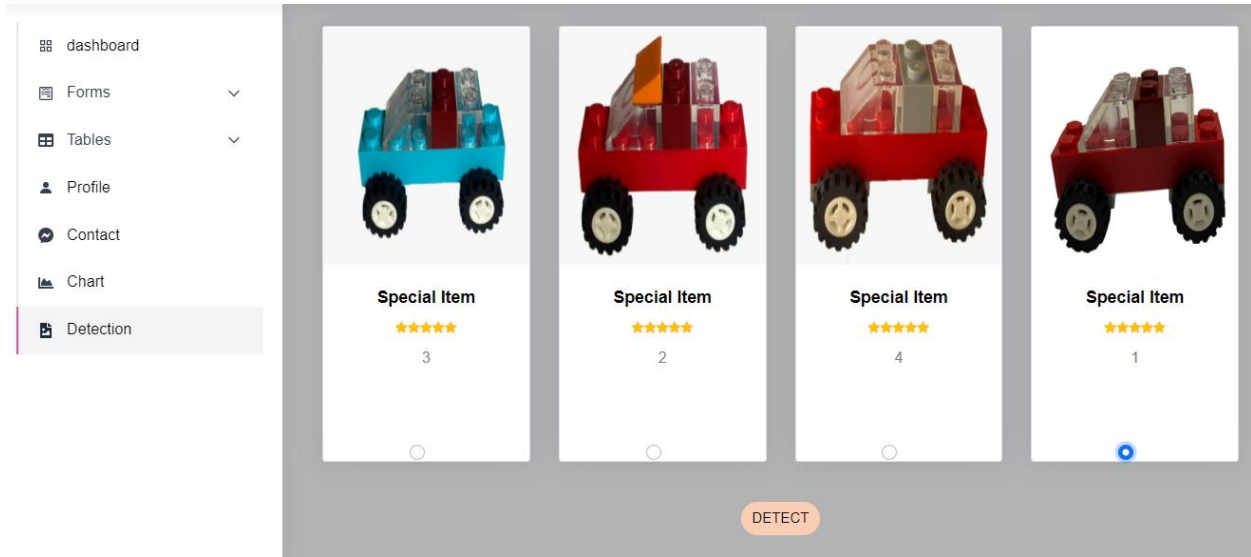


Figure 30: detection

4.2.3 Employee Page

If the user has an employee role when logging in to the site, the product page will appear first. This page contain all product and if employee press on view page the product details page will appear this page contain all information about this product and how the employee can production it and what the materials who needed it.

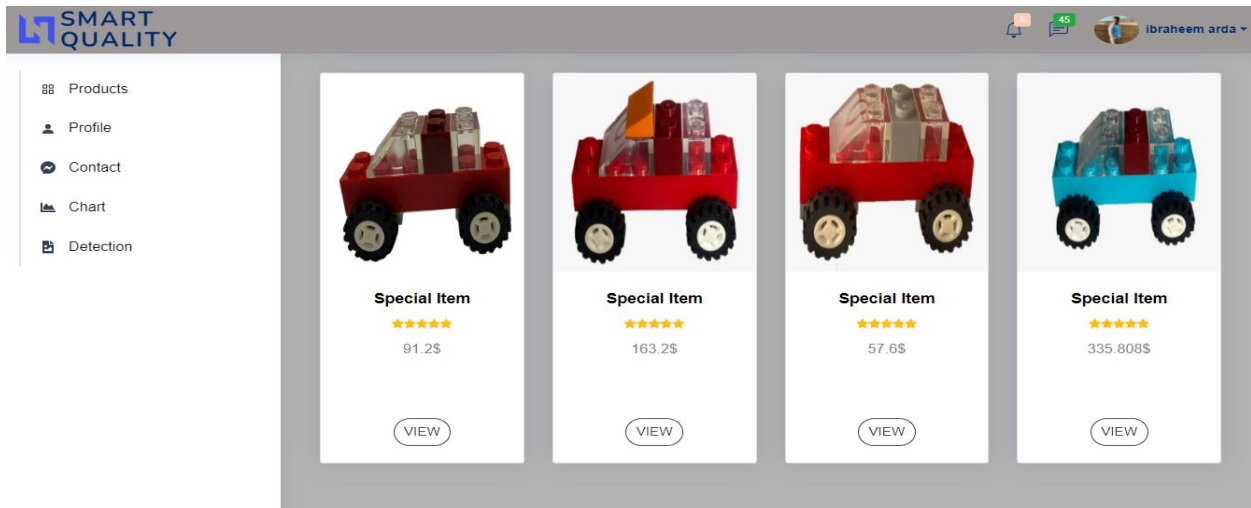


Figure 31: product view



Product information

description:put material 1 and material 2 then ..

estimated time:50

Price:91.2\$

Dimension:100* 100

Count:50

Material nedded:

id:1 , count:2

id:2 , count:3

Figure 32: product details

Side bar contains only the pages that the employee can control

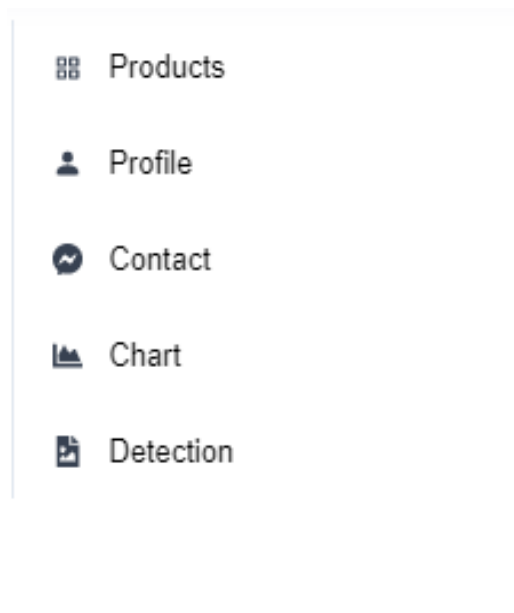


Figure 33: sidebar employee

Contact page display chatting, you can see the messages that manager sent it and the employee can send message to all employee

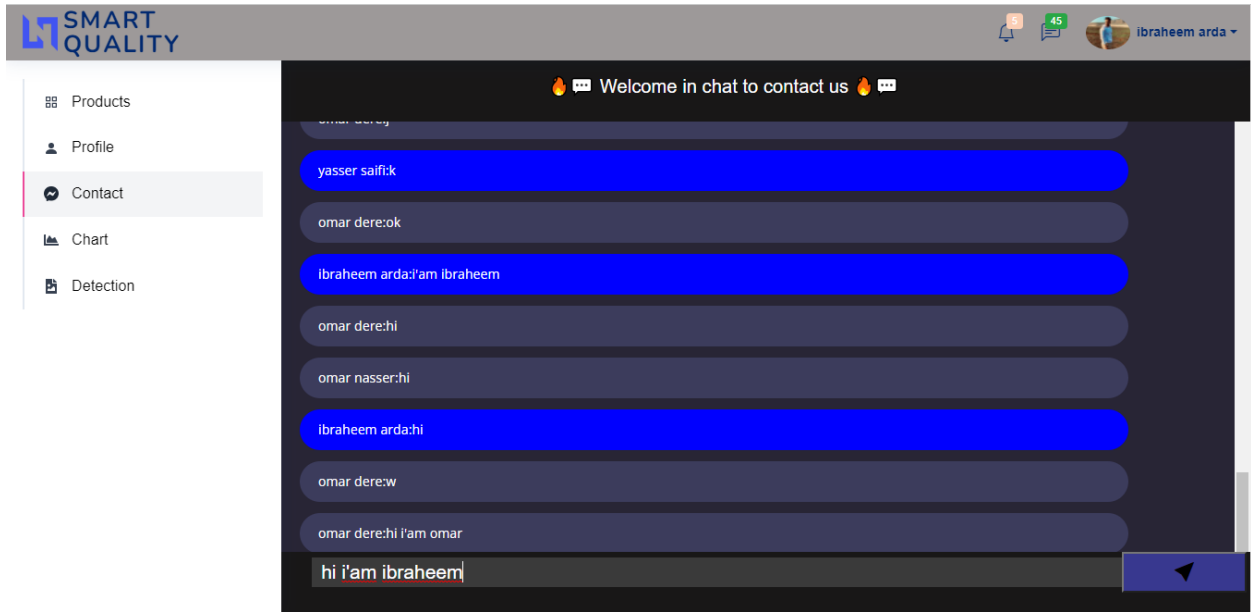


Figure 34: employee chat1

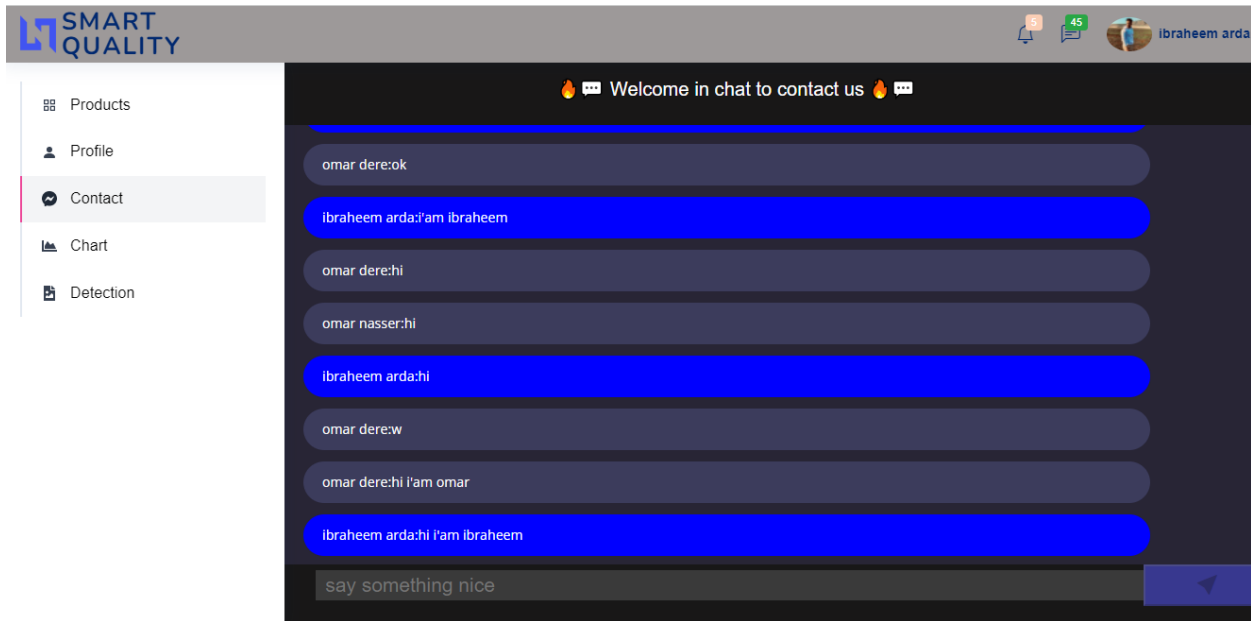


Figure 35: employee chat2

4.2.4 Image processing

In image processing part we put the product in front of the camera. We processing real time video and detect defects if the area of product change or if the color of material change. If product have an area similar to that stored in the Database and the color ok the number of ok in data base increase by 1.

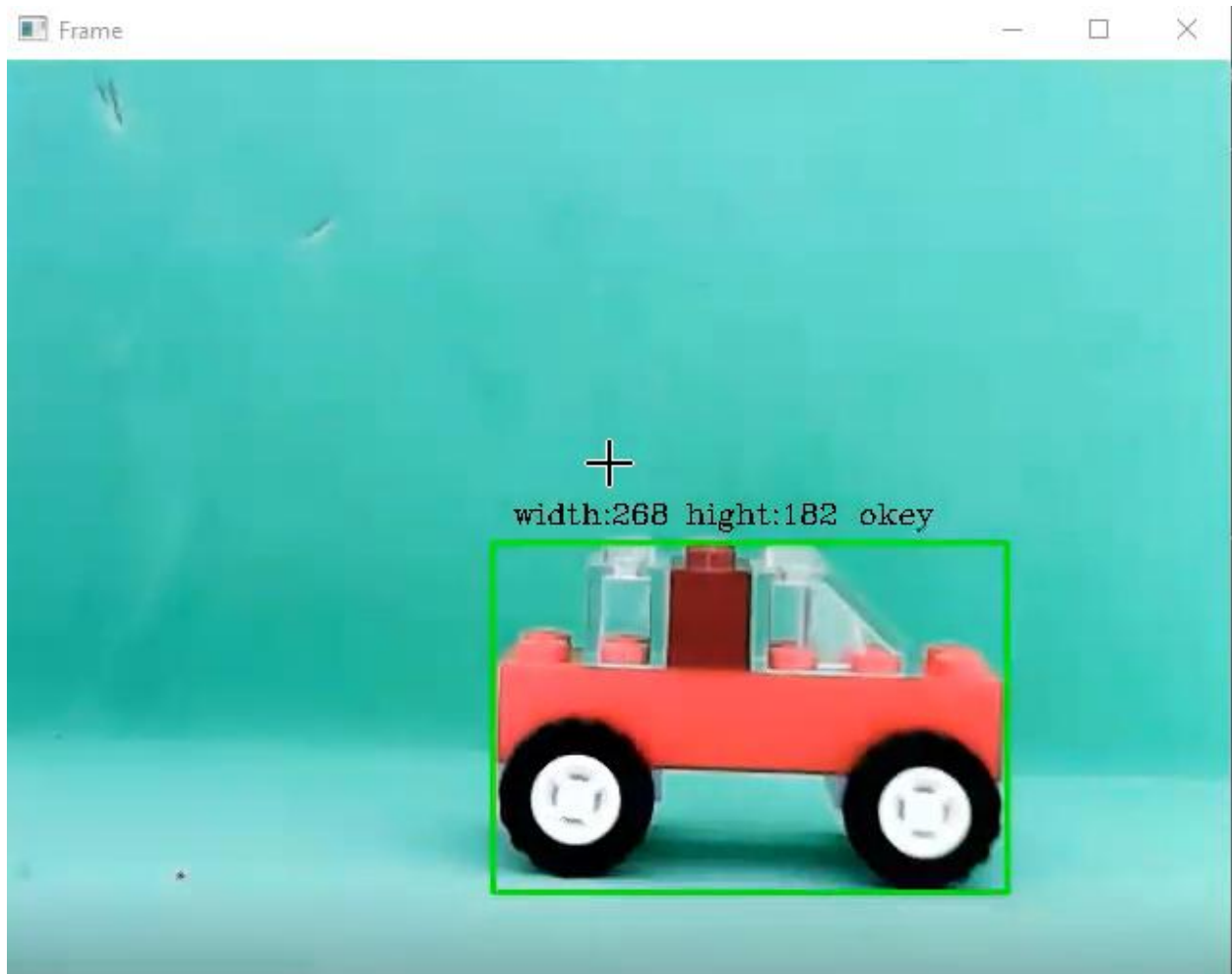


Figure 36: product ok



Figure 37: mask ok

If the employee forgot to put a certain part of the product or made a mistake and put another part in a different color then the number of defects in database increase by 1. We used the number of ok and the number of defect to display charts and notification to admin.

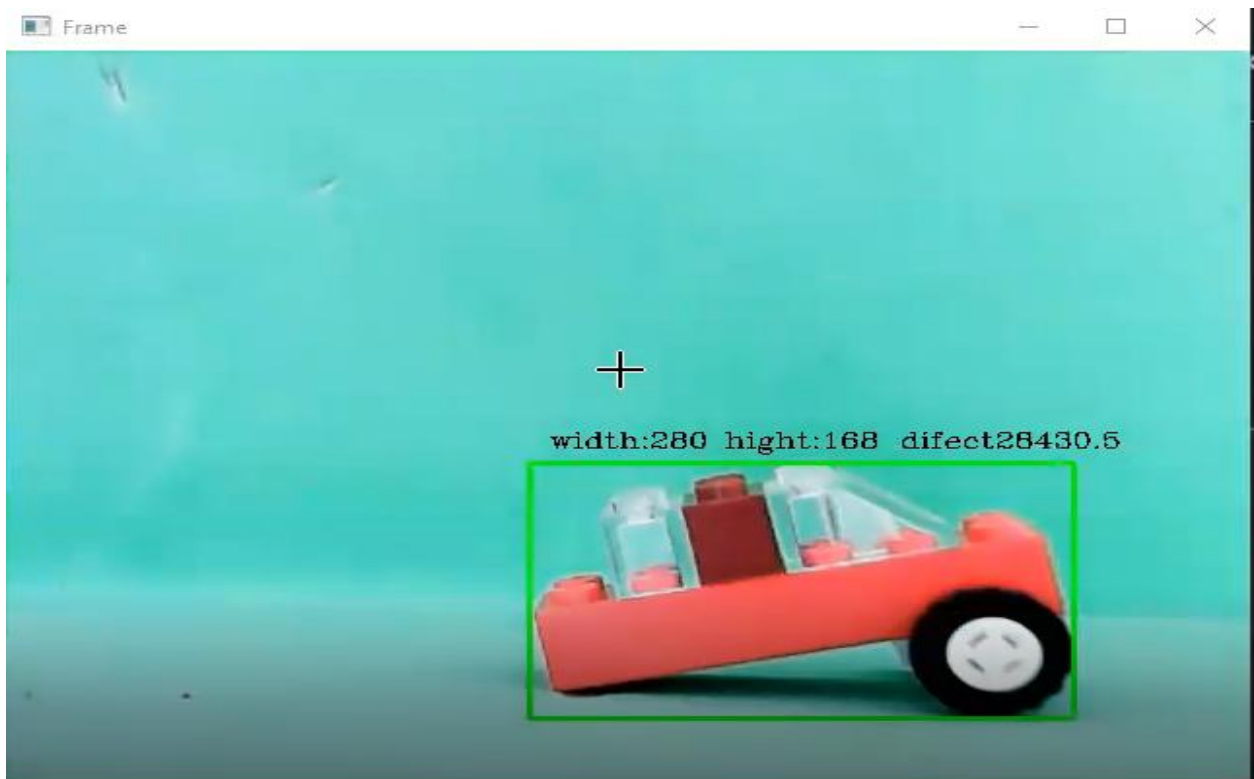


Figure 38: defect area



Figure 39: mask defect area

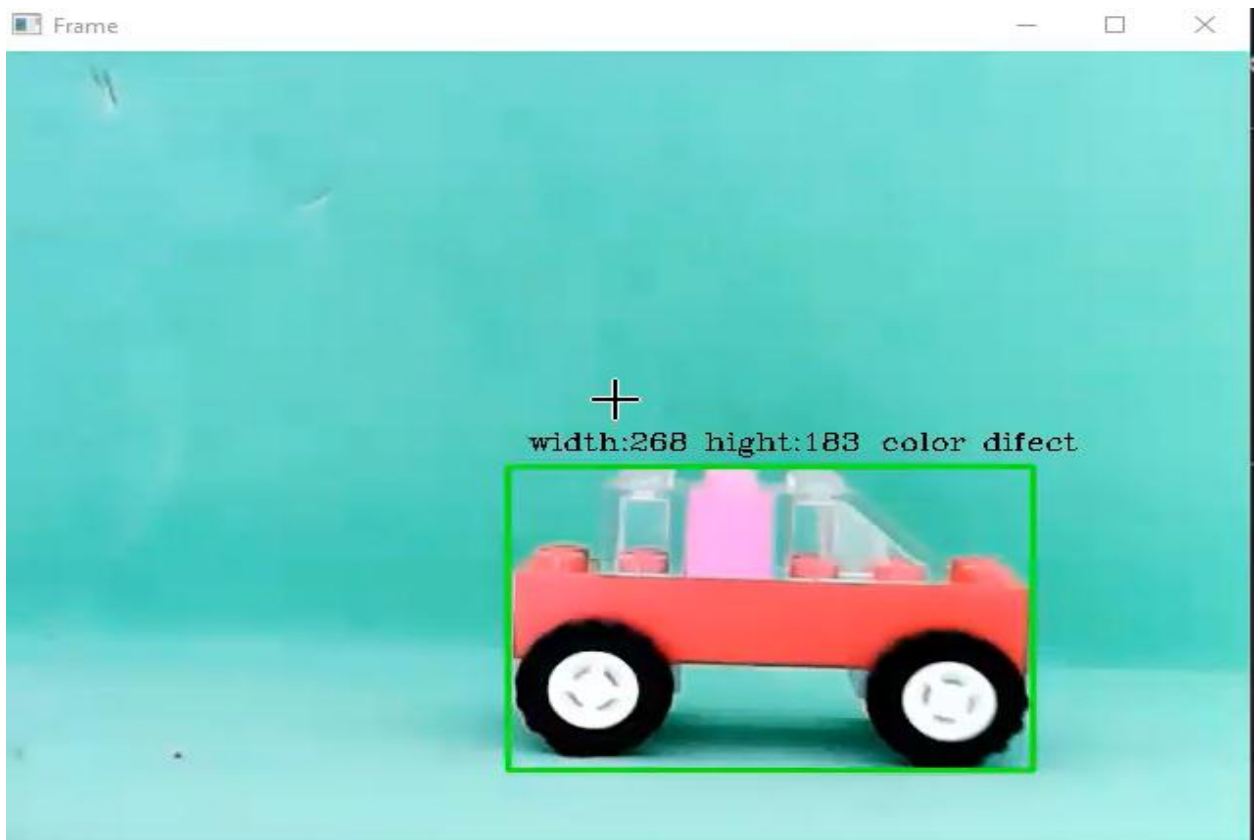


Figure 40: color defects

4.2.5 Mobile application

Because the website has all the features, we designed a mobile app to display charts and report to the employee.

The mobile app contains sign in page to allow the administrator to access the site.

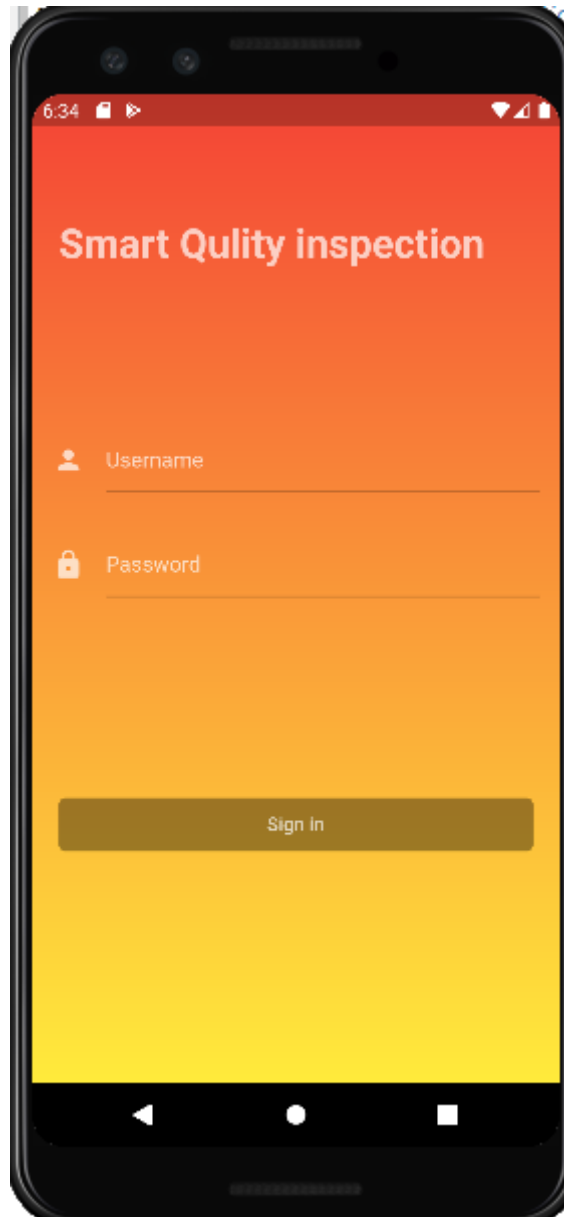


Figure 41: sign in mobile

After log in the main page appears, and if admin pressing the menu button, a menu appears containing pages and information about the administrator.

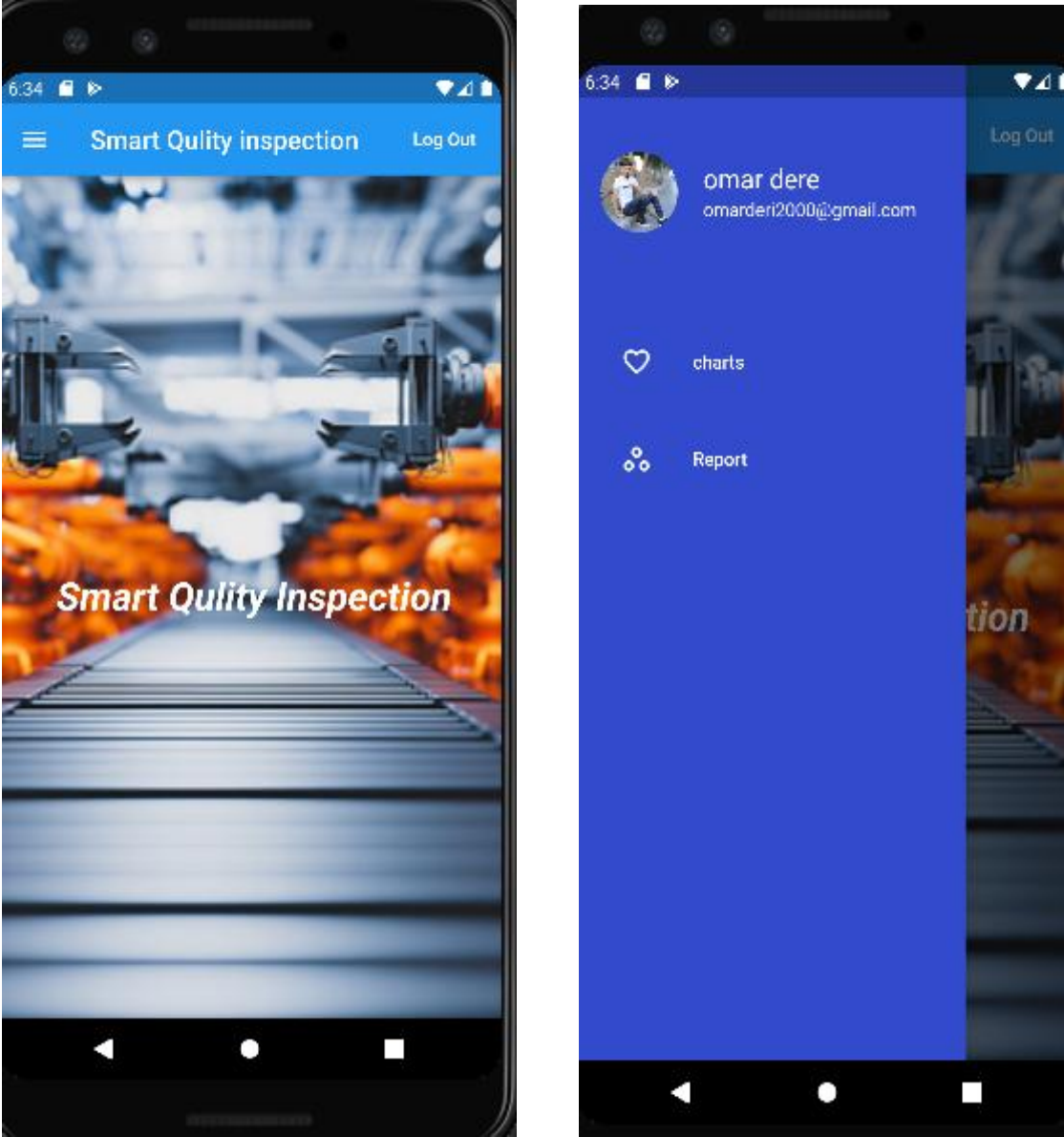


Figure 42: home page mobile and menu

Charts page contain several charts about products

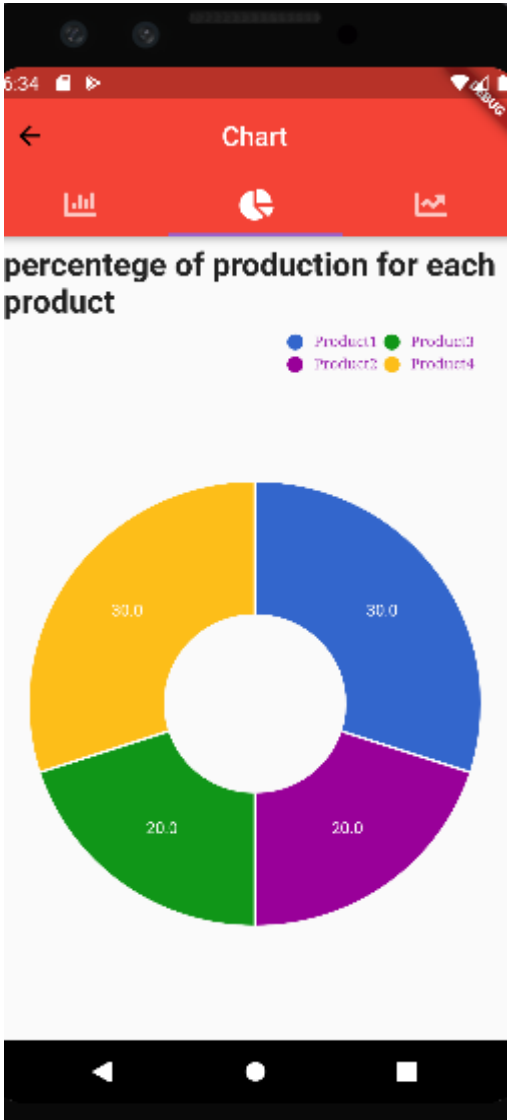
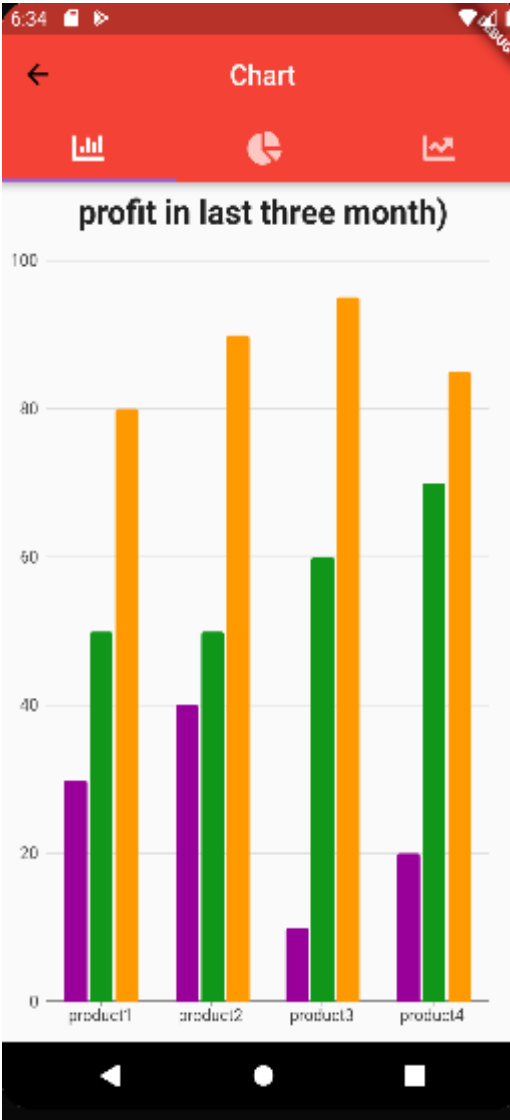


Figure 43: charts mobile

Report page contain button if pressed it display pdf page contain information about employee and company

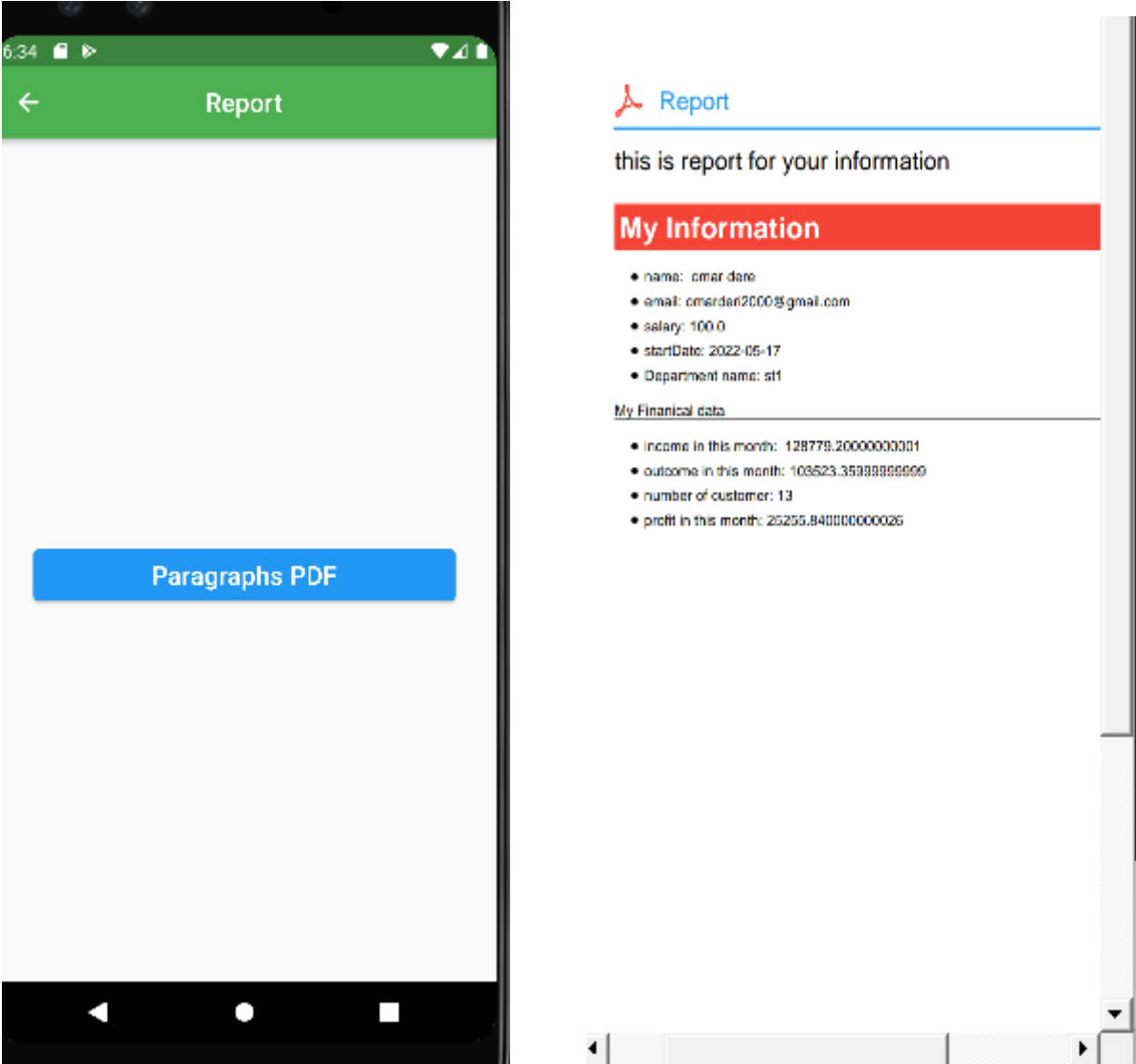


Figure 44: report page mobile

Chapter 5

Results & Discussion

6.1 Final Application

The final product is a web application that allows users to:

1. The customer can create a new account and can purchase our products.
2. The manager can manage the site and control users and employees.
3. The manager can calculate the income, outcome and profit.
4. Monitor production and stop it if defects exceed a certain limit.
5. Chat between all managers and employees.

6.2 Project Outcomes

Hopefully, this application will:

1. Increase production efficiency and give the manager more control over the assembly line.
2. Allow employees to communicate with each other via chat.
3. Save time and effort on employees.

Chapter 7

Conclusion & Future Work

7.1 Summary

Our project is an assembly line management and product detection application that aims to create a simple and easy to use software user interface.

There are many other alternatives, but most of them are limited to a specific scope such as a management system alone or product discovery alone, but our application contains both.

In addition, our project aims to enhance the communication experience between employees and managers through the use of group chat,

Our project also aims to improve the quality of production, If the percentage of defects increases, a notification will be sent to the employee, who in turn will stop production and fix the defect.

We hope that our project has fulfilled the purpose for which it was created. There was a small A set of small features that we were hoping to implement in the project which unfortunately were Dropped due to tight schedule and focus on more important features.

7.2 Future Work

There are some aspects in which our project could be improved, which include:

1. The mobile application does not contain all the features provided by the website.

These features can be implemented.

2. Using artificial intelligence to discover more defects in the product.

3. Improve the reporting system to report additional information on the project.

Bibliography

- [1] R. a. K. M. Ozdemir, "A Quality Control Application on a Smart Factory Prototype Using Deep Learning Methods," in *2019 IEEE 14th International Conference on Computer Sciences and Information Technologies (CSIT)*, -, CSIT, 2019, pp. 46-49.
- [2] N. K. a. R.-H. Y. a. J.-J. Lin, "Quality Inspection and Maintenance: The Framework of Interaction," *Procedia Manufacturing*, pp. 244-251, 2015.
- [3] react, "React," react, 13 05 2022. [Online]. Available: <https://reactjs.org/>. [Accessed 13 05 2022].
- [4] React, "React-Bootstrap," React, 05 13 2022. [Online]. Available: <https://react-bootstrap.github.io/>. [Accessed 05 13 2022].
- [5] "Data visualization with Chart.js," Chartjs, 13 5 2022. [Online]. Available: <https://tobiasahlin.com/blog/introduction-to-chartjs/#:~:text=wait%2Dwhats%2Dchartjs-,Chart.,and%20they're%20all%20responsive..> [Accessed 13 5 2022].
- [6] "Aiven for PostgreSQL," PostgreSQL, 13 05 2022. [Online]. Available: https://landing.aiven.io/en/aiven-for-postgresql/?gclid=Cj0KCQjwg_iTBhDrARIsAD3Ib5gyYPb1DIAxTu1zobUm1cwBYmCSOGKAQZeFNmmdZ28doZYX9N07bQ4aArCLEALw_wcB. [Accessed 13 05 2022].
- [7] "About opencv," Opencv, 14 05 2022. [Online]. Available: <https://opencv.org/about/>. [Accessed 14 05 2022].
- [8] "Firebase," 14 5 2022. [Online]. Available: <https://firebase.google.com/>. [Accessed 14 5 2022].
- [9] bambo agile, 17 05 2022. [Online]. Available: <https://bambooagile.eu/insights/pros-and-cons-of-using-spring-boot/?fbclid=IwAR0E4B1C808LgH21VIJr-l2FW-dtfJ-GHIAitbZmclAOd0xBwfg3x7-fANs>. [Accessed 17 05 2022].
- [1 "React Router," 14 5 2022. [Online]. Available: <https://reactrouter.com/>. [Accessed 14 5 2022].
0]
- [1 "Json Web Token," 14 5 2022. [Online]. Available: <https://reactrouter.com/>. [Accessed 14 5 2022].
1]
- [1 "bcrypt," 14 5 2022. [Online]. Available: <https://www.npmjs.com/package/bcrypt>. [Accessed 14 5
2] 2022].

[1 "axios," 14 5 2022. [Online]. Available: <https://github.com/axios/axios>. [Accessed 14 5 2022].
3]

[1
4]