# NDT(NON DESTRUCTIVE TESTING)

# Report system

**A PROJECT REPORT**

***Submitted by***


**HUDKA TULSI MANSUKHBHAI**

**191020107008**


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

**BACHELOR OF ENGINEERING**

***in***

**Computer Engineering**


**OM Engineering College, Junagadh**




**Gujarat Technological University, Ahmedabad**

**May – 2023**

**OM ENGINEERING COLLEGE,**
**Junagadh**

# CERTIFICATE

This is to certify that the project report submitted along with the project entitled **NDT(Non-destructive testing) Report system** has been carried out by **Tulsi M. Hudka** under my guidance in partial fulfillment for the degree of Bachelor of Engineering in **Computer Engineering**, 8th Semester of Gujarat Technological University, Ahmadabad during the academic year 2022-23.

Prof. M. K. Chavda
**Internal Guide**

Prof. M. K. Chavda
**Head of the Department**

ASPIRE SOFTSERV PRIVATE LIMITED
CIN : U72900GJ2017PTC099414
GST : 24AAQCA1169D1ZU

Date: 1st May 2023

## PROJECT TRAINING CERTIFICATE

**EMP Name: Tulsi Hudka**
**Enrollment No: 191020107008**

This is to certify that **Tulsi Hudka** of Faculty of Technology, **Om Engineering College, Junagadh** has worked on an Industry Defined Project of **Aspire Softserv pvt. Ltd.** . The work embodied in this project entitled, "**Non Destructive Testing Report System**" has been carried out in fulfilment for the degree of Bachelor of Engineering. She has undergone the project for the required period **23/01/2023 to till now.**
During this period, we found her sincere, honest and diligent. We wish all success in her future endeavors.

**For Aspire Software Solutions,**

ASPIRE SOFTWARE SOLUTIONS Authorised

**Riddhi Rajwansh**
**HR Executive**

Kevin Shah

**Kevin Shah**
**Mentor**

+91-79-4602-5309 hello@aspiresoftserv.com www.aspiresoftserv.com
402, Vishwa Complex, Opp. Jain Temple, Shrimali Soc., Navrangpura, Ahmedabad, Gujarat 380009

**OM ENGINEERING COLLEGE,**
**Junagadh**

# DECLARATION

We hereby declare that the Internship report submitted along with the Internship entitled **NDT(Non-destructive testing) Report system** submitted in partial fulfillment for the degree of Bachelor of Engineering in **Computer Engineering** to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me at Aspire Softserv Pvt. Ltd. under the supervision of Mr. Chirag Sen and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

| Name of the Student | Sign of Student |
| --- | --- |
| Tulsi M. Hudka | _____ |

# ACKNOWLEDGMENT

It is truly a pleasure to express my thanks and gratitude to everyone who has helped me throughout the process of making this project. This project gave me a great opportunity to think, implement and interact with different aspects of the software development lifecycle. I would like to thank everyone who has helped me at one time or another by providing me with the support, encouragement, and foundation needed to carry out my project throughout my internship.

I express my deep gratitude to our External Project Faculty, **Mr. Kevin C. Shah** and Internal Project Guide, **Prof. M. K. Chavda** for their innovative ideas and efforts. Their seriousness to make my project successful. It was their sincerity that inspired me throughout the project to work hard using technologies adopted by the industry.

Also, I am sincerely grateful to **Prof. M. K. Chavda** for his unconditional and objective support throughout the research and development process. All of which provide me with a supportive environment. Without them, I would not have achieved my goal. Also, I thank the manager of **Aspire Softserv Pvt. Ltd.** for giving us the opportunity to work at.

The combination of gratitude, joy and great satisfaction is what I feel like passing my debt on to all those who have contributed directly or indirectly to the implementation of the project.

With Sincere Regards,

Tulsi M. Hudka

# ABSTRACT

*In growing country, India, employers go through many difficulties especially those who are a scarce of employees and find it hard to manage the company records. They often get perplexed with the amount of data they have to manage and record in different files. The project I worked on during this internship at Aspire Softserv Pvt. Ltd. was to solve the above faced issue. The company works of creating software for its clients.*

*The company is currently working on a inspection related application for various companies. Furthermore, I was assigned to work alongside to make a web application for the same. We learnt React and Node during the period of internship and eventually implemented the acquired knowledge in creating a web application. We will be taught how to host the application on web server later on.*

*The Web Application is used for storing all the data or status of product testing . Additionally, the data about the transaction of payment and reception is all a feature included in the web application. Various components learnt during the internship are also implemented in the project like Crowd funding portal , Recruiter based web app, etc.*

# LIST OF FIGURES

# LIST OF TABLES

# TABLE OF CONTENTS

# ABBREVIATIONS

| | |
|---|---|
| NDT | Non-Destructive Testing |
| DOM | Document Object Model |
| API | Application Programming Interface |
| I/O | Input/Output |
| DFD | Data Flow Diagram |
| HTML | Hyper Text Markup Language |
| CSS | Cascading Style Sheet |
| AWS | Amazon Web Services |
| CI/CD | Continuous Integration/Continuous Deployment |
| HTTPS | Hyper Text Transfer Protocol |
| OS | Operating System |

# Chapter # 1: OVERVIEW OF THE COMPANY

1. **HISTORY**
2. **DIFFERENT PRODUCTS**
3. **ORGANIZATION CHART**

# 1. Overview of The Company

## *1.1. History*

Aspire is a leading Java EE, Liferay, Mobility BigData and Customized Software Development Services providing company with development center in Ahmedabad, Gujarat, India. We have been providing high-quality, high-value software development services to the Independent Software Vendors and the enterprises since 2010 by leveraging best-in-class people, processes and technologies.

Aspire is dedicated to developing(14+years of experience) effective outsourcing partnerships with clients in order to accelerate time to market, reduce operational cost, and empower them to devote more time to their core business.

Since our founding, Aspire has built a successful business by focusing on a simple yet compelling mission: clients' success is our success. We do so through careful, strategic investments in technology, process, and—most significantly—people. Indeed, Aspire's people are our greatest strength for high-quality, custom software development and business process outsourcing support.

## *1.2.Different Products*

Our passion for quality and expertise in software product development result in finely tuned, reliable software solutions. Our capabilities and expertise are comprehensive—covering every phase of the software development product lifecycle. Our support is unparalleled—with a dedicated team of project management, engineering, and testing experts assigned to you and your product. And our approach is uniquely effective—with a suite of proven processes and tools to accelerate the development of high-quality software.

## *1.3.Organization Chart*

# 2

## Chapter # 2: INTRODUCTION TO PROJECT

1. **PROJECT SUMMARY**
2. **PROJECT PURPOSE**
3. **PROJECT SCOP & OUTCOMES**
4. **GOALS & OBJECTIVES**
5. **KEY FEATURES**
6. **TECHNOLOGY & LITERATURE OVERVIEW**

## 2. Introduction to Project

### *2.1. Project Summary*

Non-destructive testing (NDT) is a wide group of analysis techniques used in science and technology industry to evaluate the properties of a material, component or system without causing damage. It is the process of inspecting, testing, or evaluating materials, components or assemblies for discontinuities, or differences in characteristics without destroying the serviceability of the part or system.

The application is a platform for the customer to automate the NDT(non-destructive testing) reports Along with all back-office work related to it. It is general platform for all the NDT testing companies, from where companies can manage customer/technician details, perform auto job assignment, get Real time job progress, perform auto test result reports, manage invoices, inventory and wastage of Raw materials.

### *2.2. Project Purpose*

The purpose of the Non-Destructive Testing (NDT) Report System is to provide an efficient and accurate method for reporting the results of NDT inspections. The system is designed to improve the quality of NDT reports by standardizing the reporting process and reducing the potential for human error.

The system aims to address the limitations and challenges of current NDT report systems, such as inconsistencies in reporting, time-consuming manual data entry, and the potential for errors in data transcription. The proposed NDT report system will leverage technology to automate data collection and analysis, and to provide a standardized reporting format.

Overall, the purpose of the NDT report system is to improve the accuracy and efficiency of NDT reporting, while also ensuring compliance with industry standards and regulations. This will ultimately lead to increased safety and reliability in industries that rely on NDT for quality assurance and control.

### *2.3. Project Scope & Outcomes*

The scope of the Non-Destructive Testing (NDT) Report System includes the following:

Data Collection: The system will automate the process of collecting data from NDT inspections using various sensors, such as ultrasonic or radiographic sensors. The collected data will be stored in a database and processed for analysis.

Reporting: The system will generate a standardized report that presents the results of the NDT inspection in a clear and concise format. The report will include detailed information about the inspection process, the equipment used, the data collected, and the results of the analysis. The report will be customizable to meet specific industry standards and requirements.

Integration: The system will be designed to integrate with existing NDT inspection equipment and software. This will allow for seamless data transfer and integration with other systems.

The expected outcome of the NDT Report System is to improve the accuracy and efficiency of NDT inspections and reporting, reduce the potential for human error, and increase safety and reliability in industries that rely on NDT for quality assurance and control. The system will also improve compliance with industry standards and regulations, and provide a valuable tool for quality control and risk management.

### *2.4. Goals & Objectives*

- To manage customer/technician details, perform auto job assignment, get real time job progress, perform test result reports, manage invoices, inventory, and wastage of raw materials.

### *2.5. Key Features*

Standardized Reporting: The system generates a standardized report that presents the results of the NDT inspection in a clear and concise format. The report includes detailed information about the inspection process, the equipment used, the data collected, and the results of the analysis. The report is customizable to meet specific industry standards and requirements.

Real-time Monitoring: The system provides real-time monitoring of the inspection process, allowing for immediate feedback on the quality and accuracy of the data being collected. This helps to ensure that the inspection is completed correctly and efficiently.

Integration with Existing Systems: The system is designed to integrate with existing NDT inspection equipment and software. This allows for seamless data transfer and integration with other systems, such as asset management or quality control systems.

User-friendly Interface: The system offers a user-friendly interface that is easy to navigate and use. The interface is designed to be intuitive and customizable to meet the needs of different users.

Overall, these key features make the NDT Report System a powerful tool for enhancing the accuracy and efficiency of NDT reporting, while also improving compliance with industry standards and regulations.

### *2.6.Technology and Literature Overview*

***Technology Overview*:**

The Non-Destructive Testing (NDT) Report System can be developed using various technologies, depending on the specific requirements and needs of the project. Some of the technologies that can be used for developing the system are:

MongoDB: This is a NoSQL database that is commonly used for building scalable and high-performance applications. It can store large amounts of data and offers high availability and fault tolerance.

Express.js: This is a popular Node.js framework that is used for building web applications and APIs. It offers several features such as routing, middleware, and templating.

React: This is a popular JavaScript library that is used for building user interfaces for web applications. It offers several features such as component-based architecture, virtual DOM, and reusable UI components.

Node.js: This is a JavaScript runtime that is used for building server-side applications. It offers several features such as event-driven architecture, non-blocking I/O, and scalability.

***Literature Overview:***

There is a significant amount of literature available on Non-Destructive Testing (NDT) and related technologies. Some of the key topics that can be explored for the NDT Report System are:

NDT inspection techniques and equipment: This includes various techniques and equipment used for NDT inspections, such as ultrasonic testing, radiographic testing, eddy current testing, and magnetic particle testing.

Web development frameworks and technologies: This includes research on various web development frameworks and technologies, such as MongoDB, Express.js, React, and Node.js, and how they can be used for developing web applications.

Case studies and examples of NDT report systems: This includes research on existing NDT report systems, their features, and how they are being used in various industries, such as aerospace, automotive, and construction.

**Chapter # 3: PROJECT MANAGEMENT**

1. **PROJECT PLANNING**
2. **PROJECT SCHEDULING**
3. **ESTIMATION**
4. **RISK IDENTIFICATION, ANALYSIS AND MANAGEMENT**

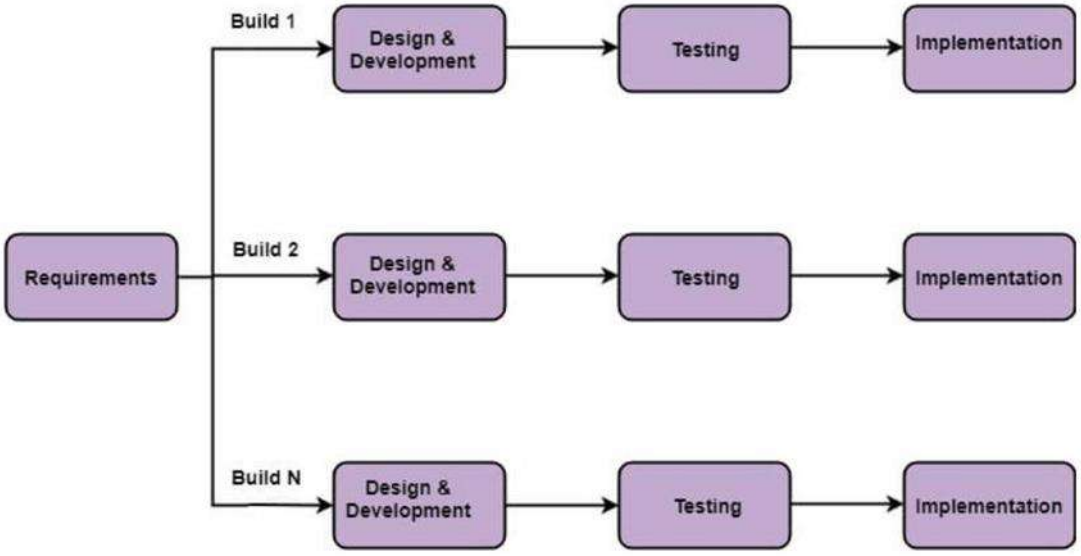
## *3. Project Management*

### *3.1. Project Planning*

We are using Incremental Methodology in development process**.**

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation, and testing phases. Every subsequent release of the module adds function to the previous release

Fig 3.1 Incremental Methodology (“Courtesy of javapoint.com”)




The various phases of incremental model are as follows:

1. Requirement analysis: In the first phase of the incremental model, the product analysis expertise identifies the requirements. And the system functional requirements are understood by the requirement analysis team.
2. Design & Development: In this phase of the Incremental model of SDLC, the design of the system functionality and the development method are finished with success.
3. Testing: In the incremental model, the testing phase checks the performance of each existing function as well as additional functionality.
4. Implementation: Implementation phase enables the coding phase of the development system. It involves the final coding that design in the designing and development phase and tests the functionality in the testing phase

### *3.2. Project Scheduling*

The project scheduling phase for the NDT Report System involves creating a detailed project plan that includes tasks, dependencies, and timelines.

Table 3.1 Project Schedule

| | February | March | April |
|---|---|---|---|
| Start Of project | | | |
| Requirement Gathering | | | |
| Planning | | | |
| System design | | | |
| Coding | | | |
| Testing | | | |

### *3.3. Estimation*

The estimation phase for the NDT Report System involves estimating the time and effort required to complete the project. The key steps involved in this phase are:

Define project scope: This involves clearly defining the project scope, including the features and functionality that need to be developed.

Determine project size: This involves estimating the size of the project, such as the number of lines of code or the number of user interface screens.

Estimate task effort: This involves estimating the time and effort required to complete each task, based on historical data or expert judgment.

Create a project budget: This involves using the estimated task effort to create a budget for the project, including the costs of hardware, software, development, and support.

### *3.4. Risk Identification, Analysis and Management*

Identify project risks: This involves identifying potential risks that could impact the project, such as changes in requirements, technology failures, or team member turnover.

Analyze risk impact and likelihood: This involves analyzing the impact and likelihood of each identified risk, based on factors such as the severity of the risk and the likelihood of it occurring.

Develop risk response strategies: This involves developing strategies to mitigate or manage each identified risk, such as contingency planning or risk transfer.

Implement risk response strategies: This involves implementing the risk response strategies and monitoring the effectiveness of these strategies.

Review and update risk management plan: This involves regularly reviewi

# 4

**Chapter # 4: SYSTEM REQUIREMENT STUDY**

1. **USER CHARACTERSTICS**
2. **FUNCTION/ NON- FUNCTIONAL REQUIREMENTS**
3. **SOFTWARE & HARDWARE REQIUREMENTS**

# 4. System Requirement Study

## *4.1.User Characteristics*

4.1.1. *Admin*: The admin is responsible for managing the overall system, ensuring its smooth functioning, and handling any issues that arise. The characteristics of an admin include:

- Strong technical skills and knowledge of the system's architecture and components
- Good communication and leadership skills to manage the team and coordinate with other stakeholders
- Ability to handle complex technical issues and make critical decisions
- Excellent problem-solving and analytical skills to identify and address system-related problems

4.1.2. *Manager*: The manager is responsible for overseeing the NDT operations, including scheduling inspections, assigning tasks to technicians, and reviewing inspection reports. The characteristics of a manager include:

- Good organizational skills and attention to detail to manage multiple inspections and tasks effectively
- Strong communication skills to communicate with clients and technicians
- Knowledge of NDT techniques and their applications to ensure high-quality inspections
- Ability to handle time-sensitive situations and work under pressure

4.1.3. *Technician*: The technician is responsible for conducting NDT inspections and collecting data, which is then used to generate inspection reports. The characteristics of a technician include:

- Strong technical skills and knowledge of NDT techniques and equipment
- Attention to detail and accuracy in collecting data to ensure reliable inspection results
- Good communication skills to explain inspection results to clients and managers
- Ability to work independently and make sound decisions during inspections

*4.1.4* *Client*: The client is the end-user of the NDT Report System and relies on the system to access and review inspection reports. The characteristics of a client include:

- Limited technical knowledge of NDT techniques and the system's operation
- A strong focus on the accuracy and reliability of inspection results
- Ease of use and accessibility to the system to access and download reports

- Good communication skills to provide feedback and report issues to the system administrators.

### *4.2.Functional / Non-Functional Requirements*

#### *4.2.1. Functional Requirements:*

- User Management: The system must provide user authentication and authorization features, allowing admins to manage user accounts and access levels.
- Inspection Scheduling: The system must allow managers to schedule inspections, assign technicians, and track inspection progress.
- Inspection Data Collection: The system must allow technicians to collect inspection data and upload it to the system.
- Report Generation: The system must generate comprehensive and accurate reports based on the collected data, adhering to industry standards.
- Report Review and Approval: The system must allow managers to review and approve reports before they are made available to clients.
- Report Access: The system must provide clients with secure access to their inspection reports.

#### *4.2.2. Non-Functional Requirements:*

- Performance: The system must be able to handle large volumes of inspection data and generate reports efficiently.
- Security: The system must comply with industry-standard security protocols to protect data integrity and privacy.
- Usability: The system must be user-friendly and intuitive, with clear navigation and user interfaces.
- Scalability: The system must be able to scale up or down based on changing inspection volumes and user traffic.
- Availability: The system must be available 24/7 to allow users to access the reports at any time.
- downtime and disruption to user access.

### *4.3.Software and Hardware Requirements*

#### ***4.3.1.*** *Software requirements*

- MongoDB
- ExpressJS
- NodeJS
- ReactJS
- Visual Studio Code

#### *4.3.2. Hardware requirements*

- Windows
- Core i5 CPU
- 8gb RAM
- 50 GB storage space

**Chapter # 5: SYSTEM ANALYSIS**

# 5. System Analysis

## *5.1. Study of Current System*

To develop the NDT Report System, a study of the current NDT inspection and reporting process was conducted. The study aimed to identify the existing challenges and limitations of the current system and propose solutions that could be addressed by the new system.

The study involved several activities, including:

- *Gathering requirements:* The requirements for the new system were gathered by conducting interviews and surveys with stakeholders, including NDT technicians, managers, and clients. The gathered requirements were then documented in a detailed requirements specification document.
- *Reviewing current processes:* The current NDT inspection and reporting process were reviewed to identify inefficiencies, redundancies, and other limitations that could be addressed by the new system.
- *Analysing data flow:* The data flow within the current system was analysed to identify potential bottlenecks and areas for improvement. The analysis included the inspection scheduling process, data collection process, report generation, and report distribution.
- *Identifying technical limitations:* The study also identified the technical limitations of the current system, including outdated hardware and software, incompatible data formats, and limited data storage capacity.
- *Analysing industry standards:* The study analysed industry standards for NDT inspections and reporting to ensure that the new system adheres to the latest best practices and regulations.

Based on the study's findings, the new NDT Report System was designed to address the identified limitations and challenges. The system's key features, functional and non-functional requirements, and user characteristics were developed based on the study's results.

## *5.2. Problem in Current System*

The following Problems exist in current system,

- User is currently using desktop-based application which available only from his computer.
- There are several drawbacks of the existing system
- Not available from every machine

- No real time job updates.
- Support only Radiology testing reports.
- Not able to find wastage of materials
- Dashboard is not available to know overall idea of business using graphical
- Auto invoice reminders are not available.

### *5.3.Requirement of New System*

- Conceptualization of the application to monetize application with minimal investment.
- Real-time data updates for end user.
- Module based development to install it on subscription bases.
- Microservice based project architecture.
- Dashboard design to meet expectation of every company in same domain.
- Automate the job workflow to minimize manual task handling.

### *5.4.Feasibility Study*

#### *5.4.1 Technical Study*

The technical feasibility assessment is focused on gaining an understanding Of the present technical resources of the organization and their applicability to The expected needs of the proposed system. It is an evaluation of the hardware And software and how it meets the need of the proposed system.

To become aware of any potential problems that could occur while implementing the project. After considering all significant factors, the project Is viable that is, worth undertaking. To develop this web application all Resources are available Like server, hardware, OS etc. so our system is Technical feasible.

#### *5.4.2Economical Study*

Economic feasibility is where analysis of a project's costs and revenues in An effort to determine whether or not it is logical and possible to complete it Is called economical study.

Our system is one type of source; user cannot pay any extra fees and it is Open source for all people. They cannot purchase any additional hardware so, Our system is economical feasible.

### *5.4.3Operational Study*

Operational feasibility determine if the human resource is available to Operate the system once it has been installed. Users that do not want a new System may prevent it from becoming operationally feasible. It measures of How a proposed system solved the problems, and takes advantages of the Opportunities identified during scope definition and how it satisfied the Requirement identified in the requirement analysis phase of system Development.

User can fulfil all functionality of system, after develop system it is very user friendly so client are very familiar with our website so, our system is operational feasible.

### *5.5.Functions of System – Use Case Diagram*

Fig 5.1 Use case (Admin)

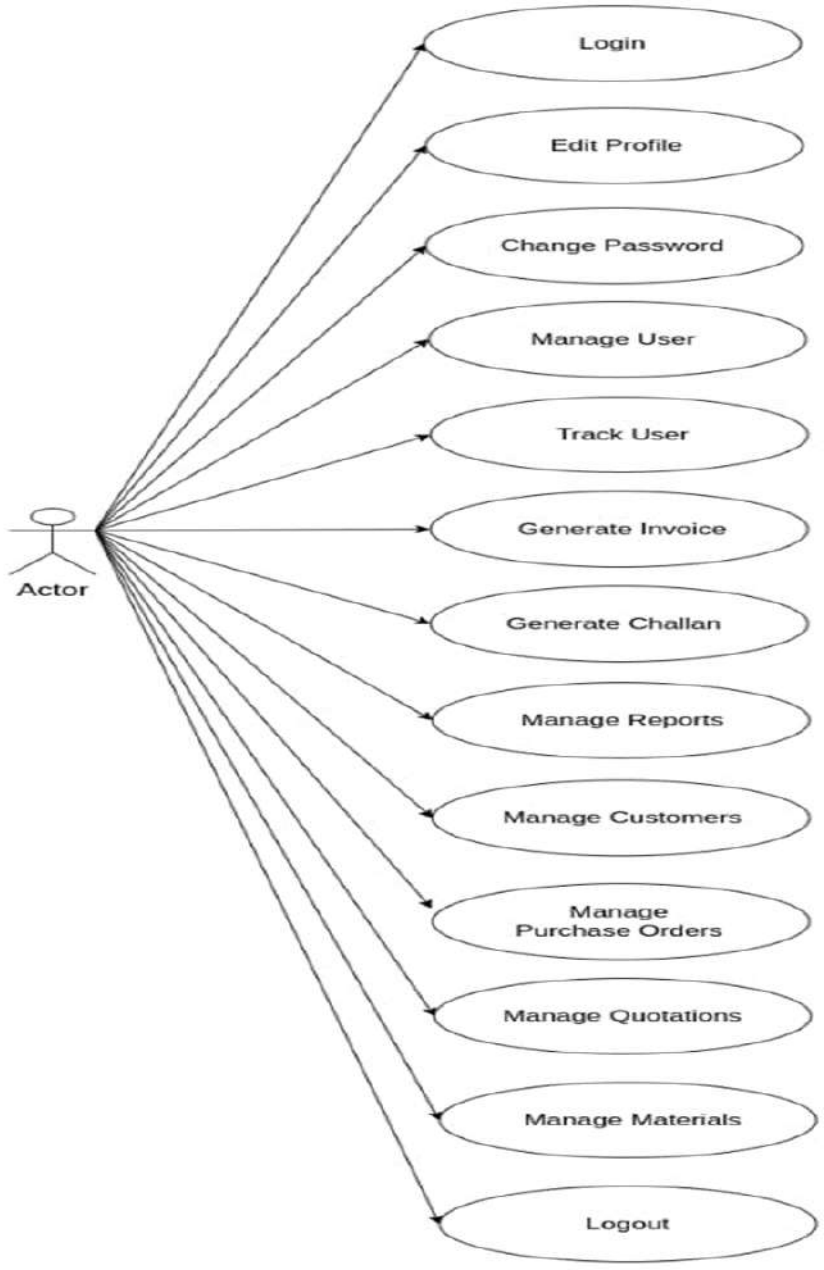


Fig 5.2 Use case (Customer)

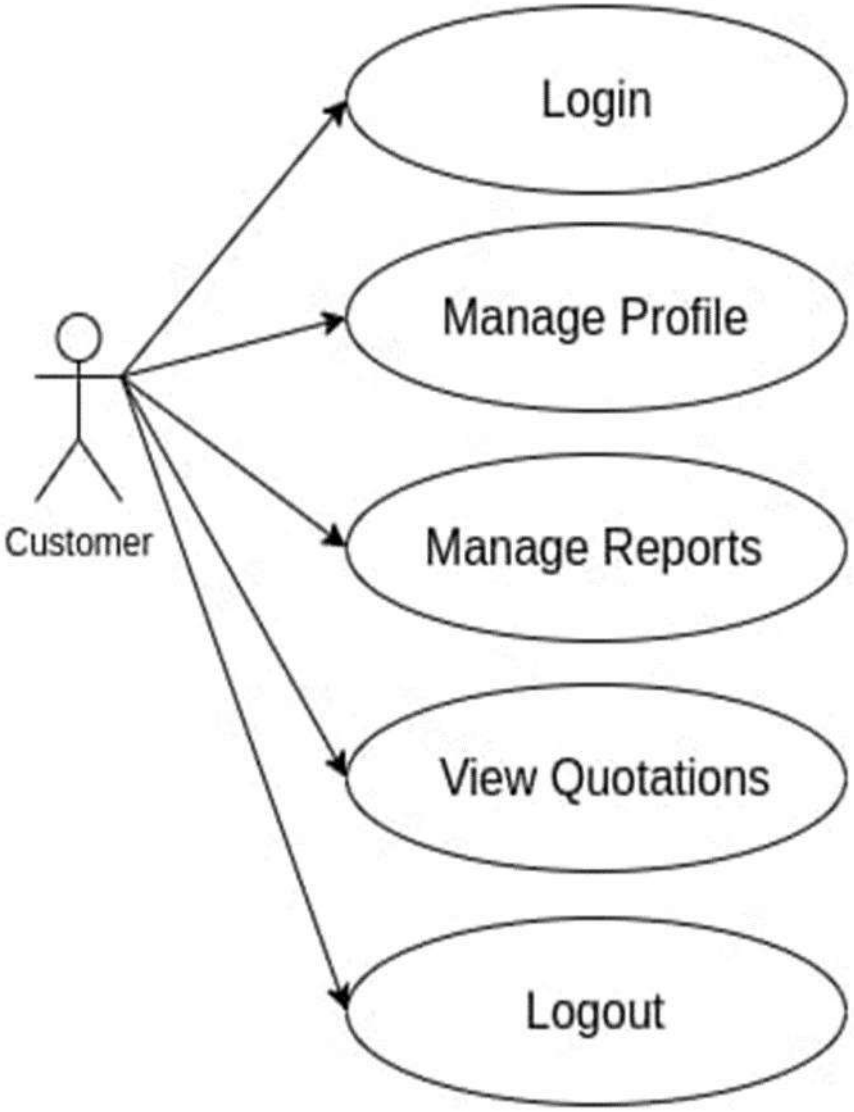


Fig .3 Use case (Technician)

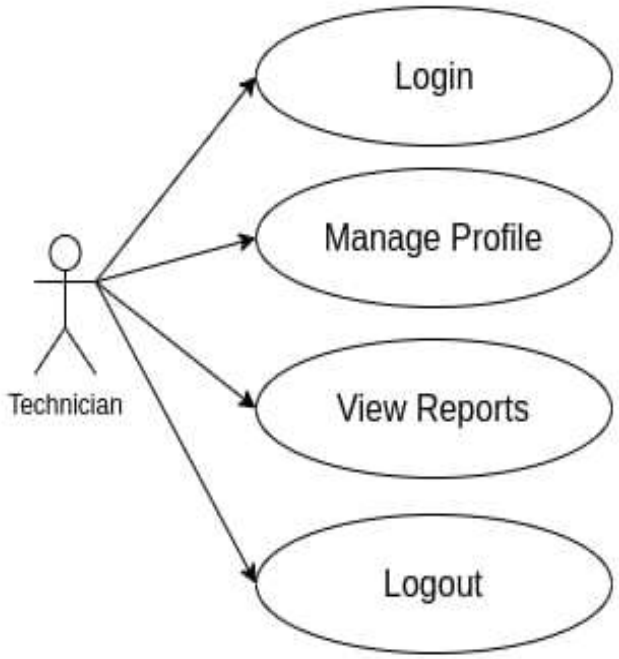
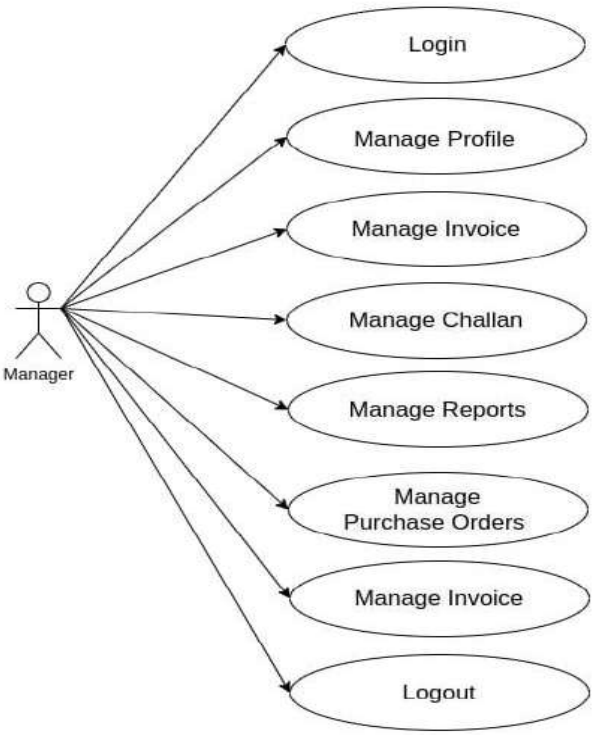



Fig 5.4 Use case (Manager)



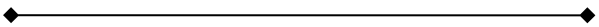
### *5.6.Features of New System*

The new NDT Report System will have the following features:

- *User Management:* The system will provide user authentication and authorization features, allowing admins to manage user accounts and access levels.
- *Inspection Scheduling:* The system will allow managers to schedule inspections, assign technicians, and track inspection progress.
- *Inspection Data Collection:* The system will allow technicians to collect inspection data and upload it to the system using a user-friendly interface.
- *Report Generation:* The system will generate comprehensive and accurate reports based on the collected data, adhering to industry standards.
- *Report Review and Approval:* The system will allow managers to review and approve reports before they are made available to clients.
- *Report Access:* The system will provide clients with secure access to their inspection reports, with the ability to search and filter the reports based on various criteria.
- *Dashboard:* The system will provide a customizable dashboard for each user role, showing relevant information and metrics.
- *Notifications:* The system will provide automatic notifications to users for new inspection schedules, report approvals, and other important events.
- *Data Analytics:* The system will provide data analytics and visualization features, allowing users to analyze trends and patterns in the inspection data.
- *Integration:* The system will allow integration with other software and tools, such as data storage and analysis tools, to provide a more comprehensive solution for NDT inspection and reporting.

These features will help improve the efficiency, accuracy, and accessibility of the NDT inspection and reporting process, providing a more streamlined and user-friendly experience for all users.

**Chapter # 6: SYSTEM DESIGN**



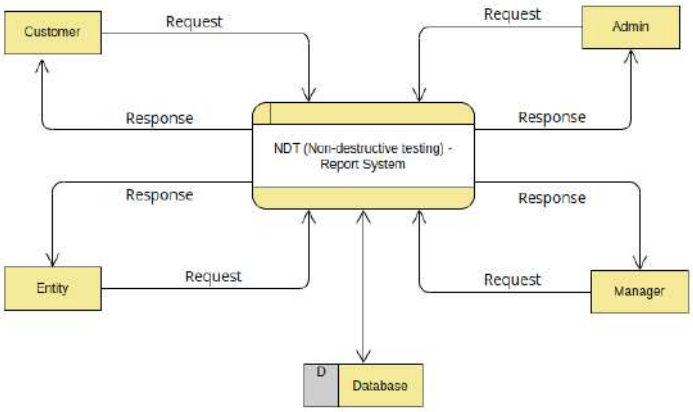
**6.1 FUNCTIONAL AND BEHAVIORAL MODELLING-DFD**
**6.2 ERD**
**6.3 DATA DICTIONARY**

# 6. Project Management

## 6.1. Functional and Behavioral Modeling – DFD

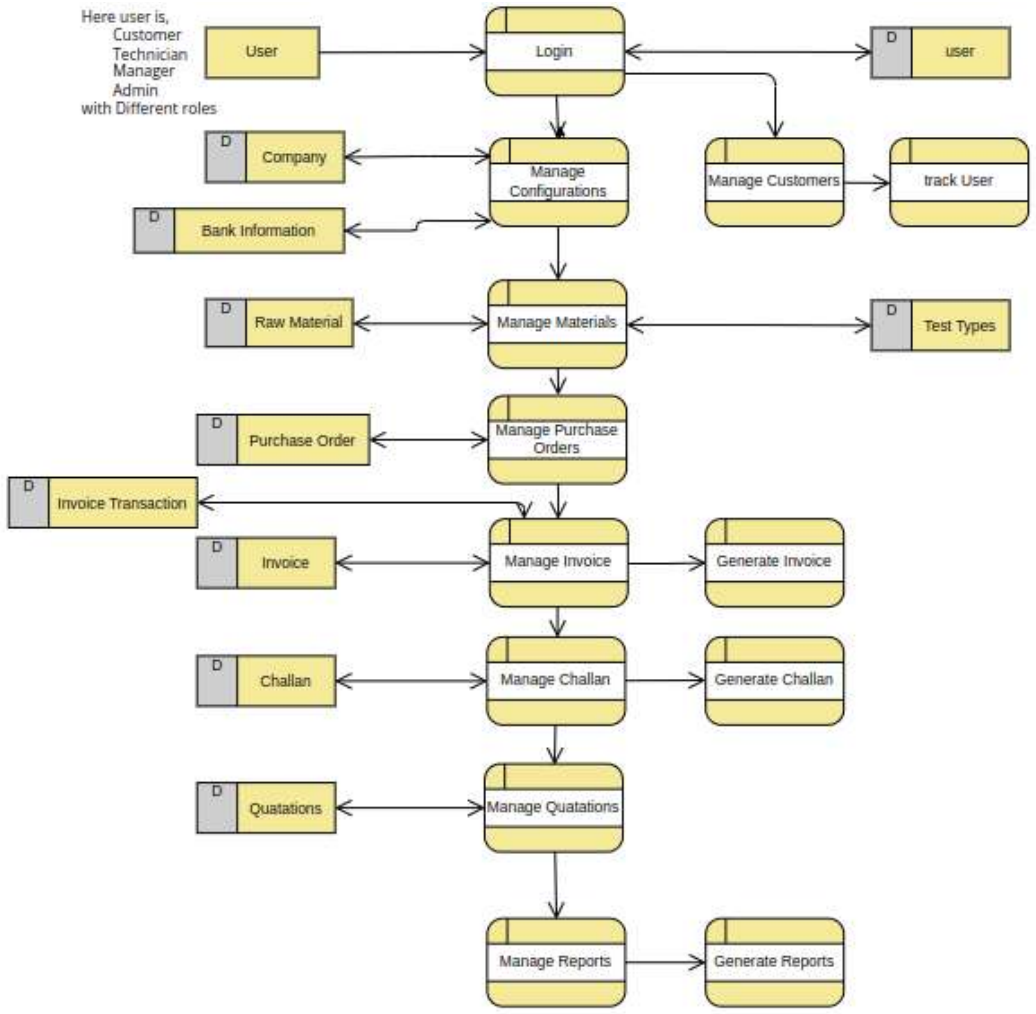
### 6.1.1. Context Level DFD

Fig 6.1 context level DFD



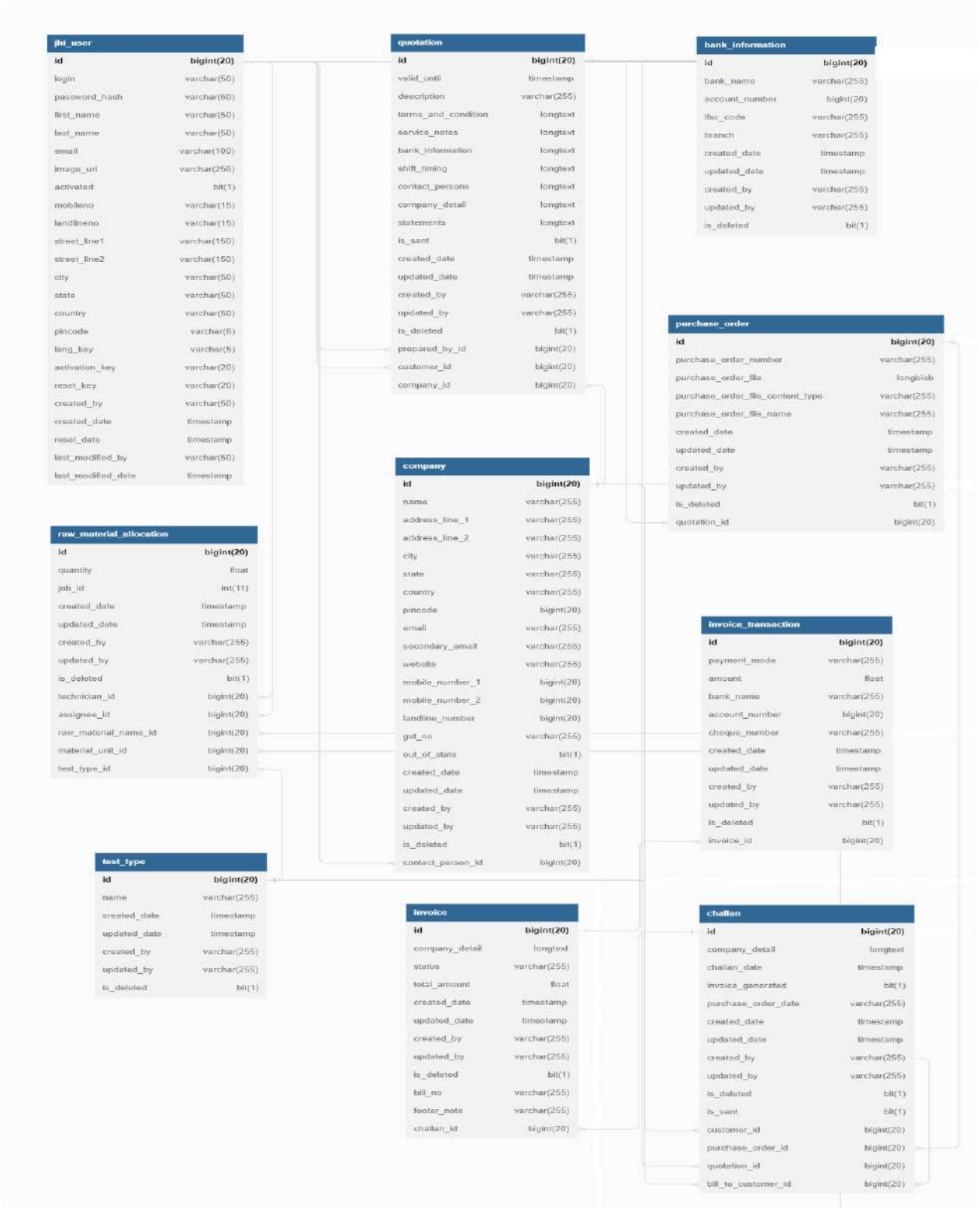

### 6.1.2. First Level DFD

Fig 6.2 First level DFD



### *6.2.ERD*

Fig 6.3 ERD



### *6.3. Data Dictionary*

Table 6.1 DD User Table 1

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| Id | bigint | Primary Key | Unique id |
| login | varchar(50) | Foreign Key | Username or credinitials |
| password_hash | varchar(60) | NOT NULL | Password |
| first_name | varchar(50) | NOT NULL | First name of user |
| last_name | varchar(50) | NOT NULL | Last name of user |
| email | varchar(100) | Foreign Key | Email of user |
| image_url | varchar(256) | NOT NULL | User profiie image |
| activated | bit(1) | NOT NULL | User is active or not |
| mobileno | varchar(15) | NOT NULL | User mobile no |
| street_line1 | varchar(150) | NOT NULL | User flat or society name |
| street_line2 | varchar(150) | NOT NULL | User area name |
| city | varchar(50) | NOT NULL | City name |
| state | varchar(50) | NOT NULL | State name |

Table 6.2 DD Bank Table 2

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| Id | bigint | Primary Key | Unique Id |
| bank_name | varchar(255) | NOT NULL | User Bank Name |
| account_number | bigint | NOT NULL | User Bank Account |
| ifsc_code | varchar(255) | NOT NULL | User Bank IFSC Code |
| branch | varchar(255) | NOT NULL | Bank Branch Name |

Table 6.3 DD Invoice Table 3

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| id | bigint | Primary Key | |
| company_detail | longtext | NOT NULL | |
| status | varchar(255) | NOT NULL | |
| total_amount | float | NOT NULL | |
| created_date | timestamp | NOT NULL | |
| updated_date | timestamp | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |
| bill_no | varchar(255) | NOT NULL | |
| footer_note | varchar(255) | NOT NULL | |
| challan_id | bigint | Foreign Key | |

Table 6.4 DD Invoice Table 4

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| id | bigint | Primary Key | |
| payment_mode | varchar(255) | NOT NULL | |
| amount | float | NOT NULL | |
| bank_name | varchar(255) | NOT NULL | |
| account_number | bigint | NOT NULL | |
| cheque_number | varchar(255) | NOT NULL | |
| created_date | timestamp | NOT NULL | |
| updated_date | timestamp | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |
| invoice_id | bigint | Foreign Key | |

Table 6.5 DD Purchase order Table 5

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| Id | bigint | Primary Key | |
| purchase_order_number | varchar(255) | NOT NULL | |
| purchase_order_file | longblob | NOT NULL | |
| purchase_order_file_content_type | varchar(255) | NOT NULL | |
| purchase_order_file_name | varchar(255) | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |
| quotation_id | bigint | Foreign Key | |

Table 6.6 DD Challan Table 6

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| Id | bigint | Primary Key | |
| company_detail | longtext | NOT NULL | |
| challan_date | timestamp | NOT NULL | |
| invoice_generated | bit(1) | NOT NULL | |
| purchase_order_date | varchar(255) | NOT NULL | |
| created_date | timestamp | NOT NULL | |
| updated_date | timestamp | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |
| is_sent | bit(1) | NOT NULL | |
| customer_id | bigint | Foreign Key | |
| purchase_order_id | bigint | Foreign Key | |
| quotation_id | bigint | Foreign Key | |
| bill_to_customer_id | bigint | Foreign Key | |

Table 6.7 DD Company Table 7

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| id | bigint | Primary Key | |
| name | varchar(255) | NOT NULL | |
| address_line_1 | varchar(255) | NOT NULL | |
| address_line_2 | varchar(255) | NOT NULL | |
| city | varchar(255) | NOT NULL | |
| state | varchar(255) | NOT NULL | |
| country | varchar(255) | NOT NULL | |
| pincode | bigint | NOT NULL | |
| email | varchar(255) | NOT NULL | |
| secondary_email | varchar(255) | NOT NULL | |
| website | varchar(255) | NOT NULL | |
| mobile_number_1 | bigint | NOT NULL | |
| mobile_number_2 | bigint | NOT NULL | |
| landline_number | bigint | NOT NULL | |
| gst_no | varchar(255) | NOT NULL | |
| out_of_state | bit(1) | NOT NULL | |
| created_date | timestamp | NOT NULL | |
| updated_date | timestamp | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |
| contact_person_id | bigint | Foreign Key | |

Table 6.8 DD Quotation Table 8

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| Id | bigint | Primary Key | |
| valid_until | timestamp | NOT NULL | |
| description | varchar(255) | NOT NULL | |
| terms_and_condition | longtext | NOT NULL | |
| service_notes | longtext | NOT NULL | |
| bank_information | longtext | NOT NULL | |
| shift_timing | longtext | NOT NULL | |
| contact_persons | longtext | NOT NULL | |
| company_detail | longtext | NOT NULL | |
| statements | longtext | NOT NULL | |
| is_sent | bit(1) | NOT NULL | |
| created_date | timestamp | NOT NULL | |
| updated_date | timestamp | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |
| prepared_by_id | bigint | NOT NULL | |
| customer_id | bigint | NOT NULL | |
| company_id | bigint | Foreign Key | |

Table 6.9 DD Raw material allocation Table 9

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| Id | bigint | Primary Key | |
| quantity | float | NOT NULL | |
| job_id | int | NOT NULL | |
| created_date | timestamp | NOT NULL | |

| | | | |
|---|---|---|---|
| updated_date | timestamp | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |
| technician_id | bigint | Foreign Key | |
| assignee_id | bigint | Foreign Key | |
| raw_material_name_id | bigint | Foreign Key | |
| material_unit_id | bigint | Foreign Key | |
| test_type_id | Bigint | Foreign Key | |

Table 6.10 DD Test type Table 10

| Column | Data Type | Constraints | Description |
|---|---|---|---|
| id | bigint | Primary Key | |
| name | varchar(255) | NOT NULL | |
| created_date | timestamp | NOT NULL | |
| updated_date | timestamp | NOT NULL | |
| is_deleted | bit(1) | NOT NULL | |

# 7

**Chapter # 7: IMPLEMENTATION ENVIRONMENT**

# 7. *Implementation Environment*

## *7.1. Implementation Environment*

Server-side Technologies: The server-side of the system is built using Node.js, Express.js, and MongoDB. Node.js is used as the server-side platform, Express.js is used as the web application framework, and MongoDB is used as the database management system.

Client-side Technologies: The client-side of the system is built using React, Bootstrap, and HTML/CSS. React is used as the front-end framework, Bootstrap is used as the UI design framework, and HTML/CSS are used to structure and style the web pages.

Cloud Hosting: The system is hosted on a cloud-based server, such as Amazon Web Services (AWS), to provide scalability, reliability, and security.

Continuous Integration/Continuous Deployment (CI/CD) Tools: The system is developed and deployed using CI/CD tools, such as Jenkins and Git, to automate the software development lifecycle and ensure the quality of the code.

Issue Tracking and Project Management Tools: The system development and management process are tracked using issue tracking and project management tools, such as Jira, to ensure timely completion of tasks and efficient collaboration among team members.

## 7.2. *Screen Shots*

Fig 7.1 Home Page

- User can login here

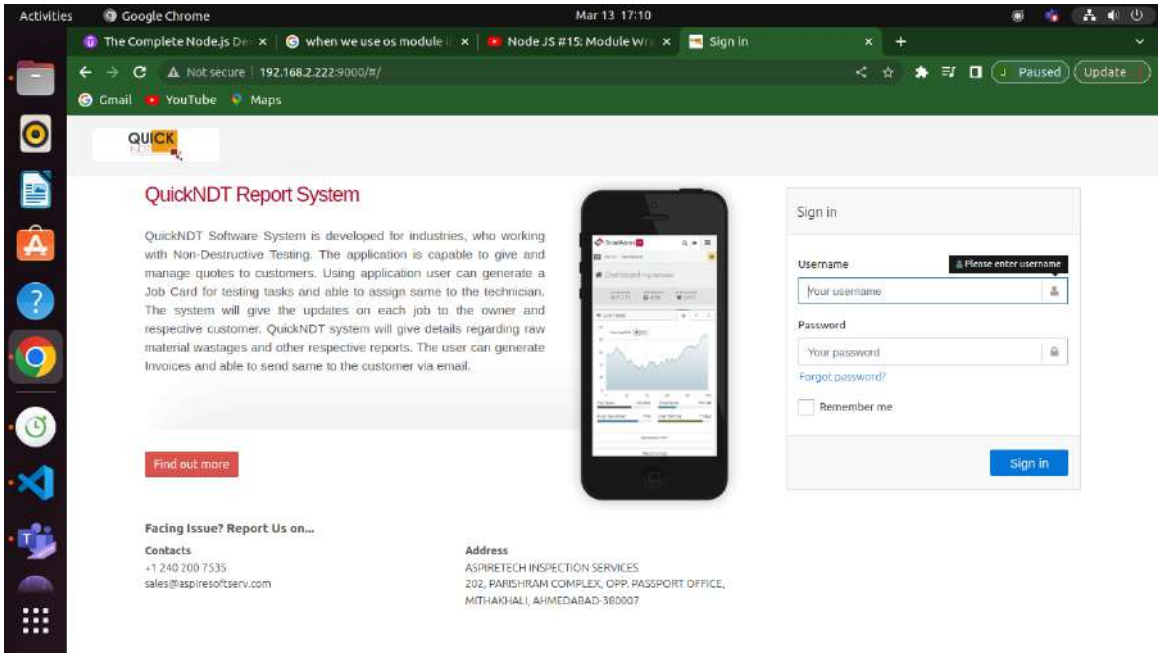


Fig 7.2 Forgot password

- User enters email here for forget their password

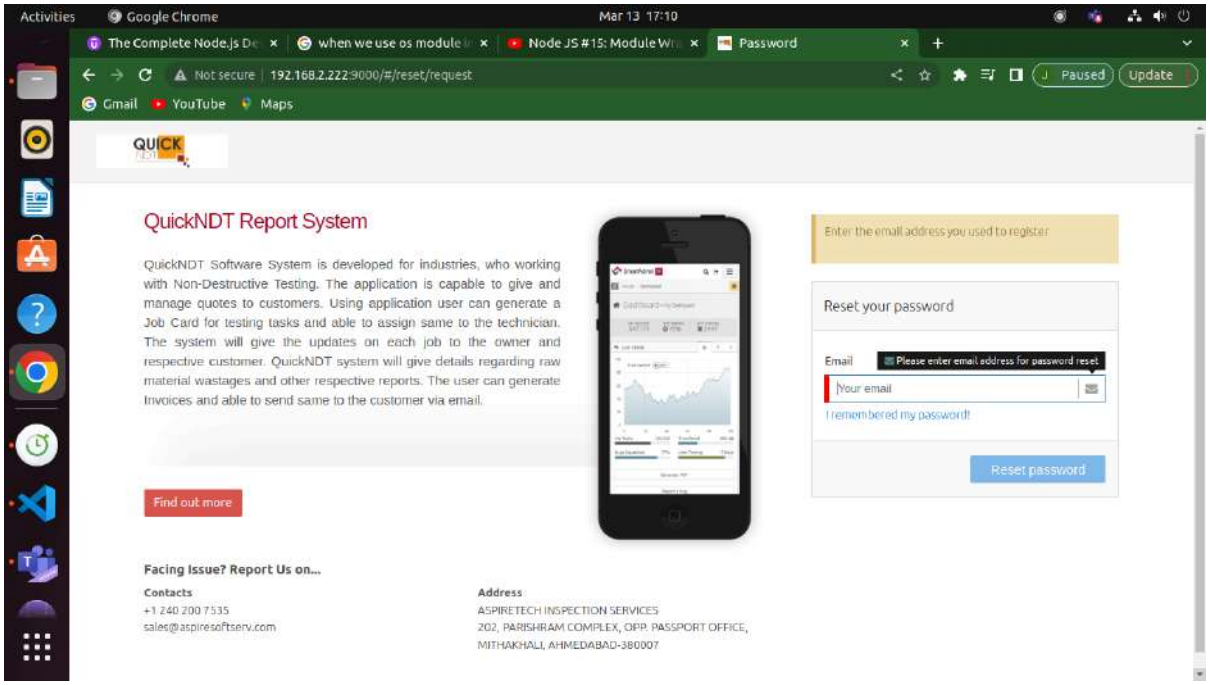



Fig 7.3 Profile Page

- User can view their profile info here

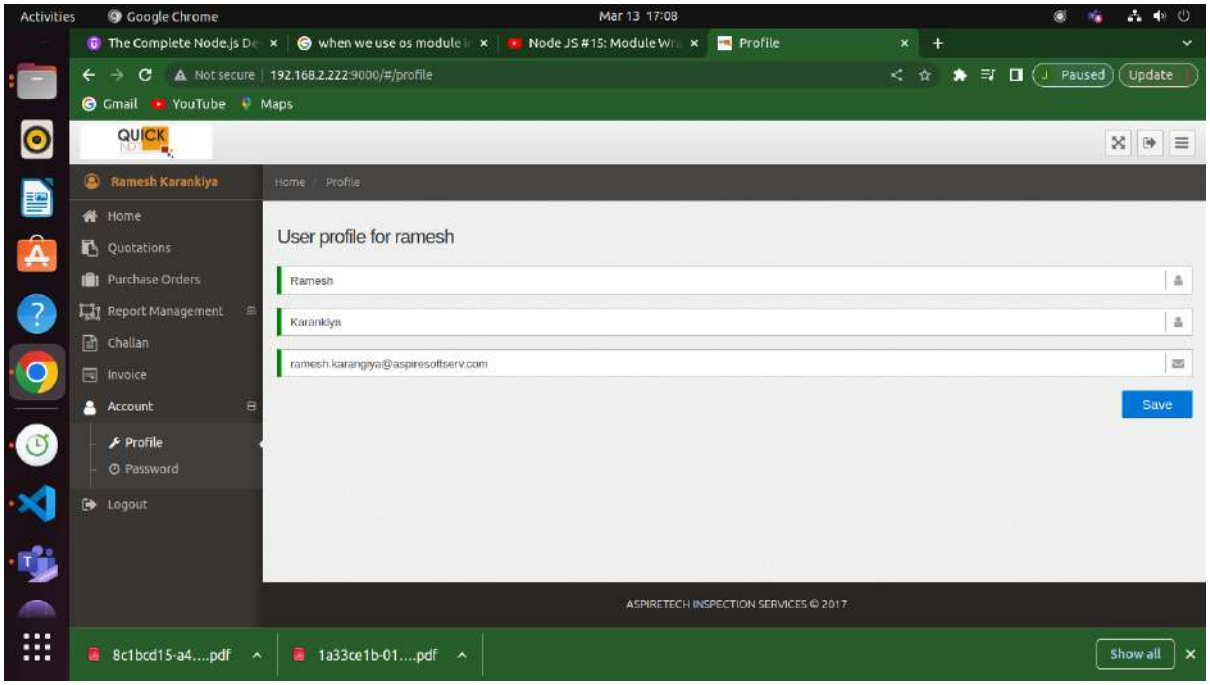


Fig 7.4 Password

- User can change their password in this page

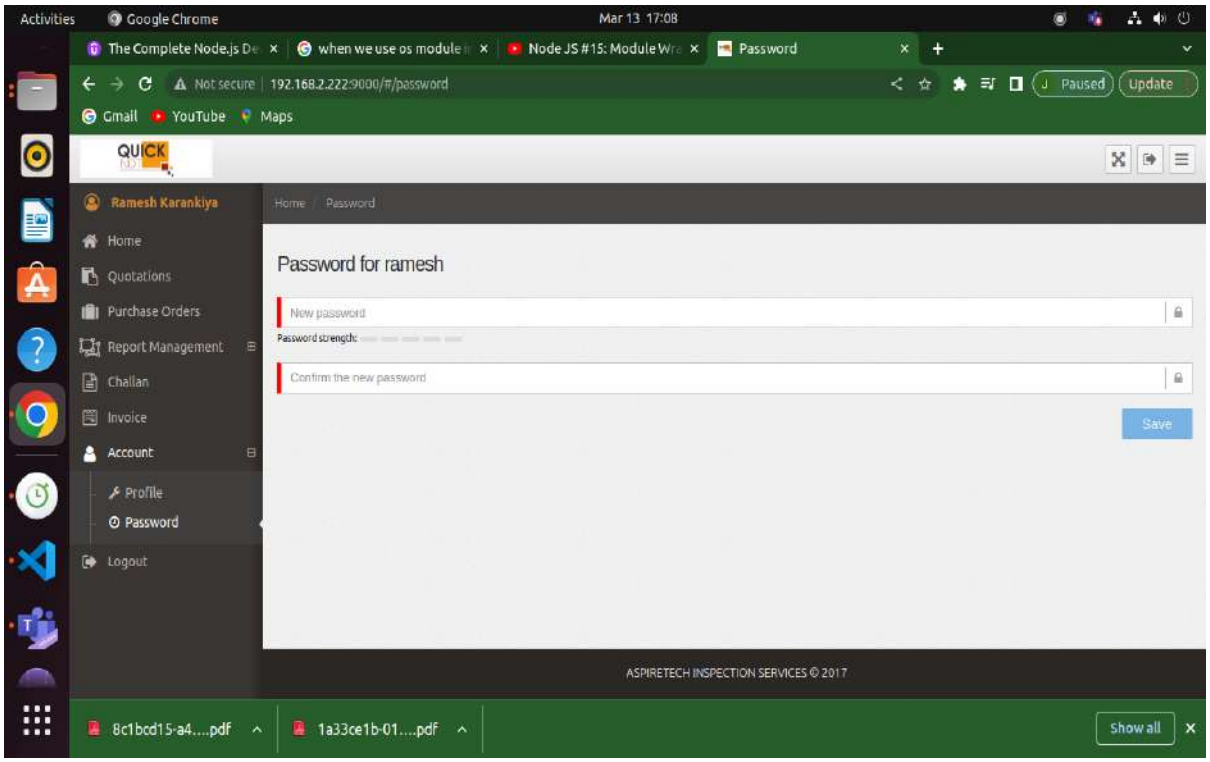



Fig 7.5 Role base Dashboard

- User dashboard based on roles

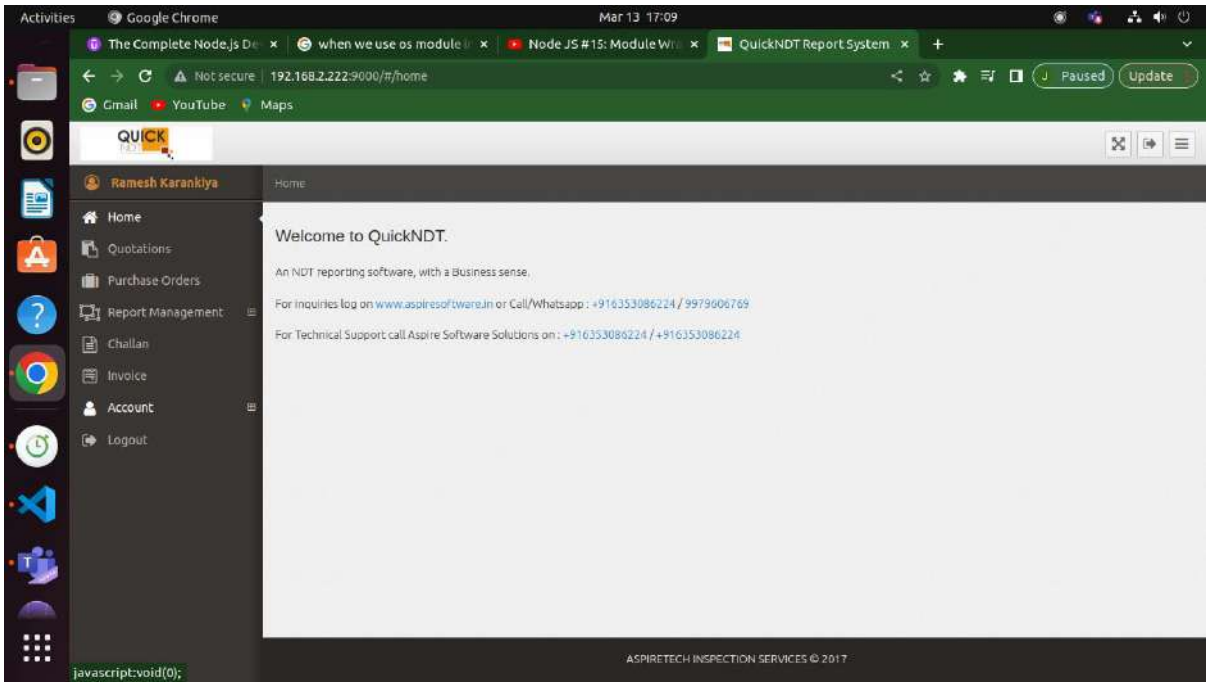


Fig 7.6 logout

- User can logout their account by this logout button.

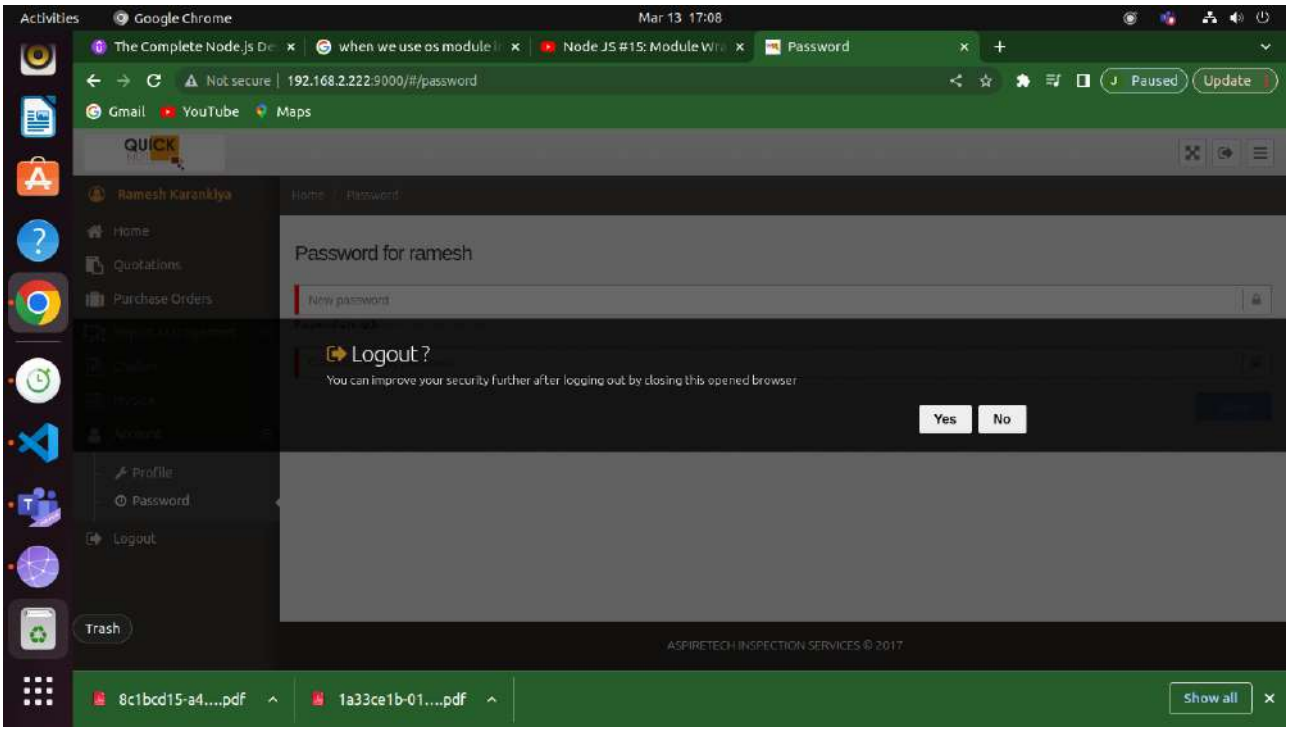



Fig 7.7 Report Management

- Here , in Report Management section there are four subsections which holds test data

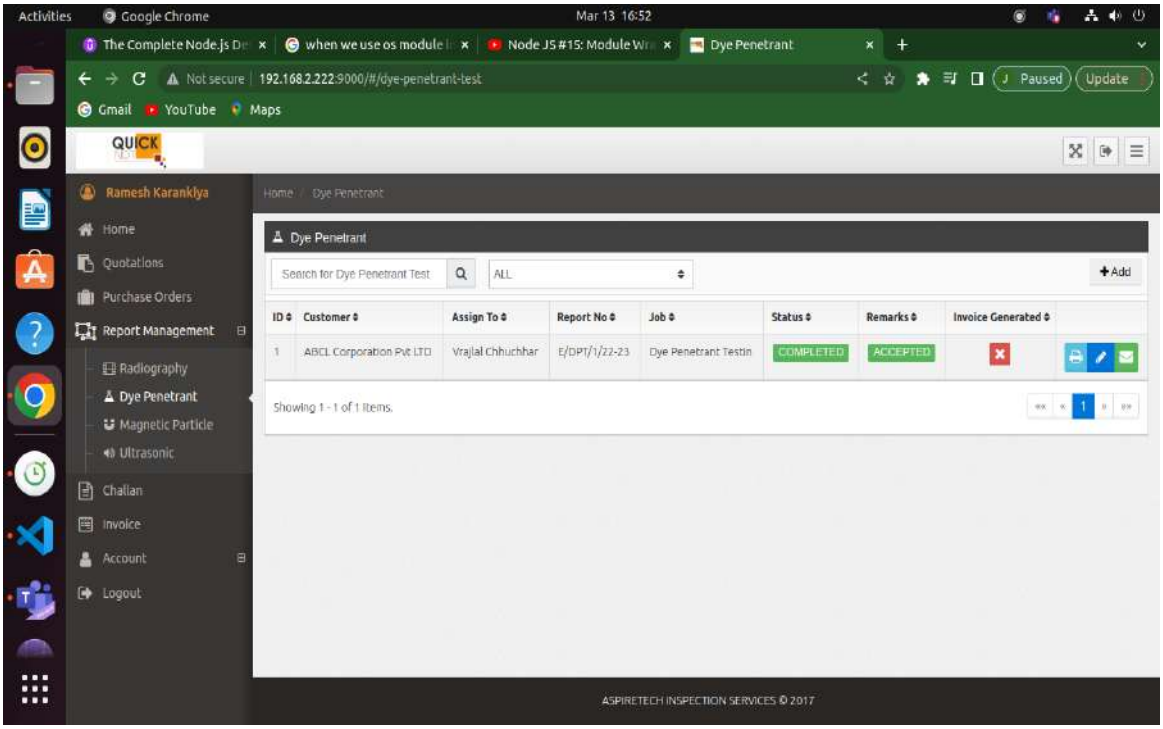


Fig 7.8 Radiography

- It holds data about Radiography Test

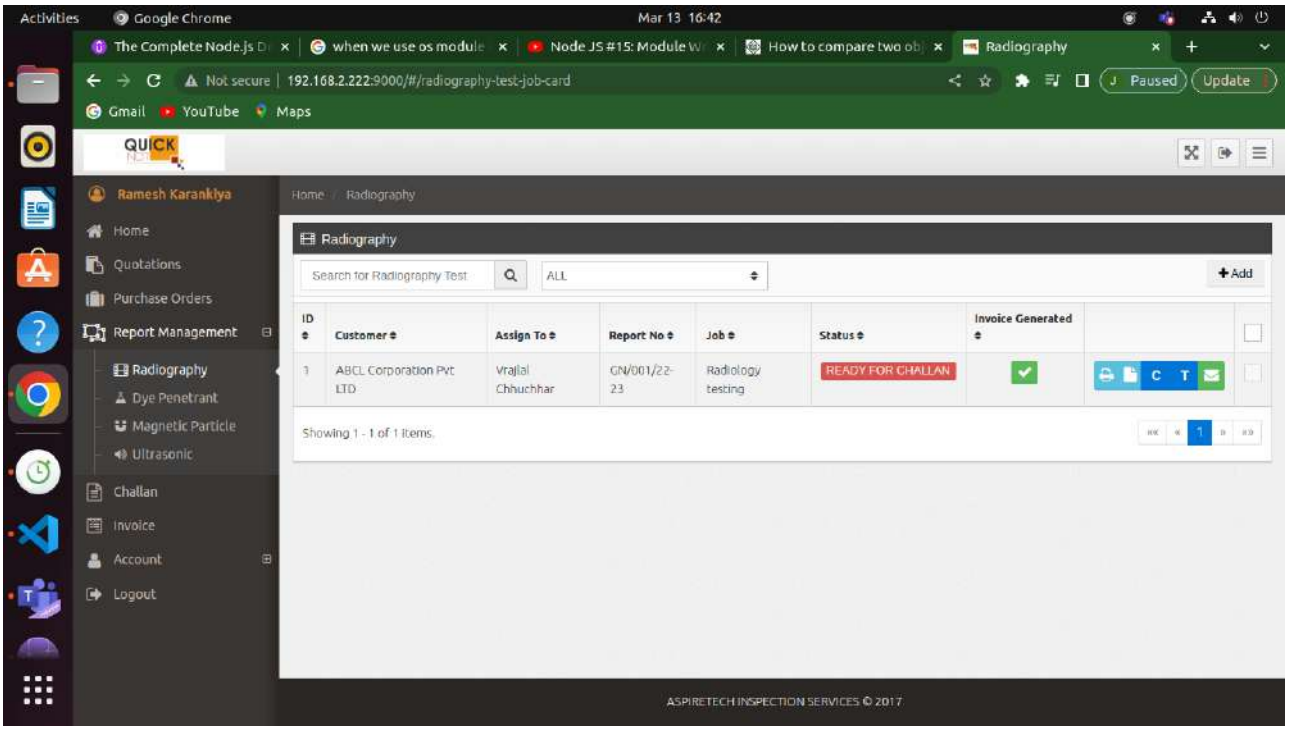



Fig 7.9 Radiography-2

- It holds data about Radiography Test

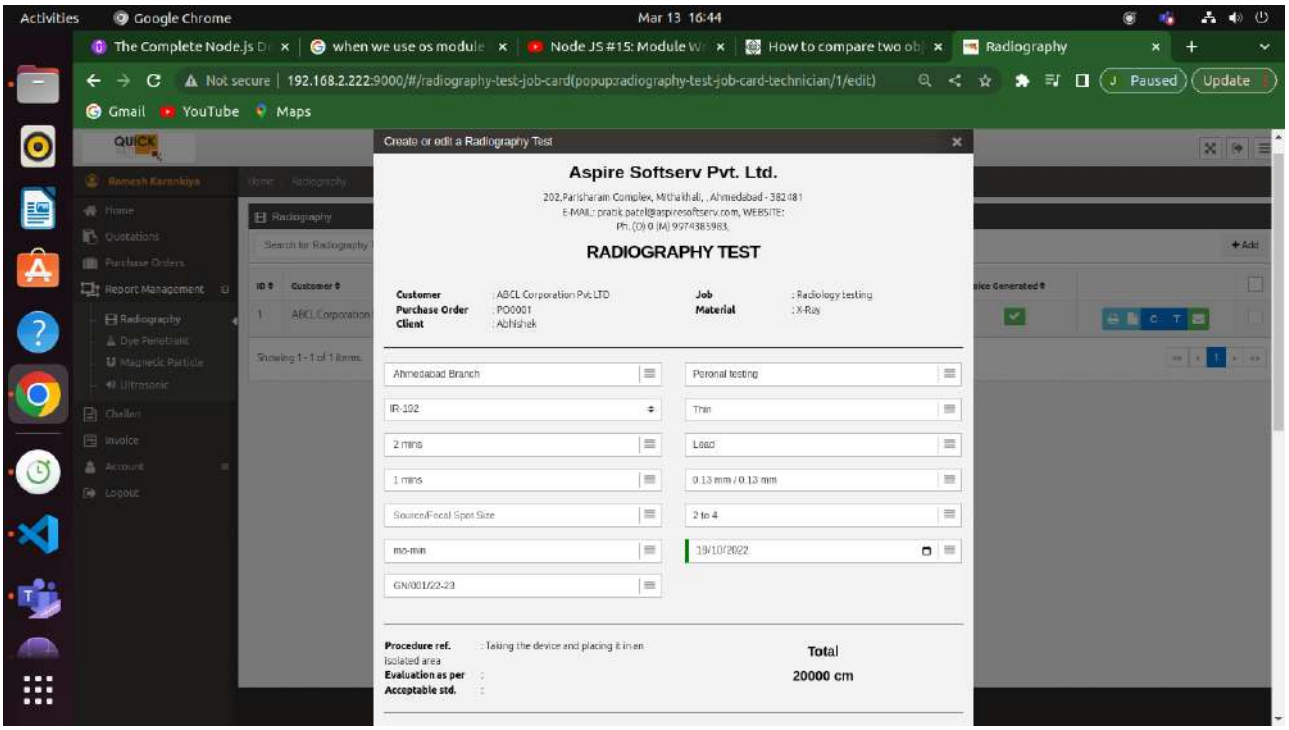


Fig 7.10 Dye Penetrate Test

- It holds data about Dye Penetrant

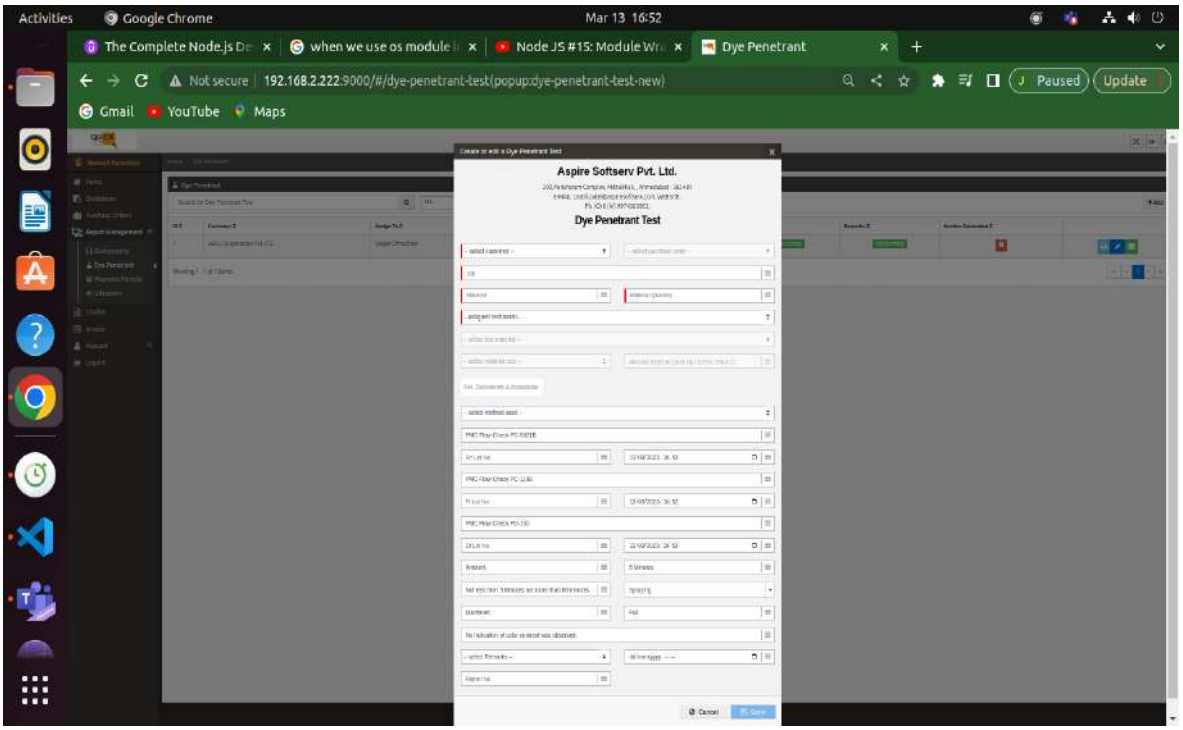



Fig 7.11 Dye Penetrate Test -2

- It holds data about Dye Penetrant

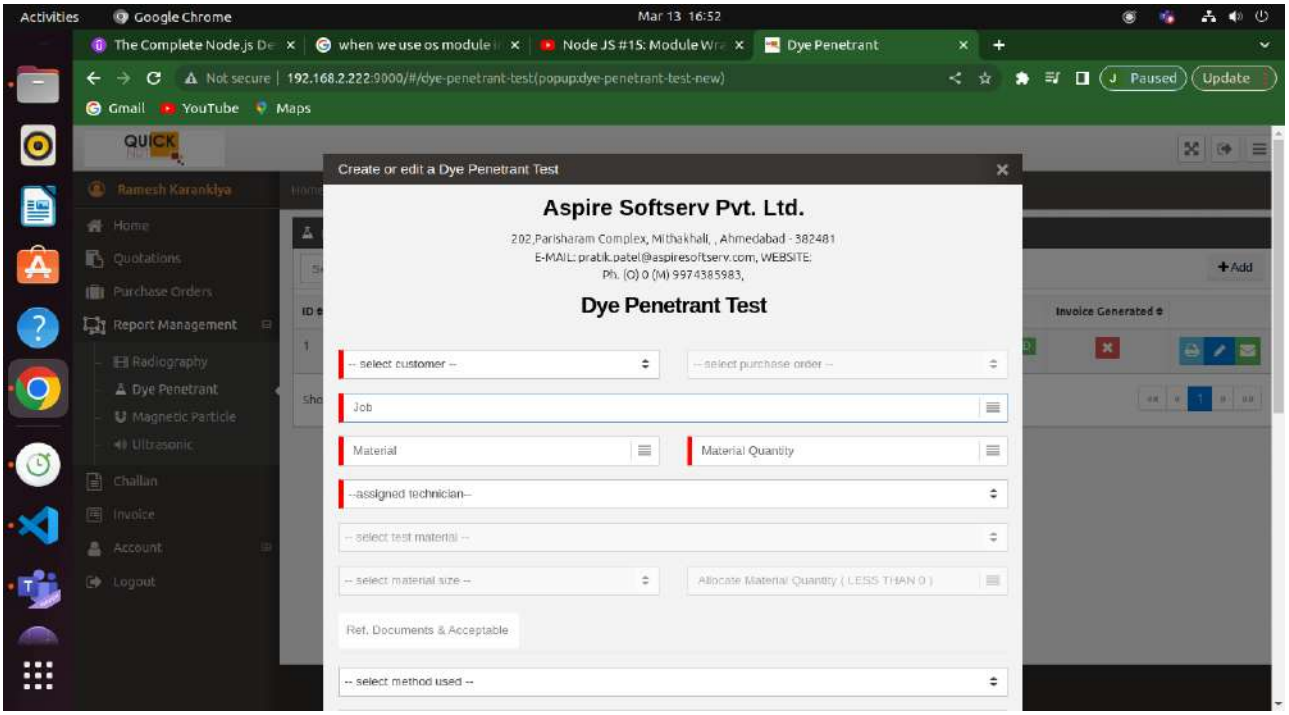


Fig 7.12 Magnetic Particle Test

- It holds data about Magnetic Particle test

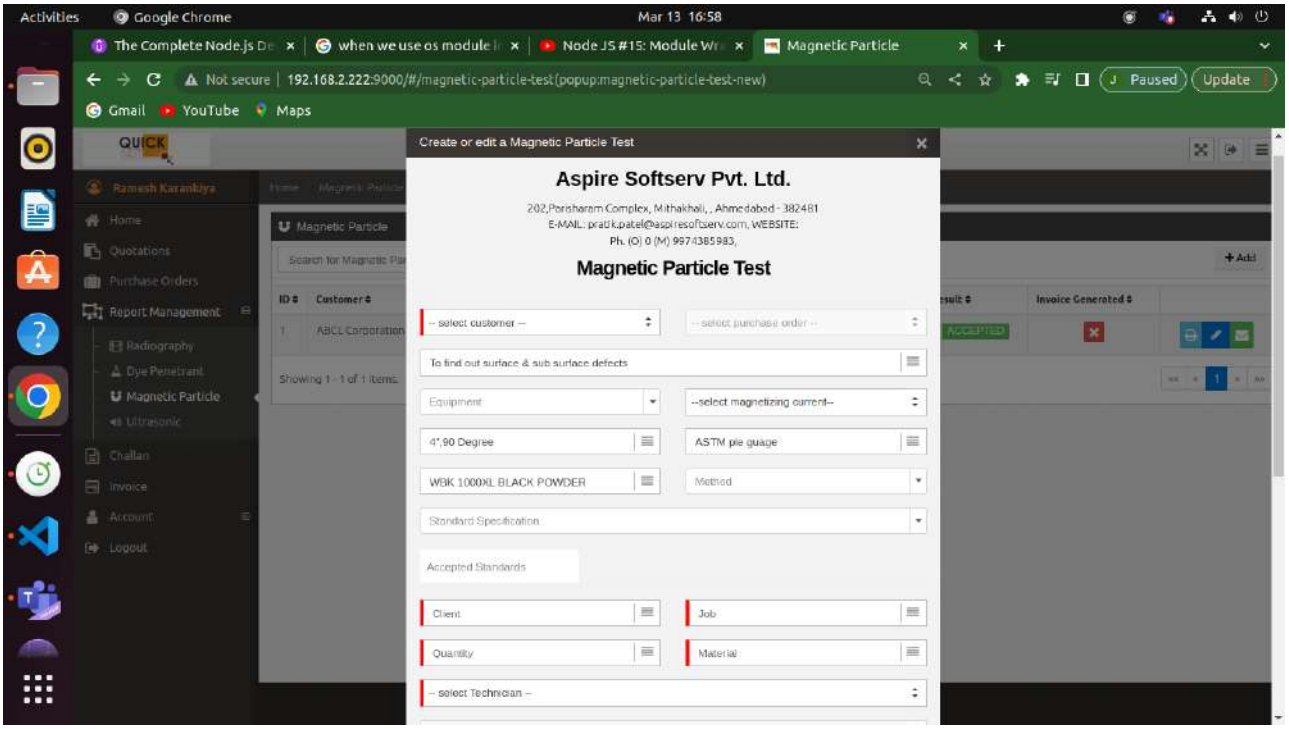



Fig 7.13 Magnetic Particle Test-2

- It holds data about Magnetic Particle test

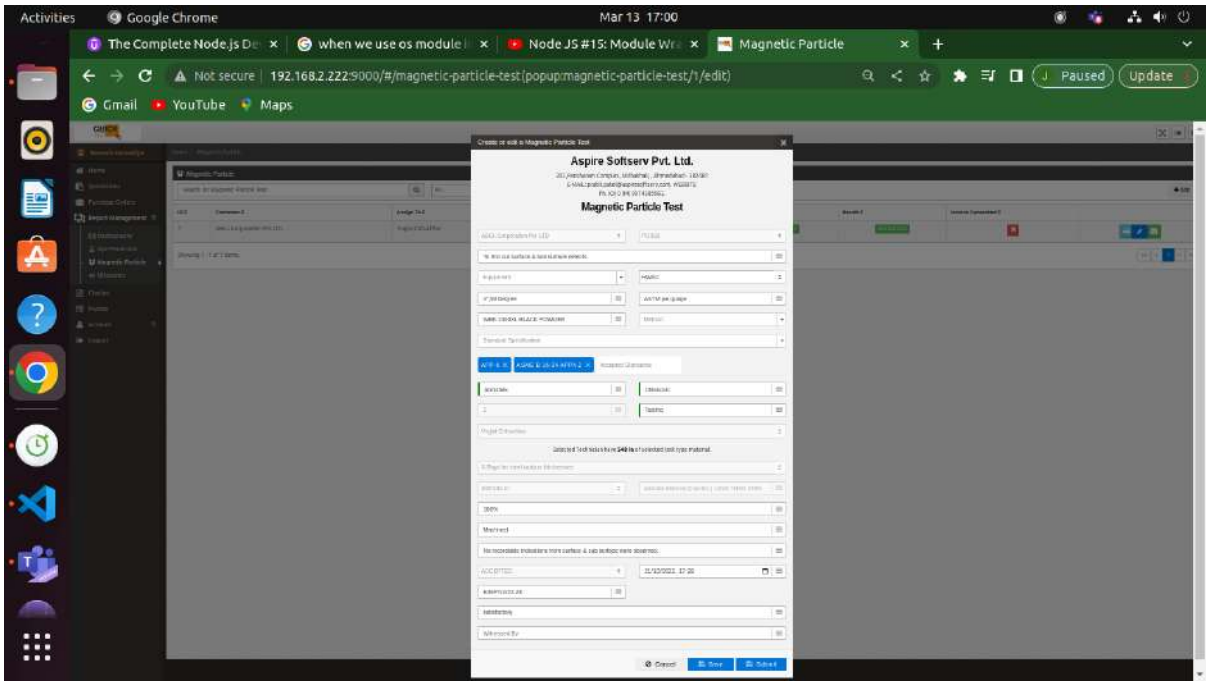


Fig 7.14 Ultrasonic Test

- It holds data about Ultrasonic Test

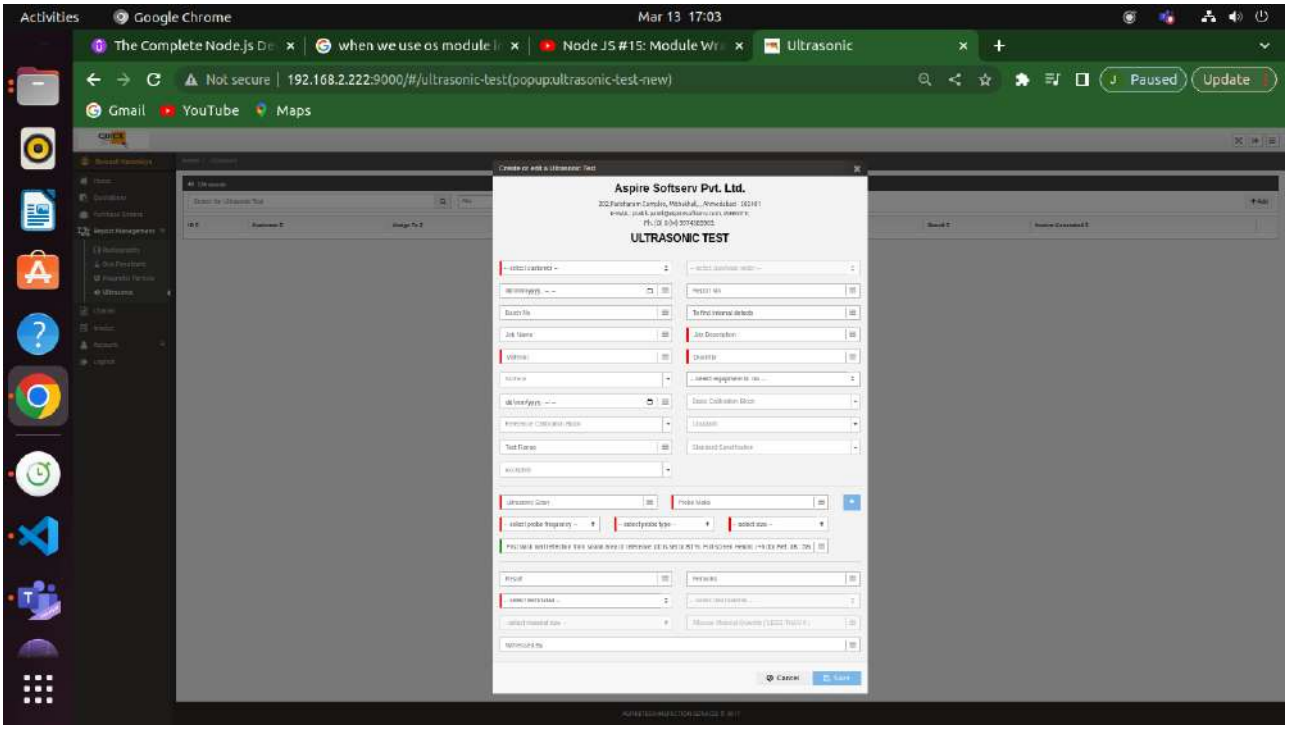



Fig 7.15 Invoice

- Here user can saw their invoices and print it out also

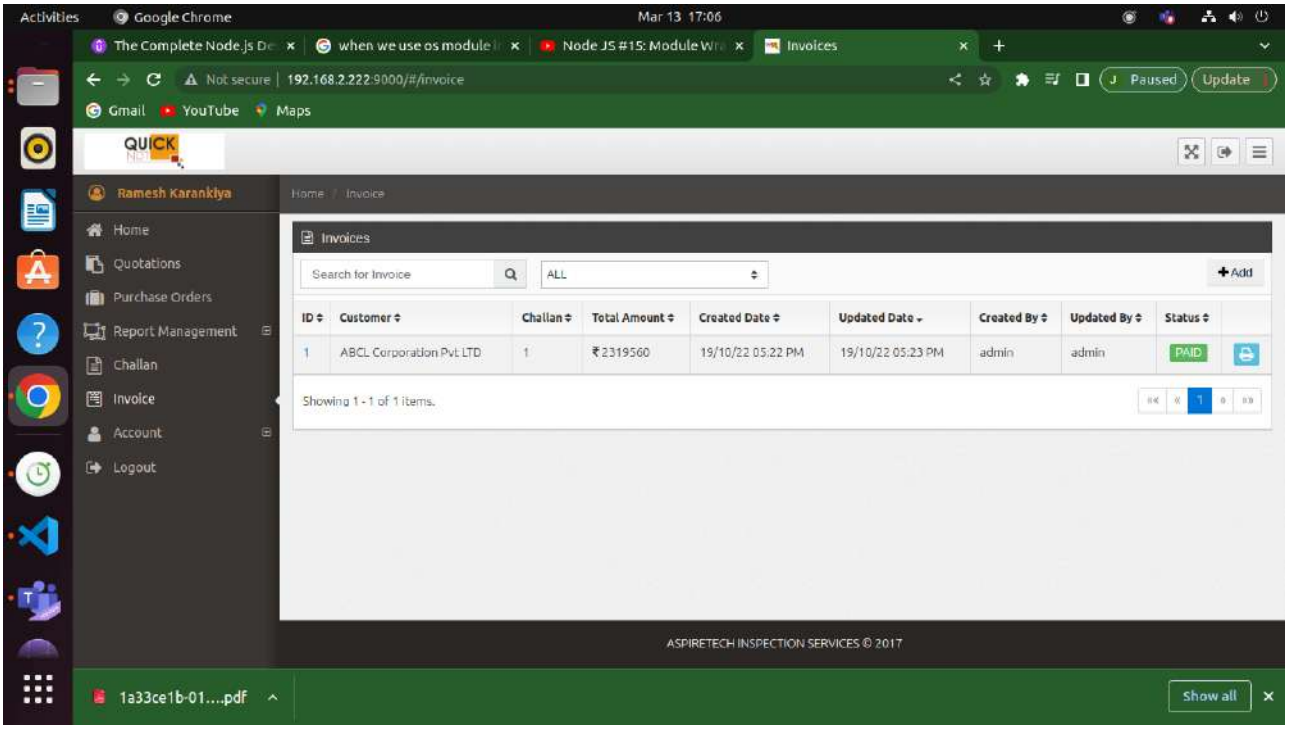


Fig 7.16 Invoice

- Add invoice in this page

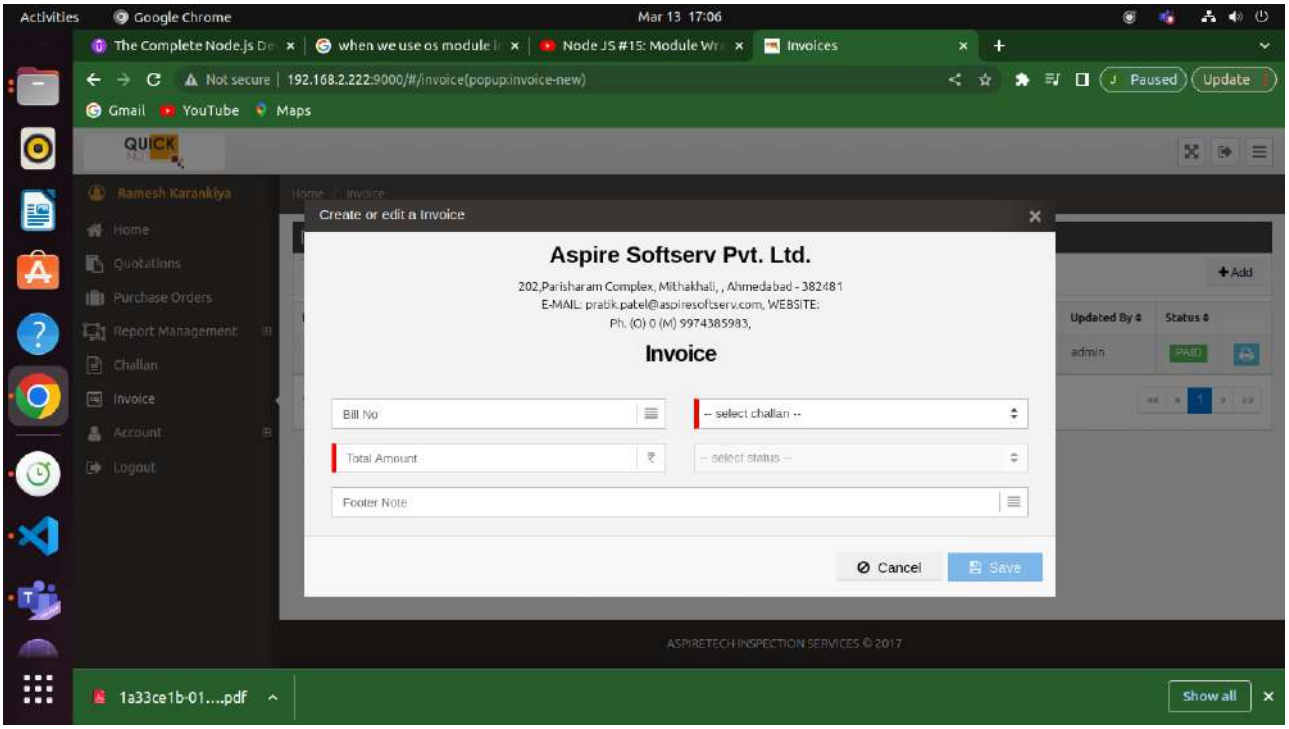



Fig 7.17 purchase order

- Here user can manage purchase orders .

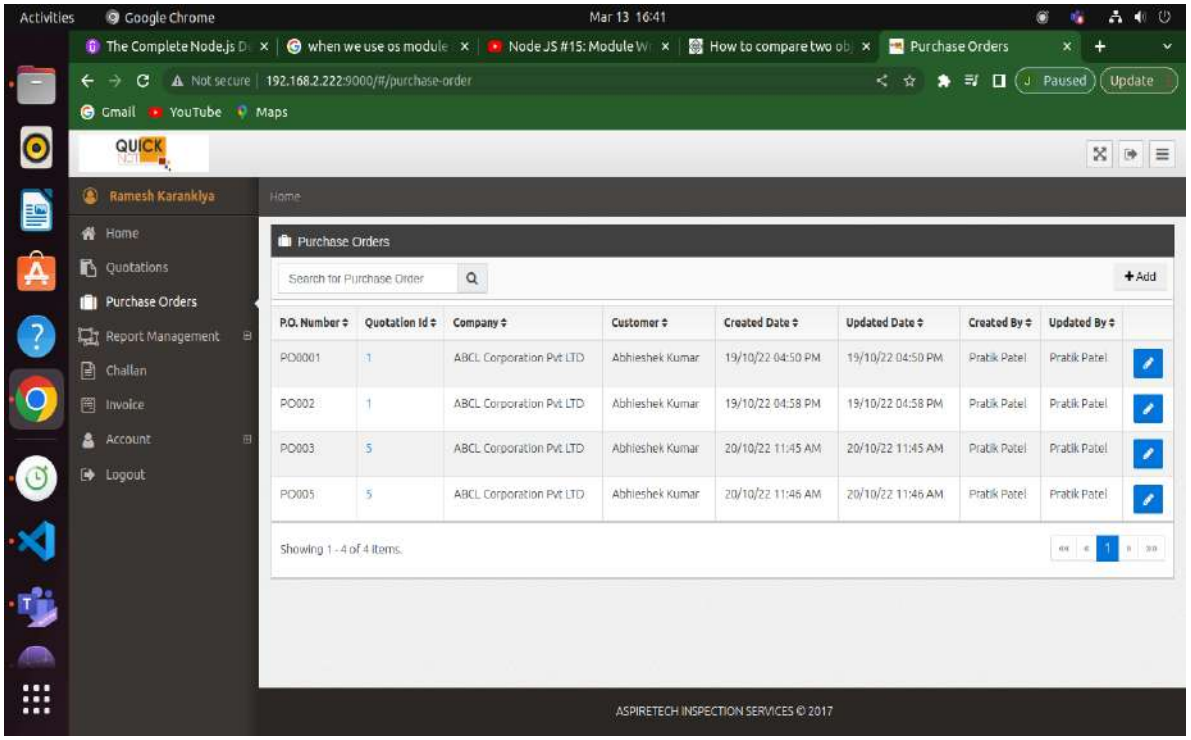


Fig 7.18 purchase order -2

- Here user can manage purchase orders .

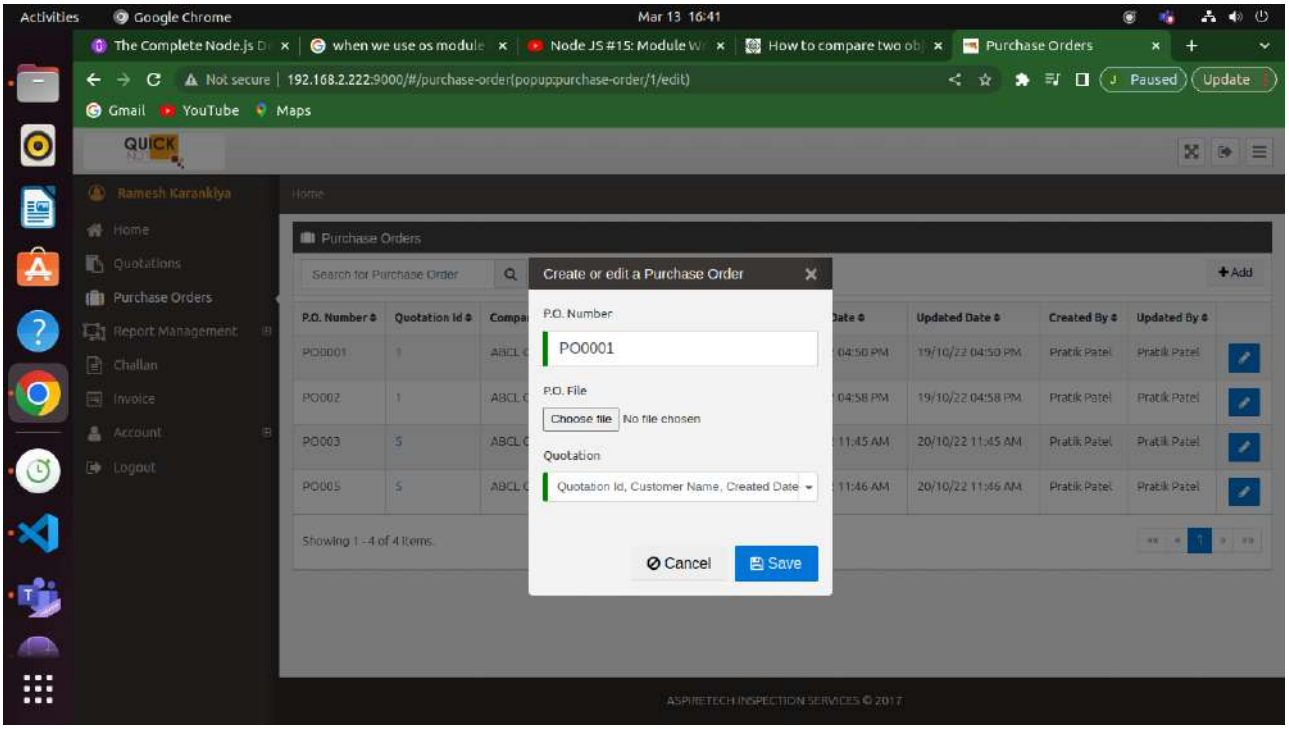



Fig 7.16 Quotation

- This page holds bank information, shift timing and contact details.

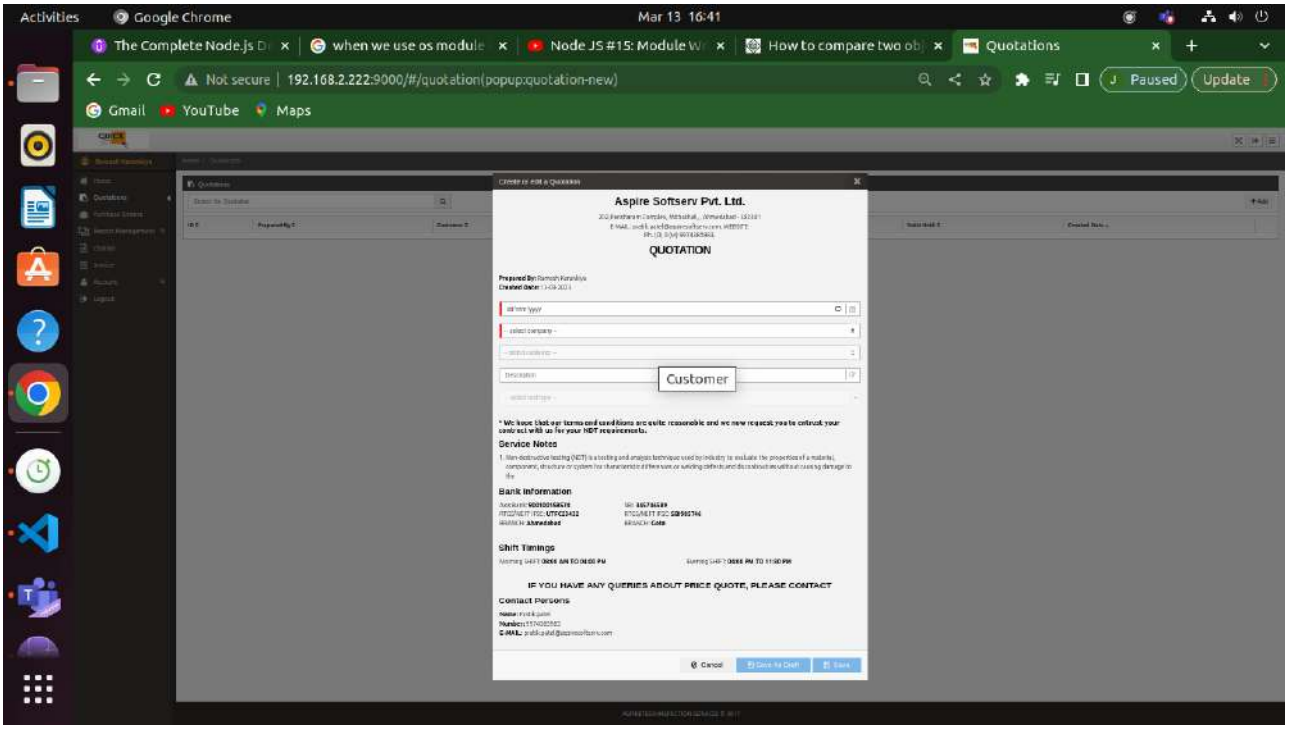


Fig 7.21 Admin dashboard

- Admin dashboard of NDT-report system

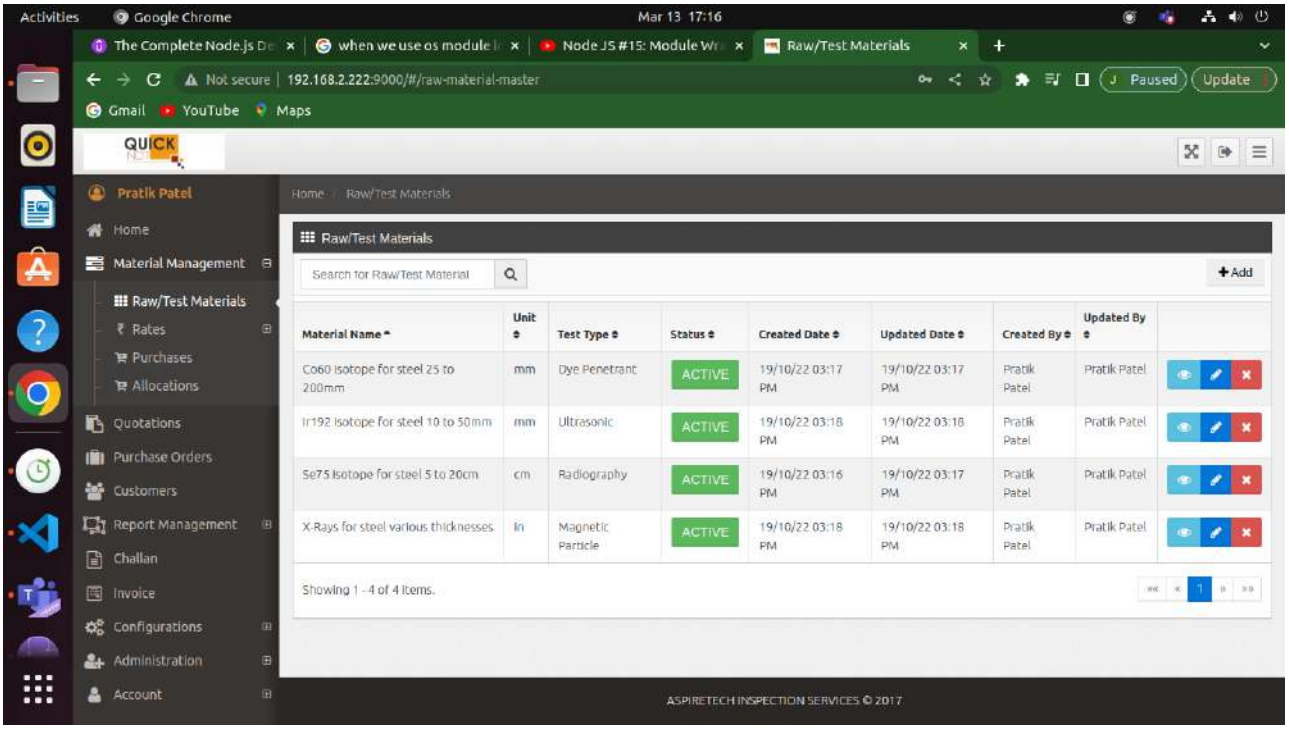



Fig 7.22 Customer Page

- Here Admin can manage customers

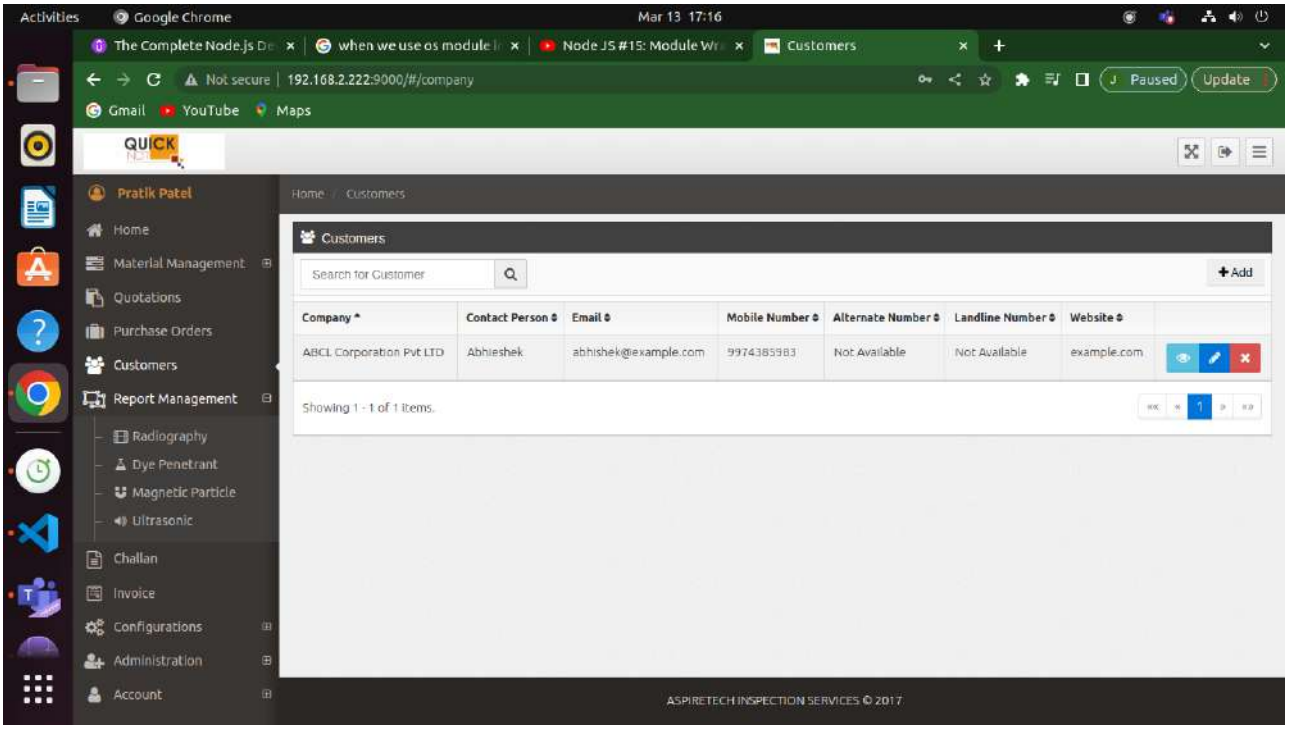
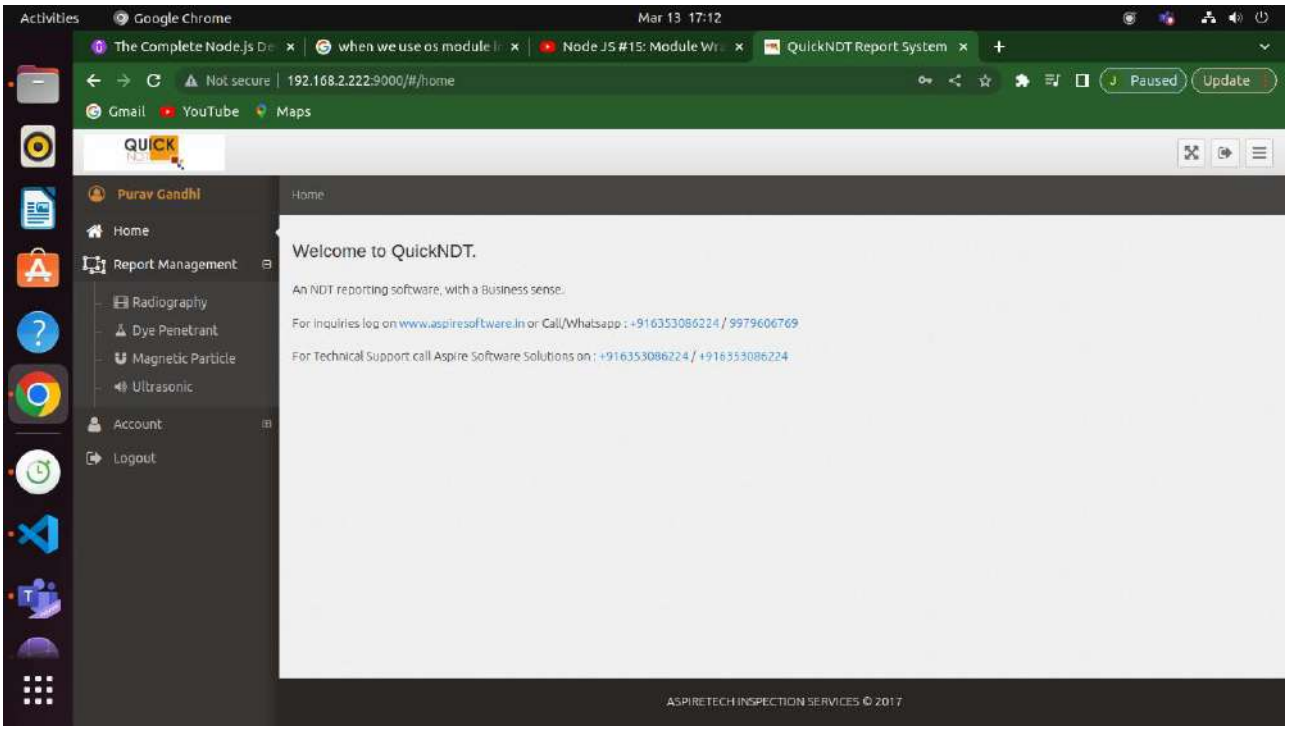


Fig 7.24 Technician Dashboard

- Technician Dashboard of NDT-report system



### *7.3. Security Features*

User Authentication and Authorization: The system requires users to log in with valid credentials and only grants access to authorized users. User roles and permissions are also defined to control access to specific system features and data.

Data Encryption: The system uses encryption algorithms to protect data in transit and at rest. This includes using HTTPS for secure communication and encrypting sensitive data such as passwords and reports.

Audit Trail: The system maintains an audit trail of all user activities, including login attempts, data access, and report generation. This helps to identify and investigate any unauthorized activities or system breaches.

Firewall and Network Security: The system is protected by a firewall and network security measures to prevent unauthorized access to the system from outside the network.

# 8

**Chapter # 8: TESTING**

# *8. Testing*

## *8.1. Testing Plan*

The testing technique that is going to be used in the project is White box testing. In White box testing the Tester has knowledge about the internal structure of the code or the program of the software.

***White Box Testing:***

It is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability, and security.

Out of the 2 methods for testing, black box testing and white box testing, we would be using the white box testing as we are well aware of the internal functionalities of our application unlike in the black box testing.

## *8.2. Testing Strategy*

The development process repeats this testing subprocess a number of times for the following phases.

### *8.2.1. Unit Testing:*

It ensures that all code meets quality standards before it's deployed. Also, it detects software bugs earlier.

### *8.2.2. Integration Testing:*

It tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

## *8.3. Test Cases*

Table 8.1 Testcase Table 1

| | |
|---|---|
| Test Case Id: | TC001 |
| Test Case Summary | The user must have a valid username and password. |

| | |
|---|---|
| Prerequisite | The user must have a valid email address and password |
| Test Procedure | 1. Open the application login page.<br>2. Enter the valid username in the username field.<br>3. Enter the valid password in the password field.<br>4. Click on the login button.<br>5. Wait for the login process to complete. |
| Test Data | Valid username: admin<br>Valid password: admin |
| Expected Result | ● The user should be logged in successfully.<br>● The user should be redirected to the application dashboard page. |
| Actual Result | ● The user is logged in successfully.<br>● The user is redirected to the application dash-board page.<br>● The application displays a welcome message with the user's name. |
| Status | Pass |
| Remark | N/A |

Table 8.2 Testcase Table 2

| | |
|---|---|
| Test Case Id: | TC002 |
| Test Case Summary | Verify the functionality of adding, editing, deleting, activating, and deactivating raw material data. |
| Prerequisite | The user must have appropriate access rights to add, edit, delete, activate, and deactivate raw material data. |
| Test Procedure | 1. Launch the application and navigate to the raw material data page. |

| | |
|---|---|
| | 2. Click on the "Add" button.<br>3. Enter the required information in the respective fields.<br>4. Click on the "Save" button.<br>5. Verify that the raw material data is added successfully.<br>6. Click on the "Edit" button for the added raw material data.<br>7. Modify the required information.<br>8. Click on the "Save" button.<br>9. Verify that the raw material data is edited successfully.<br>10. Click on the "Delete" button for the added raw material data.<br>11. Verify that the user receives a confirmation message.<br>12. Click on the "Confirm" button to delete the raw material data.<br>13. Verify that the raw material data is deleted successfully.<br>14. Click on the "Activate" button for the deactivated raw material data.<br>15. Verify that the raw material data is activated successfully.<br>16. Click on the "Deactivate" button for the activated raw material data.<br>17. Verify that the raw material data is deactivated successfully. |
| Test Data | ● Valid raw material data<br>● Modified raw material data<br>● Deactivated raw material data |
| Expected Result | 1. The user should be able to access the raw material data page.<br>2. The user should be able to add the raw material |

| | |
|---|---|
| | data.<br>3. The raw material data should be added successfully.<br>4. The user should be able to edit the raw material data.<br>5. The raw material data should be edited successfully.<br>6. The user should be able to delete the raw material data.<br>7. The user should receive a confirmation message before deleting the raw material data.<br>8. The raw material data should be deleted successfully.<br>9. The user should be able to activate the deactivated raw material data.<br>10. The raw material data should be activated successfully.<br>11. The user should be able to deactivate the activated raw material data.<br>12. The raw material data should be deactivated successfully. |
| Actual Result | 1. The user is able to access the raw material data page.<br>2. The user is able to add the raw material data.<br>3. The raw material data is added successfully.<br>4. The user is able to edit the raw material data.<br>5. The raw material data is edited successfully.<br>6. The user is able to delete the raw material data.<br>7. The user receives a confirmation message before deleting the raw material data.<br>8. The raw material data is deleted successfully.<br>9. The user is able to activate the deactivated raw material data.<br>10. The raw material data is activated successfully.<br>11. The user is able to deactivate the activated raw material data.<br>12. The raw material data is deactivated successfully. |

| | |
|---|---|
| Status | Pass |
| Remark | N/A |

Table 8.3 Testcase Table 3

| | |
|---|---|
| Test Case Id: | TC003 |
| Test Case Summary | Verify the functionality of editing and deleting the raw material rate |
| Prerequisite | The user must have appropriate access rights to edit and delete raw material rates. |
| Test Procedure | 1. Launch the application and navigate to the raw material rate page.<br>2. Click on the "Edit" button for the raw material rate.<br>3. Modify the required information.<br>4. Click on the "Save" button.<br>5. Verify that the raw material rate is edited successfully.<br>6. Click on the "Delete" button for the raw material rate.<br>7. Verify that the user receives a confirmation message.<br>8. Click on the "Confirm" button to delete the raw material rate.<br>9. Verify that the raw material rate is deleted successfully. |
| Test Data | ● Modified raw material rate<br>● Deleted raw material rate |
| Expected Result | 1. The user should be able to access the raw material rate page.<br>2. The user should be able to edit the raw material rate.<br>3. The raw material rate should be edited |

| | |
|---|---|
| | successfully.<br>4. The user should be able to delete the raw material rate.<br>5. The user should receive a confirmation message before deleting the raw material rate.<br>6. The raw material rate should be deleted successfully. |
| Actual Result | 1. The user is able to access the raw material rate page.<br>2. The user is able to edit the raw material rate.<br>3. The raw material rate is edited successfully.<br>4. The user is able to delete the raw material rate.<br>5. The user receives a confirmation message before deleting the raw material rate.<br>6. The raw material rate is deleted successfully. |
| Status | Pass |
| Remark | N/A |

Table 8.4 Testcase Table 4

| | |
|---|---|
| Test Case Id: | TC004 |
| Test Case Summary | Verify the functionality of adding, updating, and deleting raw material purchase data, and generating a unique purchase ID. |
| Prerequisite | The user must have appropriate access rights to add, update, and delete raw material purchase data. |
| Test Procedure | 1. Launch the application and navigate to the raw material purchase page.<br>2. Click on the "Add" button.<br>3. Enter the required information in the respective |

| | |
|---|---|
| | fields.<br>4. Click on the "Save" button.<br>5. Verify that the raw material purchase data is added successfully.<br>6. Click on the "Edit" button for the added raw material purchase data.<br>7. Modify the required information.<br>8. Click on the "Save" button.<br>9. Verify that the raw material purchase data is updated successfully.<br>10. Click on the "Delete" button for the added raw material purchase data.<br>11. Verify that the user receives a confirmation message.<br>12. Click on the "Confirm" button to delete the raw material purchase data.<br>13. Verify that the raw material purchase data is deleted successfully.<br>14. Verify that a unique purchase ID is generated every time a raw material purchase is added. |
| Test Data | ● Valid raw material purchase data<br>● Modified raw material purchase data<br>● Deleted raw material purchase data |
| Expected Result | 1. The user should be able to access the raw material purchase page.<br>2. The user should be able to add the raw material purchase data.<br>3. The raw material purchase data should be added successfully.<br>4. The user should be able to edit the raw material purchase data.<br>5. The raw material purchase data should be updated successfully.<br>6. The user should be able to delete the raw material purchase data.<br>7. The user should receive a confirmation message before deleting the raw material purchase data.<br>8. The raw material purchase data should be deleted |

| | |
|---|---|
| | successfully.<br>9. A unique purchase ID should be generated every time a raw material purchase is added. |
| Actual Result | 1. The user is able to access the raw material purchase page.<br>2. The user is able to add the raw material purchase data.<br>3. The raw material purchase data is added successfully.<br>4. The user is able to edit the raw material purchase data.<br>5. The raw material purchase data is updated successfully.<br>6. The user is able to delete the raw material purchase data.<br>7. The user receives a confirmation message before deleting the raw material purchase data.<br>8. The raw material purchase data is deleted successfully.<br>9. A unique purchase ID is generated every time a raw material purchase is added. |
| Status | Pass |
| Remark | N/A |

Table 8.5 Testcase Table 5

| | |
|---|---|
| Test Case Id: | TC005 |
| Test Case Summary | Verify the functionality of adding, updating, and deleting raw material allocation data to assign raw material to a technician for testing. |
| Prerequisite | The user must have appropriate access rights to add, update, and delete raw material allocation data. |

| | |
|---|---|
| Test Procedure | 1. Launch the application and navigate to the raw material allocation page.<br>2. Click on the "Add" button.<br>3. Enter the required information in the respective fields.<br>4. Click on the "Save" button.<br>5. Verify that the raw material allocation data is added successfully.<br>6. Click on the "Edit" button for the added raw material allocation data.<br>7. Modify the required information.<br>8. Click on the "Save" button.<br>9. Verify that the raw material allocation data is updated successfully.<br>10. Click on the "Delete" button for the added raw material allocation data.<br>11. Verify that the user receives a confirmation message.<br>12. Click on the "Confirm" button to delete the raw material allocation data.<br>13. Verify that the raw material allocation data is deleted successfully. |
| Test Data | ● Valid raw material allocation data<br>● Modified raw material allocation data<br>● Deleted raw material allocation data |
| Expected Result | 1. The user should be able to access the raw material allocation page.<br>2. The user should be able to add the raw material allocation data.<br>3. The raw material allocation data should be added successfully.<br>4. The user should be able to edit the raw material allocation data.<br>5. The raw material allocation data should be updated successfully.<br>6. The user should be able to delete the raw material allocation data.<br>7. The user should receive a confirmation message |

| | |
|---|---|
| | before deleting the raw material allocation data.<br>8. The raw material allocation data should be deleted successfully. |
| Actual Result | 1. The user is able to access the raw material allocation page.<br>2. The user is able to add the raw material allocation data.<br>3. The raw material allocation data is added successfully.<br>4. The user is able to edit the raw material allocation data.<br>5. The raw material allocation data is updated successfully.<br>6. The user is able to delete the raw material allocation data.<br>7. The user receives a confirmation message before deleting the raw material allocation data.<br>8. The raw material allocation data is deleted successfully. |
| Status | Pass |
| Remark | N/A |

Table 8.6 Testcase Table 6

| | |
|---|---|
| Test Case Id: | TC006 |
| Test Case Summary | Verify the functionality of add, update, and delete customer data in the customers section. |
| Prerequisite | The user must have appropriate access rights to add, update, and delete customer data. |
| Test Procedure | 1. Launch the application and navigate to the customers section.<br>2. Click on the "Add" button. |

| | |
|---|---|
| | 3. Enter the required customer information in the respective fields.<br>4. Click on the "Save" button.<br>5. Verify that the customer data is added successfully.<br>6. Click on the "Edit" button for the added customer data.<br>7. Modify the required customer information.<br>8. Click on the "Save" button.<br>9. Verify that the customer data is updated successfully.<br>10. Click on the "Delete" button for the added customer data.<br>11. Verify that the user receives a confirmation message.<br>12. Click on the "Confirm" button to delete the customer data.<br>13. Verify that the customer data is deleted successfully. |
| Test Data | ● Valid customer data<br>● Modified customer data<br>● Deleted customer data |
| Expected Result | 1. The user should be able to access the customers section.<br>2. The user should be able to add the customer data.<br>3. The customer data should be added successfully.<br>4. The user should be able to edit the customer data.<br>5. The customer data should be updated successfully.<br>6. The user should be able to delete the customer data.<br>7. The customer data should be deleted successfully. |
| Actual Result | 1. The user is able to access the customers section.<br>2. The user is able to add the customer data.<br>3. The customer data is added successfully.<br>4. The user is able to edit the customer data. |

| | |
|---|---|
| | 5. The customer data is updated successfully.<br>6. The user is able to delete the customer data.<br>7. The customer data is deleted successfully. |
| Status | Pass |
| Remark | N/A |

Table 8.7 Testcase Table 7

| | |
|---|---|
| Test Case Id: | TC007 |
| Test Case Summary | Verify the functionality of add, update, delete, set status completed or not, set result accepted or not, set invoice generated or not, printout, and mail in the four sub-sections of report management. |
| Prerequisite | The user must have appropriate access rights to add, update, and delete report data. |
| Test Procedure | 1. Launch the application and navigate to the report management section.<br>2. Click on the desired sub-section (Radiography, Dye Penetrant, Magnetic Particle, or Ultrasonic).<br>3. Click on the "Add" button.<br>4. Enter the required information in the respective fields.<br>5. Click on the "Save" button.<br>6. Verify that the report data is added successfully.<br>7. Click on the "Edit" button for the added report data.<br>8. Modify the required information.<br>9. Click on the "Save" button.<br>10. Verify that the report data is updated successfully.<br>11. Click on the "Delete" button for the added report data.<br>12. Verify that the user receives a confirmation message. |

| | |
|---|---|
| | 13. Click on the "Confirm" button to delete the report data.<br>14. Verify that the report data is deleted successfully.<br>15. Set the status of the report as "completed".<br>16. Verify that the status of the report is updated successfully.<br>17. Set the result of the report as "accepted".<br>18. Verify that the result of the report is updated successfully.<br>19. Set the invoice generated flag as "true".<br>20. Verify that the invoice generated flag is updated successfully.<br>21. Click on the "Print" button to generate a printout of the report.<br>22. Verify that the report is printed successfully.<br>23. Click on the "Mail" button to mail the report.<br>24. Verify that the report is mailed successfully. |
| Test Data | ● Valid report data<br>● Modified report data<br>● Deleted report data<br>● Completed status flag<br>● Accepted result flag<br>● Invoice generated flag<br>● Printed report<br>● Mailed report |
| Expected Result | 1. The user should be able to access the report management section.<br>2. The user should be able to access the desired subsection.<br>3. The user should be able to add the report data.<br>4. The report data should be added successfully.<br>5. The user should be able to edit the report data.<br>6. The report data should be updated successfully.<br>7. The user should be able to delete the report data.<br>8. The report data should be deleted successfully.<br>9. The user should be able to set the status of the report as completed. |

| | |
|---|---|
| | 10. The status of the report should be updated successfully.<br>11. The user should be able to set the result of the report as accepted.<br>12. The result of the report should be updated successfully.<br>13. The user should be able to set the invoice generated flag as true.<br>14. The invoice generated flag should be updated successfully.<br>15. The user should be able to generate a printout of the report.<br>16. The report should be printed successfully.<br>17. The user should be able to mail the report.<br>18. The report should be mailed successfully. |
| Actual Result | 1. The user is able to access the report management section.<br>2. The user is able to access the desired sub-section.<br>3. The user is able to add the report data.<br>4. The report data is added successfully.<br>5. The user is able to edit the report data.<br>6. The report data is updated successfully.<br>7. The user is able to delete the report data.<br>8. The report data is deleted successfully.<br>9. The user is able to set the status of the report as completed.<br>10. The status of the report is updated |
| Status | Pass |
| Remark | N/A |

Table 8.8 Testcase Table 8

| | |
|---|---|
| Test Case Id: | TC008 |
| Test Case Summary | Verify the functionality of adding challan data and generating a printout in the challan section. |
| Prerequisite | ● The user must have appropriate access rights to add challan data and generate printouts. |
| Test Procedure | 1. Launch the application and navigate to the challan section.<br>2. Click on the "Add" button.<br>3. Enter the required challan information in the respective fields.<br>4. Click on the "Save" button.<br>5. Verify that the challan data is added successfully.<br>6. Click on the "Print" button for the added challan data.<br>7. Verify that the user is able to generate a printout of the challan data. |
| Test Data | ● Valid challan data |
| Expected Result | 1. The user should be able to access the challan section.<br>2. The user should be able to add the challan data.<br>3. The challan data should be added successfully.<br>4. The user should be able to generate a printout of the challan data. |
| Actual Result | 1. The user is able to access the challan section.<br>2. The user is able to add the challan data.<br>3. The challan data is added successfully.<br>4. The user is able to generate a printout of the challan data. |
| Status | Pass |
| Remark | N/A |

Table 8.9 Testcase Table 9

| | |
|---|---|
| Test Case Id: | TC009 |
| Test Case Summary | Verify the functionality of adding invoice data, setting status and generating a printout in the invoice section. |
| Prerequisite | The user must have appropriate access rights to add invoice data and generate printouts. |
| Test Procedure | 1. Launch the application and navigate to the invoice section.<br>2. Click on the "Add" button.<br>3. Enter the required invoice information in the respective fields such as customer name, chalan id, amount.<br>4. Click on the "Save" button.<br>5. Verify that the invoice data is added successfully.<br>6. Set the status of the invoice to either paid or unpaid.<br>7. Click on the "Print" button for the added invoice data.<br>8. Verify that the user is able to generate a printout of the invoice data. |
| Test Data | ● Valid invoice data |
| Expected Result | 1. The user should be able to access the invoice section.<br>2. The user should be able to add the invoice data.<br>3. The invoice data should be added successfully.<br>4. The user should be able to set the status of the invoice data to paid or unpaid.<br>5. The user should be able to generate a printout of the invoice data. |
| Actual Result | 1. The user is able to access the invoice section.<br>2. The user is able to add the invoice data.<br>3. The invoice data is added successfully. |

| | |
|---|---|
| | 4. The user is able to set the status of the invoice data to paid or unpaid.<br>5. The user is able to generate a printout of the invoice data. |
| Status | Pass |
| Remark | N/A |

Table 8.10 Testcase Table 10

| Test Case Id: | TC010 |
|---|---|
| Test Case Summary | Verify the functionality of managing company details, contact persons, terms and conditions, service notes, bank information, shift timing, and mail scheduling in the configuration section. |
| Prerequisite | The user must have appropriate access rights to manage the configuration settings. |
| Test Procedure | 1. Launch the application and navigate to the configuration section.<br>2. Click on the appropriate setting to be managed, such as company details, contact persons, terms and conditions, service notes, bank information, shift timing, or mail scheduling.<br>3. Update the settings as required.<br>4. Click on the "Save" button.<br>5. Verify that the settings are saved successfully. |
| Test Data | ● Updated company details<br>● Updated contact person details<br>● Updated terms and conditions<br>● Updated service notes<br>● Updated bank information<br>● Updated shift timing<br>● Updated mail scheduling settings |

| | |
|---|---|
| Expected Result | 1. The user should be able to access the configuration section.<br>2. The user should be able to manage the company details, contact persons, terms and conditions, service notes, bank information, shift timing, and mail scheduling settings.<br>3. The updated settings should be saved successfully. |
| Actual Result | 1. The user is able to access the configuration section.<br>2. The user is able to manage the company details, contact persons, terms and conditions, service notes, bank information, shift timing, and mail scheduling settings.<br>3. The updated settings are saved successfully. |
| Status | Pass |
| Remark | N/A |

Table 8.11 Testcase Table 11

| | |
|---|---|
| Test Case Id: | TC011 |
| Test Case Summary | Verify the functionality of adding a user, setting the user role, and setting the status in the user management section. |
| Prerequisite | The user must have appropriate access rights to add a user, set the user role, and set the status. |
| Test Procedure | 1. Launch the application and navigate to the user management section.<br>2. Click on the "Add User" button.<br>3. Enter the required user information in the respective fields such as username, email, and password. |

| | |
|---|---|
| | 4. Select the appropriate user role such as customer, admin, manager, or technician.<br>5. Set the status of the user to either active or inactive.<br>6. Click on the "Save" button.<br>7. Verify that the user is added successfully with the correct user role and status. |
| Test Data | ● Valid user information<br>● Selected user role as customer<br>● Selected user role as admin<br>● Selected user role as manager<br>● Selected user role as technician<br>● Set user status as active<br>● Set user status as inactive |
| Expected Result | 1. The user should be able to access the user management section.<br>2. The user should be able to add a user, set the user role, and set the status.<br>3. The user should be able to add the user successfully with the correct user role and status. |
| Actual Result | 1. The user is able to access the user management section.<br>2. The user is able to add a user, set the user role, and set the status.<br>3. The user is able to add the user successfully with the correct user role and status. |
| Status | Pass |
| Remark | N/A |

Table 8.12 Testcase Table 12

| | |
|---|---|
| Test Case Id: | TC012 |
| Test Case Summary | Verify the functionality of displaying user information in |

| | |
|---|---|
| | the user tracker section. |
| Prerequisite | The user must have appropriate access rights to view user information in the user tracker section. |
| Test Procedure | 1. Launch the application and navigate to the user tracker section.<br>2. Verify that the section displays the user name, IP address, current page path, and time of the user.<br>3. Verify that the information displayed is accurate and up-to-date.<br>4. Perform a user action, such as navigating to a different page or performing an operation in the application.<br>5. Verify that the user information is updated to reflect the current action. |
| Test Data | ● User information displayed in the user tracker section<br>● User navigating to a different page or performing an operation in the application |
| Expected Result | 1. The user should be able to access the user tracker section.<br>2. The user should be able to view the user name, IP address, current page path, and time of the user in the section.<br>3. The information displayed should be accurate and up-to-date.<br>4. The user information should be updated to reflect the current action. |
| Actual Result | 1. The user is able to access the user tracker section.<br>2. The user is able to view the user name, IP address, current page path, and time of the user in the section.<br>3. The information displayed is accurate and up-to-date.<br>4. The user information is updated to reflect the current action. |

| | |
|---|---|
| Status | Pass |
| Remark | N/A |

# 9

## Chapter # 9: LIMITATION AND FUTURE ENHANCEMENT

# 9. *Limitation and Future Enhancement*

## *9.1. Limitations*

Technical limitations: The system's performance may be limited by the technical capabilities of the hardware and software used to develop and run it.

Integration limitations: The system may face challenges when integrating with other tools and software used in the NDT inspection process.

Data privacy and security limitations: The system's security measures may not be foolproof, leading to potential data breaches or other privacy violations.

## *9.2. Future Enhancement*

Integration with emerging technologies: The system could be integrated with emerging technologies such as artificial intelligence and machine learning to enhance its data analytics capabilities.

Mobile Application: The system could have a mobile application that allows technicians to collect inspection data and access reports on the go, providing more flexibility and efficiency.

Cloud-based solution: The system could be migrated to a cloud-based solution to improve scalability and reduce the burden of maintaining hardware and software.

Real-time analytics: The system could be enhanced to provide real-time data analytics and alerts to managers and clients, allowing for more timely decision-making.

Enhanced security: The system could be improved with more advanced security features, such as two-factor authentication and encryption, to enhance data privacy and security.

# 10

**Chapter # 10: CONCLUSION**

## 10. Conclusion

In conclusion, the NDT Report System is a comprehensive solution for managing non-destructive testing inspections and generating accurate and compliant reports. The system's features include user management, inspection scheduling, data collection, report generation and review, report access, dashboard, notifications, data analytics, and integration with other software and tools.

The system provides significant benefits to its users, including improved efficiency, accuracy, and accessibility in the NDT inspection and reporting process. The system's scalability, user-friendly interface, and compliance with industry standards make it a valuable tool for NDT professionals.

However, the system may face limitations such as technical limitations, integration challenges, and data privacy and security limitations. To mitigate these limitations and improve the system's functionality, future enhancements such as integration with emerging technologies, mobile applications, cloud-based solutions, real-time analytics, and enhanced security can be considered.

Overall, the NDT Report System provides a comprehensive and reliable solution for managing NDT inspections and generating compliant reports, and has the potential to revolutionize the NDT industry with its features and benefits

## 11. Bibliography

https://legacy.reactjs.org/docs/

https://www.mongodb.com/

https://nodejs.org/en

https://expressjs.com/en/starter/

https://chat.openai.com/

https://online.visual-paradigm.com/

https://dbdiagram.io/d