Industrial Internship Report

TECH ELECON PVT LTD

Submitted by

PATEL DEV PRADIPKUMAR

12102040701027

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

Department of Computer Engineering

MADHUBEN & BHANUBHAI PATEL INSTITUTE OF TECHNOLOGY

The Charutar Vidya Mandal (CVM) University,

Vallabh Vidyanagar – 388120

May,2025

CERTIFICATE

This is to certify that **PATEL DEV PRADIPKUMAR (12102040701027)** has submitted the Industrial Internship report based on internship undergone at **Tech Elecon** for a period of **16** weeks from **1st Jan** to **30th April** in partial fulfilment for the degree of Bachelor of Engineering in **Computer Engineering, Madhuben and Bhanubhai Patel Institute of Technology** at The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar during the academic year 2024 – 25.

Pro. Nirali Pandya   Dr. Gopi Bhatt

Internal Guide   Head of the Department

DECLARATION

I, Patel Dev Pradipkumar (12102040701027), hereby declare that the Industrial Internship report submitted in partial fulfillment for the degree of Bachelor of Technology in CE , Madhuben and Bhanubhai Patel Institute of Technology, The Charutar Vidya Mandal (CVM) University, Vallabh Vidyanagar, is a Bonafide record of work carried out by me at Tech Elecon Pvt. Ltd. under the supervision of Satyam Raval 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

  Patel Dev Pradipkumar

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Tech Elecon Pvt. Ltd. for providing me with the opportunity to undertake my industrial internship at their esteemed organization. This experience has been invaluable for my professional growth and has significantly enhanced my understanding of real-world software development practices.

My special thanks to Mr. Satyam Raval, Deputy General Manager, for his guidance and support throughout my internship period.

I would also like to express my appreciation to Pro. Nirali Pandya, my internal guide from Madhuben and Bhanubhai Patel Institute of technology for the constant guidance, suggestions, and encouragement which helped me to complete this project successfully.

I am grateful to Dr. Gopi Bhatt, Head of the Department of Information Technology, for providing the necessary infrastructure and resources required for the completion of my internship project.

Finally, I would like to thank my family and friends for their unwavering support and encouragement throughout my academic journey.

PATEL DEV PRADIPKUMAR
 12102040701027

ABSTRACT

This industrial internship report outlines the development of a **Quotation Management System** using the **MERN stack (MongoDB, Express.js, React.js, Node.js)** along with **JWT-based authentication** and **role-based access control**. The internship was completed at **Tech Elecon Pvt. Ltd.**, a subsidiary of the Elecon Group with over 25 years of experience in delivering comprehensive IT solutions.

The Quotation Management System is designed to streamline and digitize the company's quotation and project workflow, ensuring transparency, efficiency, and better coordination between clients, employees, and administrators. The platform features **three distinct dashboards**—Client, Employee, and Admin—each optimized for specific roles while maintaining a seamless user experience.

The **Client Dashboard** allows customers to register, submit RFQs (Request for Quotations), view quotation history, fill out Purchase Order (PO) forms, track invoices, and monitor the status of deliverables. The **Employee Dashboard** enables team members to view RFQs and generate quotations through an integrated form. The **Admin Dashboard** provides tools for overseeing RFQs, quotations, and purchase orders, offering a comprehensive view of the system's operations.

The system follows modern software development principles, incorporating **responsive design**, **secure authentication mechanisms**, and **real-time status updates**. It also adheres to the **Software Development Life Cycle (SDLC)**, covering the entire process from requirement gathering and system design to development, testing, and deployment.

This report presents a detailed account of the development process, including key technical challenges, implemented solutions, and recommendations for future improvements to enhance functionality and scalability.

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
QMS	Quotation Management System
CSS	Cascading Style Sheets
DGM	Deputy General Manager
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JWT	JSON Web Token
MERN	MongoDB, Express.js, React.js, Node.js
DB	Database
RFQ	Request for Quotation
REST	Representational State Transfer
UI	User Interface
UX	User Experience
RBA	Role based Authentication

Table 1 : List of Abbreviations

Table of Content

CHAPTER 1 OVERVIEW OF THE COMPANY

Company Profile

Tech Elecon Pvt. Ltd. is the It division of the Elecon group of companies and has more than 25 years of experience in the fields of hardware, software, and networking solutions.

It is situated in the heart of Vithal Udyognagar Industrial Estate and in the proximity of the educational town of Vallabh Vidyanagar.

Tech Elecon is all set to reach new heights in the field of IT solutions. Tech Elecon is ready with all sorts of solutions and delivers any application that is web based and further our solutions are designed to adapt your business rather than your business adapting the software. Their solutions are 100% fruitful and empower you to take control of client's business online and in real time.

Tech Elecon have more than 100 employees with specialized skills in software development, custom software development, and e-commerce software development using custom software programming including .NET, C#.NET, PHP, and Open Source and Oracle.

Tech Elecon delivers quality products and services with a focus on integrating the same with existing technologies, providing the required automation to our customers to help them achieve their business objectives.

Mr. Nilesh Naik, the company's Vice President, is at the helm of the Tech Elecon organization. Mr. Satyam Raval, as Deputy General Manager, and after that, Manager and Associate Manager positions are listed. At the bottom, there are trainees at entry level, who follow up to engineer, senior engineer, also executive and senior executive manager.

1.1 Different Service's of the Company

Tech Elecon has extensive experience in providing IT services and has successfully adapted to technological advancements, making it the leading IT infrastructure management service provider in the region.

Our cutting edge delivery model covers all the stages of the solution lifecycle, including planning, deploying, managing, maintaining, auditing, upgrading, and improving Tech Elecon recognize that each client has unique needs and expectations when it comes to infrastructure and service providers. Our clients have the flexibility to choose from a wide range of IT infrastructure management and performance services based on their specific requirements. They can opt for on-site services on hybrid solutions that include on-site troubleshooting and support services.

Tech Elecon provides various services for business:

- Hardware maintenance and repairing
- Service desk management
- Desktop management
- Network management
- Messaging administrator
- Back-up management

Other services:

- " Software Development Services
- " Software Licensing
- Microsoft Product Implementation
- Linux Servers/ Desktop Implementation

CHAPTER 2 OVERVIEW OF DIFFERENT DEPARTMENTS

2.1Work Carried Out in Each Department

Tech Elecon Pvt. Ltd. is the IT division of the Elecon group, renowned for its diverse and extensive experience in hardware, software, and networking solutions. The company operates from its headquarters situated in the heart of Vithal Udyognagar Industrial Estate, close to the educational town of Vallabh Vidyanagar. Tech Elecon's workforce comprises over 100 employees, with specialized skills in software development, custom software development, and e-commerce solutions using technologies like .NET, C#.NET, PHP, and open-source frameworks.

Each department within the organization plays a vital role in delivering quality IT solutions:

- **Software Development Department:** This department focuses on custom software development, creating tailored solutions for various client needs. It employs technologies such as .NET, C#.NET, and PHP.

- **Networking and Infrastructure Management Department:** Responsible for the design, installation, and maintenance of client networks, ensuring seamless connectivity and efficient data flow.

- **Hardware Maintenance Department:** Handles the repair and maintenance of hardware components, ensuring systems run smoothly and without interruption.

- **Service Desk Management:** Provides ongoing technical support and troubleshooting services to clients, ensuring quick resolution of issues.

- **Messaging and Backup Management:** Manages communication systems and backup services for clients, ensuring data security and business continuity.

- **E-commerce and Web Application Development:** Focuses on developing web-based applications that are tailored to meet specific business requirements and provide real-time control to clients.

3

2.2 Technical Specifications of Major Equipment Used in Each Department

1. **Software Development Department:**

   o **Systems:** High-performance servers and workstations for coding, testing, and deployment.

   o **Development Tools**: Visual Studio (for .NET, C# development), PHPStorm, Oracle, and other open-source tools.

2. **Networking and Infrastructure Management:**

   o **Network Devices:** Routers, switches, firewalls, and wireless access points.

   o **Monitoring Tools:** Network monitoring software to track performance, uptime, and security.

3. **Hardware Maintenance Department:**

   o **Diagnostic Equipment:** Multimeters, oscilloscopes, and thermal cameras for assessing hardware performance and diagnosing issues.

   o **Repair Tools:** Soldering stations, replacement parts, and specialized kits for hardware repair.

4. **Service Desk Management:**

   o **Support Software:** Helpdesk management tools such as ServiceNow and Zendesk.

   o **Communication Tools:** VoIP systems and ticketing systems for managing service requests.

4

5. **E-commerce and Web Application Development**

   o **Servers:** Web servers and application servers for hosting client applications.

   o **Database Management Systems:** Oracle, MySQL, and SQL Server for data storage and management.

2.3 Schematic Layout of Operations for Manufacturing of End Product

The production and development process at Tech Elecon is closely linked to the design, development, and maintenance of IT solutions. The schematic layout follows these key stages:

1. **Client Requirement Gathering:** Meetings with clients to understand business needs.

2. **System Design and Architecture:** Designing the infrastructure and software solutions.

3. **Development and Implementation:** Writing the code, configuring systems, and deploying solutions.

4. **Testing and Quality Assurance:** Comprehensive testing to ensure the product works flawlessly.

5. **Deployment and Maintenance:** Deploying the system in the client's environment and providing ongoing support.

The process flows from one department to the next as each stage is completed. Software and network engineers collaborate, while service desk teams ensure proper management of any issues that arise during deployment.

2.4 Explanation of Each Stage of Production

1. **Client Requirement Gathering:** The first step involves understanding the client's specific requirements. This is achieved through consultations and collaboration with client representatives to ensure the solution will meet their business objectives.**System Design and Architecture:** Based on the client's requirements, a design is created, which includes network architecture, server configurations, and software specifications. This phase ensures that the solution is scalable, secure, and functional.

2. **Development and Implementation:** During this stage, the development team uses tools like .NET, C#, PHP, and other technologies to create the necessary software. Simultaneously, the infrastructure team sets up the required hardware and network components.

3. **Testing and Quality Assurance:** Once the system is developed, it undergoes a rigorous testing process, which includes functional testing, performance testing, and security audits. Any bugs or issues found are fixed before moving to the next stage.

4. **Deployment and Maintenance:** After successful testing, the solution is deployed in the client's environment. Post-deployment, the service desk team provides technical support, maintenance, and troubleshooting as needed to ensure continuous operation.

CHAPTER 3: INTRODUCTION TO INTERNSHIP AND PROJECT

3.1 INTERNSHIP SUMMARY

Internship is part of our curriculum. I have joined TEPL as Intern, and my major role is to contribute to the development of the Quotation Management System. It was in offline mode. My training phase and project work were conducted in offline mode. **Project Title: Quotation Management System.**

This project report details the design, development, and implementation of a web-based Quotation Management System, aimed at automating the quotation, purchase order (PO), and invoice management process. The system provides role-based access, with specific dashboards for clients, admins, and employees, and allows seamless interactions such as quotation requests, PO creation, and invoice generation. Developed by a multidisciplinary team, the project combines frontend and backend technologies to deliver a user-friendly, secure, and scalable solution. The frontend ensures an intuitive interface for easy navigation, while the backend securely handles data management. The system's features, such as client management, quotation history, and automated invoice generation, contribute to smoother business operations.

3.2 PURPOSE

The Quotation Management System is designed to streamline the process of quotation handling within an organization by automating the creation, management, and tracking of quotations, purchase orders, and invoices. Traditionally cumbersome manual procedures are replaced with a seamless, efficient online platform, reducing administrative overhead and enabling quicker response times. This digital approach enhances accuracy, security, and efficiency throughout the quotation process, providing clients and admins with real-time access to critical data.

The system also improves accountability by maintaining comprehensive digital records for each transaction. By providing automated notifications, secure login, and role-based access, it enhances operational efficiency and allows for better decision-making.

3.3 OBJECTIVE

1. **Efficient and Rapid Quotation Processing:** Develop a user-friendly, web-based application that simplifies the quotation process, from request submission to quotation approval, ensuring quick and accurate responses.

2. **Real-Time Tracking of Quotations and Purchase Orders:** Implement a robust tracking system that allows admins and clients to monitor the status of their quotations and purchase orders in real time.

3. **Automated Invoice Generation:** Enable the system to automatically generate invoices upon PO approval, ensuring quick, error-free billing.

4. **Role-Based Access Control:** Design the system to allow different levels of access for clients, admins, and employees, ensuring proper data segregation and enhanced security.

5. **Enhanced Security with Authentication:** Implement secure login features for clients, admins, and employees using JWT-based authentication.

6. **Dynamic Reporting and Analytics:** Implement a reporting system that provides insights into quotation history, PO statuses, and outstanding invoices.

7. **User-Friendly Interface:** Ensure the system is intuitive, minimizing the learning curve for clients and employees alike.

8. **Data Privacy and Security Compliance:** Apply best practices to ensure sensitive client and business data is stored securely, adhering to relevant privacy regulations.

3.4 SCOPE

What the System Can Do:

- **Comprehensive Quotation Management:** The system allows clients to request quotations and admins to generate and manage these quotations, including details like product names, quantities, and prices.

- **Role-Based Dashboards:** The system includes separate dashboards for admins, employees, and clients. Admins can manage users, quotations, POs, and invoices, while employees can only create quotations and view requests. Clients can view and submit RFQs, POs, and invoices.

- **Automated Notifications:** Automated email notifications are sent to clients and employees upon submission of a quotation request, PO approval, or invoice generation.

- **Invoice Generation and Management:** Upon approval of a PO, the system automatically generates an invoice, which can be downloaded by the client or viewed by the admin.

- **Quotation History and Tracking:** Admins and clients can view the history of all quotations and POs, including statuses (pending, approved, completed), allowing for effective tracking.

What the System Cannot Do (Current Limitations):

- **Advanced Financial Reporting:** The current system does not support advanced financial analytics, such as profit margins or detailed cost analysis.

- **Integration with External ERP Systems:** The system does not currently integrate with other enterprise resource planning (ERP) systems for data synchronization.

10

- **Real-Time Collaboration:** While the system facilitates document management and approval workflows, it does not support live collaboration or chat features.

3.5 TOOLS AND TECHNOLOGY

1. **React.js:** Frontend library for building the user interface.

2. **Redux:** State management library for handling complex state within the application.

3. **React Router:** A library for managing routing within React applications.

4. **Tailwind CSS:** A utility-first CSS framework used for modern and responsive web design.

5. **Node.js:** Backend runtime environment used for building server-side applications.

6. **Express.js:** Web application framework for Node.js used for building APIs.

7. **MongoDB:** NoSQL database for storing system data, including user, quotation, PO, and invoice records.

8. **Mongoose:** ODM (Object Data Modeling) library for MongoDB, used to interact with the database.

9. **JWT Authentication:** For secure login and role-based access control.

10. **Axios or Fetch API:** For making HTTP requests from the frontend to the backend API.

11. **Vite:** A modern build tool for faster development in React applications.

12. **Prettier**: Tools for maintaining code quality and consistent formatting.

13. Git and GitHub: Version control tools for tracking changes in the codebase.

14. VS Code: IDE for writing, testing, and debugging code.

15. Google Chrome: Used for testing and debugging the frontend application

3.6 Project Planning and Management

3.6.1 Development Approach and Justification

The project followed an Agile development approach, allowing iterative progress with continuous feedback and improvements. Agile was chosen due to its flexibility, responsiveness to change, and its focus on incremental development, which suited the modular structure of the Quotation Management System (QMS). Each core module—RFQ, Quotation, PO, Invoice, and Deliverables—was developed in sprints and reviewed periodically.

3.6.2 Time and Cost Estimation

The project was executed over a 4-month internship period. Time allocation was distributed as follows:

- **Requirement Analysis:** 2 weeks

- **Design:** 2 weeks

- **Development (Frontend + Backend):** 8 weeks

- **Testing & Bug Fixing:** 2 weeks

- **Documentation and Final Deployment:** 2 weeks

Cost estimation was minimal since the project was executed during an internship. Development was done using open-source technologies (React.js, Node.js, MongoDB, Tailwind CSS), making it cost-effective.

3.6.3 Roles and Responsibilities

As the sole intern handling the project:

- **Requirement Gathering –** Interacted with mentors to understand the business needs.

- **System Design –** Created the wireframes, database schema, and UI design.

- **Frontend Development –** Built using React.js and Tailwind CSS.

- **Backend Development –** Developed RESTful APIs using Node.js and Express.js.

- **Database Design –** Used MongoDB for storing client, RFQ, quotation, and invoice data.

- **Testing & Debugging –** Conducted manual testing and ensured system functionality.

- **Documentation –** Prepared reports, user guide, and developer documentation.

3.6.4 Group Dependencies

Although it was an individual project, periodic inputs and reviews from:

- **Company Mentor** – Helped align features with real business use-cases.

- **College Guide** – Provided academic validation and evaluation checkpoints.

3.7 Project Schedule (Gantt Chart).

Here's a sample Gantt chart breakdown for a 16-week timeline:

Week	Task
1-2	Requirement Gathering & Initial Research
3-4	UI/UX Design & Database Schema Design
5-6	Frontend Development (Dashboard, Forms)
7-8	Backend API Development (Auth, RFQ, Quotations)
9-10	PO & Invoice Module Integration
11	Deliverables and Approval System
12	Testing & Bug Fixing
13-14	Project Optimization and Final Review
15	Documentation & Report Writing
16	Final Submission and Presentation

CHAPTER 4 SYSTEM ANALYSIS

4.1 Study of Current System

Before the development of the Quotation Management System, most service-based companies followed a manual or semi-digital workflow involving emails, spreadsheets, and offline communication for handling RFQs, quotations, POs, and invoices. This led to delays, data duplication, lack of transparency, and difficulty in tracking the status of each request.

4.2 Problems and Weaknesses of Current System

- No centralized platform for managing quotation workflows.

- Difficulty in tracking status of RFQs, Quotations, POs, and Deliverables.

- Manual data entry caused errors and inefficiencies.

- Communication gaps between client, employee, and admin roles.

- No access control or role-based data visibility.

- Lack of proper documentation and digital record-keeping.

4.3 Requirements of New System

The new system must:

- Digitize the full quotation workflow.

- Enable different roles: Admin, Employee, and Client with proper access control.

- Allow clients to raise RFQs and track quotation history.

- Enable employees/admins to generate quotations and manage POs/invoices.

- Provide real-time status updates and notifications.

- Store all data securely and maintain proper logs.

4.4 System Feasibility

4.4.1 Contribution to Organizational Goals

The system supports service-based businesses by streamlining their quotation workflow, reducing turnaround time, improving customer experience, and enhancing operational efficiency.

4.4.2 Technical and Cost Feasibility

The system is feasible using open-source and industry-standard technologies like React.js, Node.js, MongoDB, and Tailwind CSS. Since these tools are free and widely supported, the solution is both technically viable and cost-effective.

4.4.3 Integration with Existing Systems

The system can be integrated with basic email services or extended in the future with CRM tools or payment gateways. Data is stored in MongoDB, which can be connected to analytics platforms for reporting.

4.5 Activity / Process in New System

- **Client:** Registers → Logs In → Submits RFQ → Views Quotation → Approves PO → Checks Deliverables → Receives Invoice.

- **Employee:** Views Assigned RFQs → Prepares and Submits Quotations.

- **Admin:** Manages users, oversees RFQ/PO/Invoice modules, monitors project flow.

4.6 Features of New System / Proposed System

- Role-based login (Admin, Employee, Client)

- RFQ form and history view for clients

- Quotation form for employees and tracking

- Purchase Order creation and approval

- Deliverable status tracking

- Invoice generation with client-specific details

- Secure authentication using JWT

- Responsive UI with Tailwind CSS

4.7 Main Modules / Components of Proposed System

- **Authentication Module** – JWT-based secure login/signup

- **RFQ Module** – Submit and track quotation requests

- **Quotation Module** – Create, view, and manage quotations

- **PO Module** – Purchase order form and approval flow

- **Deliverables Module** – Project output submission and approval

- **Invoice Module** – Generate and track invoices

- **Dashboard** – Summary and analytics for each role

4.8 Selection of Tools and Justification

Component	Selection	Justification
Frontend	React.js + Tailwind CSS	Fast, component-based UI with modern styling
Backend	Node.js + Express.js	Lightweight, efficient for REST API development
Database	MongoDB	NoSQL, flexible schema, scalable
Auth	JWT (JSON Web Token)	Secure and stateless authentication
Hosting	Render / Vercel / MongoDB Atlas	Easy cloud deployment, scalable and free tiers

17

CHAPTER 5 SYSTEM DESIGN

- ## System Design and Methodology

5..1 System Architecture
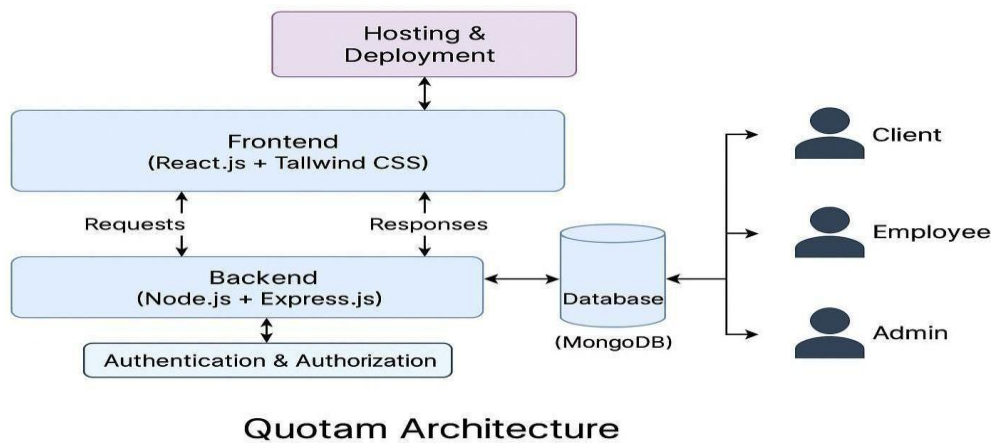
Fig 1 System Architecture

5..2 Use Case Diagram
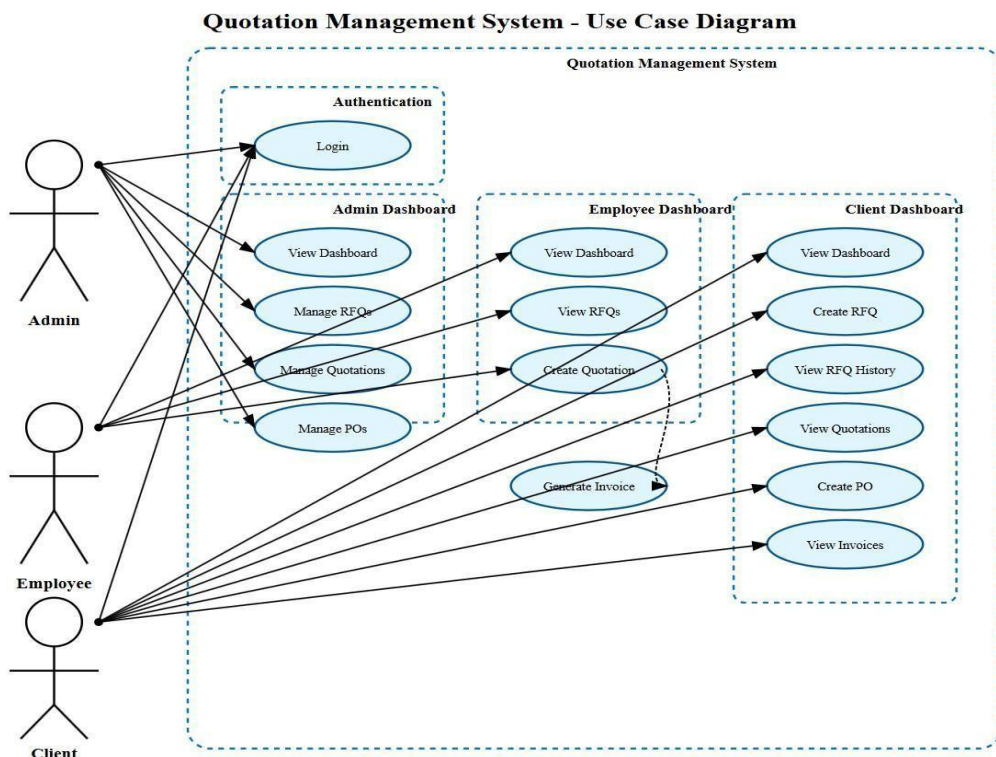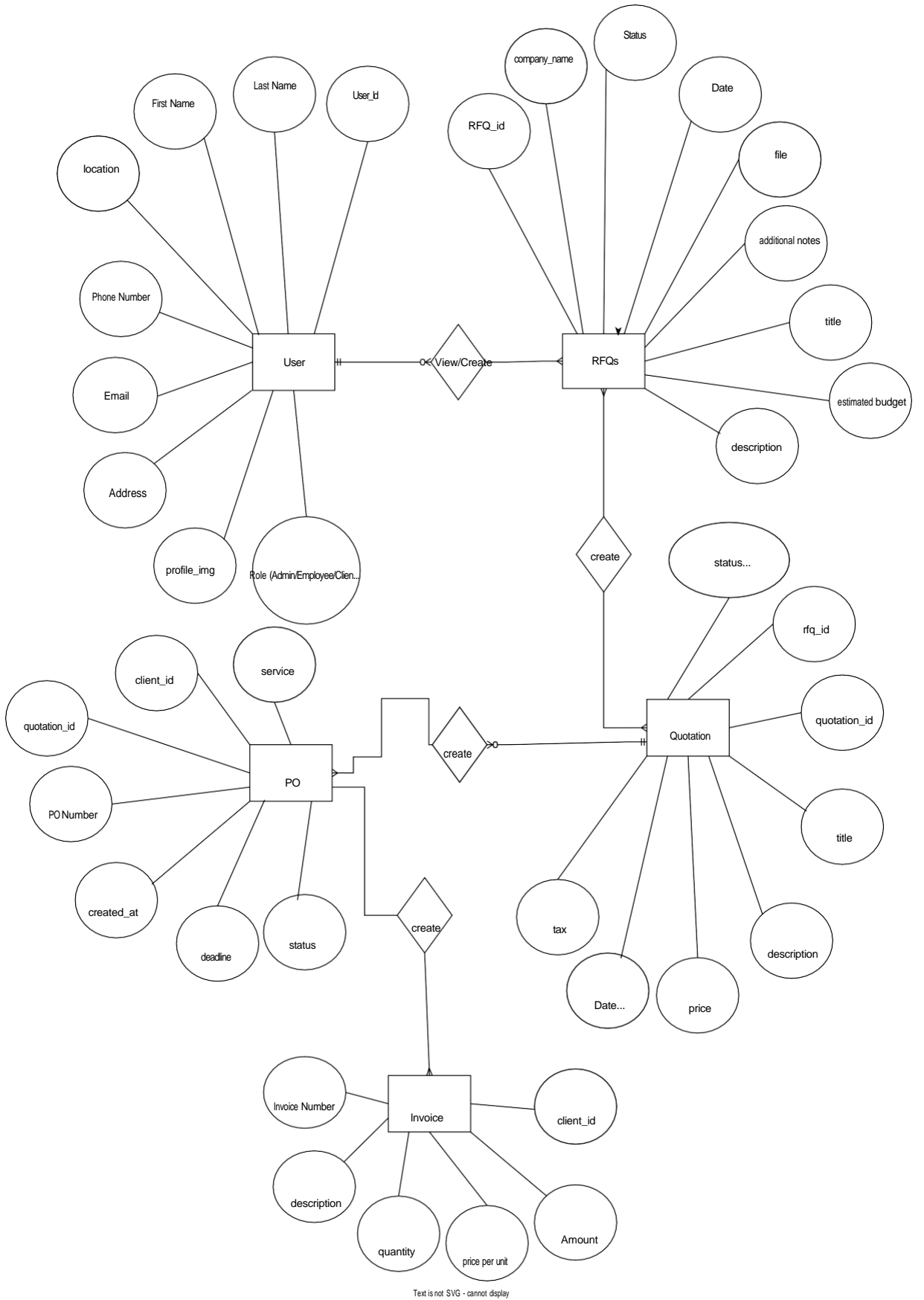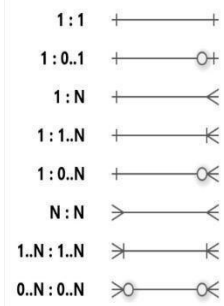
Fig 3 E-R Diaram

5.1.4 Sequence Diagram
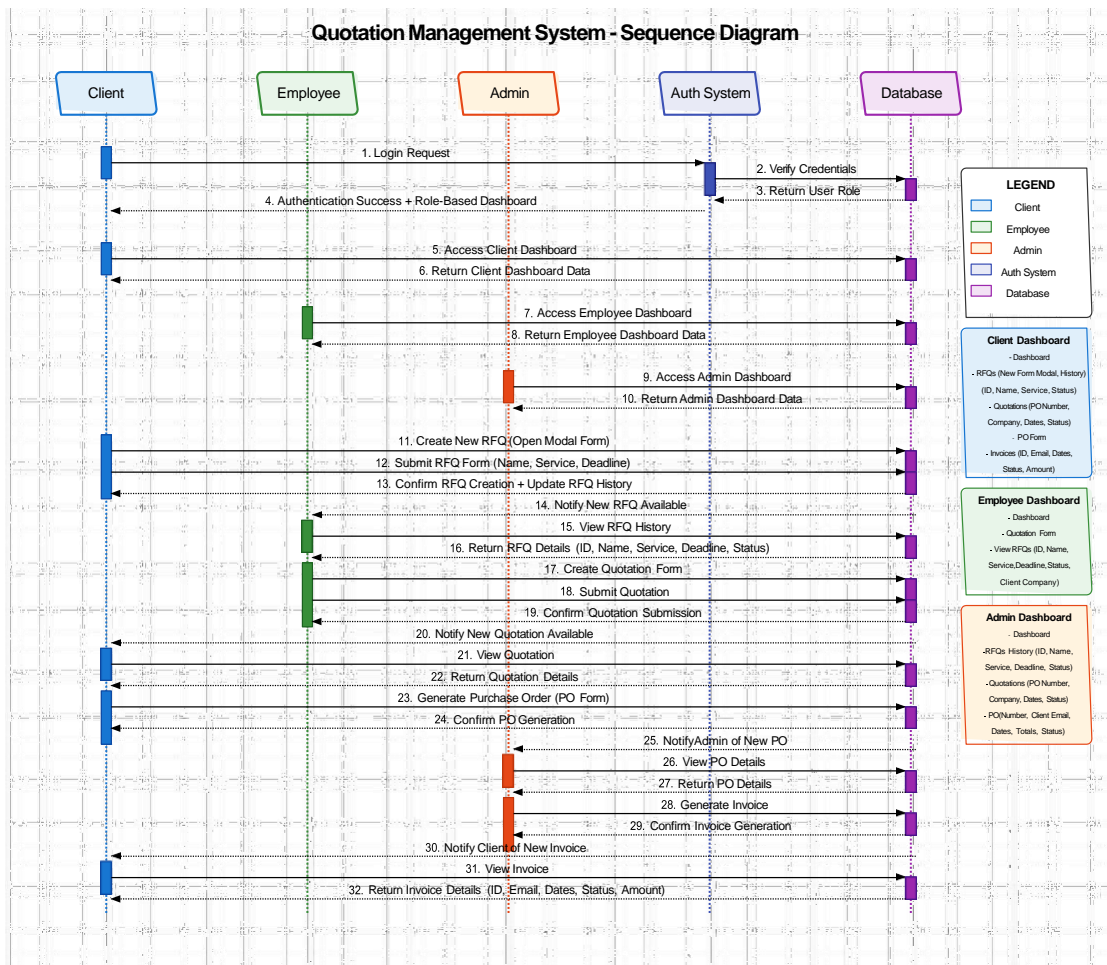
Fig 4 Sequence Diagram

The **Quotation Management System (QMS)** is designed following a **Modular and Layered Approach**, ensuring separation of responsibilities and easier future maintenance.

The development methodology used is a **Phased Incremental Development Model**. In this model, the project was divided into several logical phases (modules like Authentication, RFQ, Quotation, PO, Deliverables, and Invoicing), and each was completed and validated before moving to the next. This allowed continuous feedback, minimized risk, and provided an opportunity for early error detection.

The system is based on the **MVC (Model-View-Controller)** architecture:

- **Model**: Manages the data and business rules (MongoDB).

- **View**: Handles the presentation logic (React.js + TailwindCSS).

- **Controller**: Processes the incoming requests, manipulates the model, and updates the view (Node.js + Express.js).

Key Design Principles followed:

- **Separation of Concerns** (Frontend and Backend separated)

- **Scalability** (System can handle more users and data in future)

- **Security** (JWT Authentication, Role-Based Access Control)

- **User-Centric Design** (Simple, Clean and Intuitive UI)

- **API-First Architecture** (REST APIs used for frontend-backend communication)

• Database Design / Data Structure Design

The database chosen is **MongoDB**, a NoSQL database suited for flexible and scalable storage.
The database is designed using **collections** instead of traditional relational tables.

Major Collections and Key Fields:

Collection	Key Fields
Users	userId, name, email, passwordHash, role (admin/employee/client), createdAt
RFQs	rfqId, clientId (foreign key), productDetails, requirements, submittedDate, status (pending/approved/rejected)
Quotations	quotationId, rfqId (foreign key), employeeId, quoteDetails, price, deliveryDate, quotationStatus
POs	poId, quotationId (foreign key), clientId, poNumber, issueDate, approvalStatus
Deliverables	deliverableId, poId (foreign key), submittedFiles, clientFeedback, approvalStatus
Invoices	invoiceId, poId (foreign key), clientId, invoiceDate, amount, dueDate, paymentStatus

21

Relationships Between Collections:

- One **Client** can create multiple **RFQs**.

- One **RFQ** can have one **Quotation**.

- One **Quotation** can generate one **Purchase Order** (PO).

- One **PO** will have one **Deliverable** and one **Invoice**.

Database Storage is designed in such a way that

- Minimal redundancy is ensured.

- Faster retrieval of client records, quotation history, and invoice data.

5..1 Data Collection:

1. Users Collection (in MongoDB for

👤 **Users Table – 14 Fields**

Field Name	Data Type	Null	Key	Description
user_id	INT	NO	Primary	Unique ID for each user
role_id	INT	NO	Foreign	FK from `roles.role_id` – defines access level
full_name	VARCHAR(100)	NO		Full name of the user
email	VARCHAR(100)	NO	Unique	User's email address (login credential)
phone_number	VARCHAR(15)	YES		Contact number
username	VARCHAR(50)	YES	Unique	Optional custom username
password_hash	VARCHAR(255)	NO		Hashed password for authentication
profile_picture	VARCHAR(255)	YES		URL or path to profile image
status	ENUM('Active', 'Inactive', 'Banned')	NO		Current account status
last_login	DATETIME	YES		Timestamp of last login
login_attempts	INT	YES		Number of failed login attempts
email_verified	BOOLEAN	NO		Whether email is verified (`true` or `false`)
created_at	DATETIME	NO		Account creation timestamp
updated_at	DATETIME	YES		Last profile update timestamp

2.

2.RFQs (Request for Quotation) Collection (MongoDB Format)

4. RFQs (Request for Quotation)

Field Name	Data Type	Null	Key	Description
rfq_id	INT	NO	Primary	Unique ID for RFQ
client_id	INT	NO	Foreign	FK from `client_companies.company_id`
name	VARCHAR(100)	NO		RFQ title/name
service	VARCHAR(100)	NO		Type of service requested
description	TEXT	YES		Description/details of RFQ
deadline	DATE	NO		Submission deadline
reference_no	VARCHAR(50)	YES	Unique	Client's internal reference number
status	ENUM('Pending', 'Reviewed', 'Quoted', 'Rejected')	NO		Current RFQ status
created_by	INT	YES	Foreign	FK to user who created RFQ
created_at	DATETIME	NO		Timestamp of RFQ submission

3. Purchase Orders (PO) Collection (MongoDB Format)

Field Name	Data Type	Null	Key	Description
po_id	INT	NO	Primary	Unique PO ID
quotation_id	INT	NO	Foreign	FK from `quotations.quotation_id`
client_email	VARCHAR(100)	NO		Email of the client
billing_address	TEXT	YES		Billing details
shipping_address	TEXT	YES		Delivery address
po_date	DATE	NO		Date of PO
delivery_date	DATE	NO		Delivery expectation
total	DECIMAL(10,2)	NO		Order total
status	ENUM('Created', 'Approved', 'Shipped', 'Completed', 'Cancelled')	NO		PO status
remarks	TEXT	YES		Additional PO notes

4.Invoices Collection (MongoDB Format)

Field Name	Data Type	Null	Key	Description
invoice_id	INT	NO	Primary	Unique invoice ID
po_id	INT	NO	Foreign	FK from `purchase_orders.po_id`
client_email	VARCHAR(100)	NO		Client email
created_date	DATE	NO		Invoice generation date
due_date	DATE	NO		Invoice due date
payment_method	VARCHAR(50)	YES		e.g. Bank Transfer, Credit Card
status	ENUM('Unpaid', 'Paid', 'Overdue')	NO		Payment status
total_amount	DECIMAL(10,2)	NO		Total invoice amount
notes	TEXT	YES		Notes (e.g., tax info, late fees)

5.Product Catalog Collect

Field Name	Data Type	Null	Key	Description
product_id	INT	NO	Primary	Unique identifier for each product/service
product_name	VARCHAR(100)	NO		Name or title of the product or service
product_code	VARCHAR(50)	YES	Unique	Internal reference or SKU code
category	VARCHAR(50)	YES		Product category (e.g., Software, Hardware, Consulting)
description	TEXT	YES		Detailed description of the product/service
unit_price	DECIMAL(10,2)	NO		Price per unit/service
unit_of_measure	VARCHAR(20)	YES		Unit (e.g., pcs, hrs, package)
currency	VARCHAR(10)	YES		Currency (e.g., USD, INR)
availability	ENUM('In Stock', 'Out of Stock', 'On Request')	NO		Availability status
status	ENUM('Active', 'Inactive')	NO		Whether the product is available for selection
created_at	DATETIME	NO		When the product was added
updated_at	DATETIME	YES		Last updated timestamp
created_by	INT	YES	Foreign	FK from `users.user_id` – user who created the record

- **Input / Output and Interface Design**

5..1   User Interfaces

Client Dashboard:

- New RFQ Form Submission

- View RFQ Status

- View Quotations Received

- Submit PO Form after accepting a quotation

- View Deliverables

Employee Dashboard:

- View List of RFQs Assigned

- Submit Quotation Form

- Update Status of Quotations

Admin Dashboard:

- Manage User Accounts (Add / Remove Employees and Clients.

- View and Control All RFQs, Quotations, POs, Deliverables, and Invoices.

- Monitor Complete Workflow.

5..2 Forms, Reports, and Interface Samples

Form/Report	Description
RFQ Submission Form	Client fills in product/service details, quantity, delivery expectations.
Quotation Submission Form	Employee fills in pricing, estimated delivery date, terms & conditions.
PO Form	Client accepts quotation by filling purchase order details.
Invoice Report	Admin generates invoice linked to PO and sends it to client.

Screens are **responsive** across Desktop, Tablet, and Mobile using TailwindCSS utilities.

5..3 Access Control and Security

Security was prioritized at every level.

Access Mechanism:

- **JWT Token** issued after successful login.

- **Role-Based Authorization**:

    o Admin: Full Access

    o Employee: Limited to Quotations and assigned RFQs

    o Client: Limited to own RFQs, POs, Deliverables, and Invoices

Security Measures:

- Passwords encrypted using **bcrypt hashing**.

- Secure HTTPS communication.

- Backend routes protected with JWT middleware

Chapter 6: Implementation

6.1 Implementation Platform / Environment

The Quotation Management System (QMS) was developed and deployed using a modern full-stack JavaScript environment that supports scalability, security, and modular development. Below is the list of tools, platforms, and technologies used during implementation:

Component	Technology Used
Frontend	React.js with Tailwind CSS
Backend	Node.js with Express.js
Database	MongoDB (Cloud-based: MongoDB Atlas)
Authentication	JSON Web Token (JWT)
Hosting Platform	Render (Backend API) and Vercel (Frontend UI)
Version Control	Git and GitHub
API Testing	Postman
Project Management	Trello / GitHub Projects
IDE	Visual Studio Code

The system was implemented in a modular way, ensuring each component can be developed, tested, and debugged independently.

6.2 Modules and Functional Specifications

Client-Side Modules:

6.2.1 **Authentication Module**: Handles client sign-up/sign-in with secure JWT-based login.

6.2.2 **RFQ Module**: Allows clients to raise requests for quotations through a form.

6.2.3 **Quotation View**: Clients can view the list of quotations received for each RFQ.

6.2.4 **PO Module**: Submit Purchase Orders against approved quotations.

6.2.5 **Deliverables Module**: Review deliverables and request revisions if needed.

6.2.6 **Invoice Module**: View and download invoice PDF with status updates.

Employee-Side Modules:

6.2.7 **Login System**: Employee-specific login with role-based access.

6.2.8 **Quotation Generator**: Employees can fill and submit quotation forms linked to RFQs.

6.2.9 **RFQ Tracker**: Displays the list of RFQs assigned to employees for action.

Admin-Side Modules:

6.2.10 **Dashboard Overview**: See metrics and status of RFQs, Quotations, POs, Deliverables.

6.2.11 **Manage Clients/Employees**: Add, update, or remove users.

6.2.12 **Monitor Activity**: View logs of submissions and system events.

6.3 Findings / Results / Outcomes

6.3.1 A centralized platform was successfully developed to **streamline the RFQ-to- Invoice** workflow between service providers and clients.

6.3.2 The use of JWT and role-based access ensures **secured access control**.

6.3.3 Real-time updates and form validations enhance the **user experience and accuracy** of submitted data.

6.3.4 Admins can now **easily monitor** quotation requests and purchase cycles in a structured dashboard view.

6.3.5 The modularity of the system allows for **easy future upgrades**, such as adding chat support or real-time quotation tracking.

6.4 Result Analysis and Comparison

Criteria	Before QMS	After QMS Implementation
RFQ Submission	Manual (Email or Paper)	Online Form with Tracking
Quotation Management	Delayed, Unstructured	Digital, Filtered, Linked to RFQs
PO Process	Offline, prone to errors	Digital PO form, with client validation
Deliverables Feedback	Handled over mail/calls	System-based with approval workflow
Invoice Tracking	Spreadsheet-based	Auto-linked to PO and downloadable
Security & Access	Open access, insecure	JWT Auth + Role-based

The system has not only improved operational efficiency but also **reduced time, confusion, and manual dependencies** between stakeholders.

6.5 : UI Screenshots of the project

Landing Page:
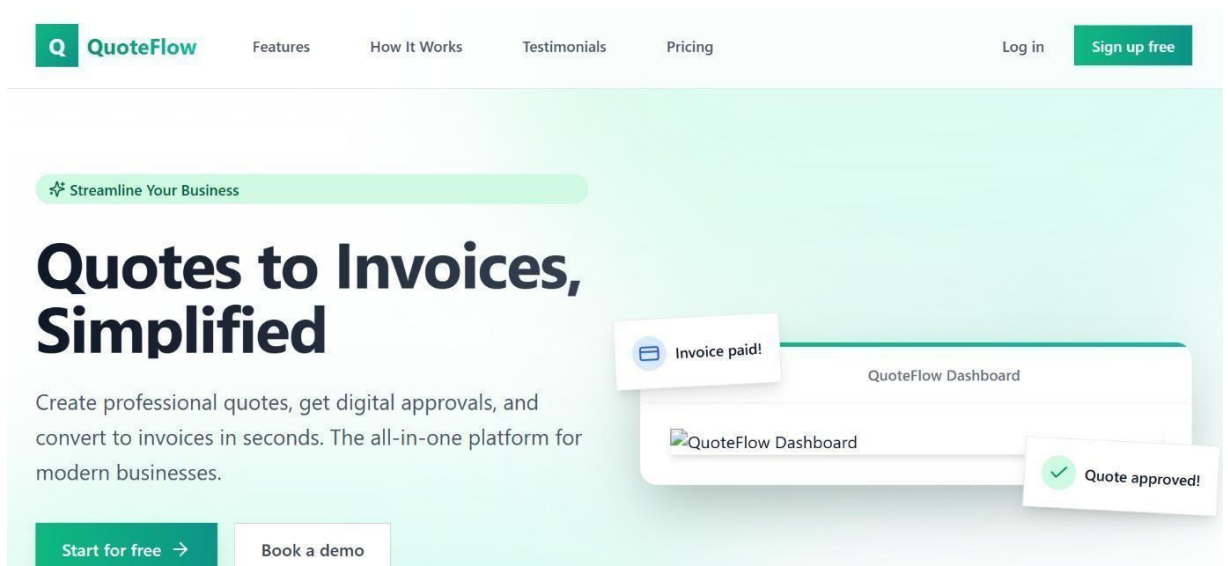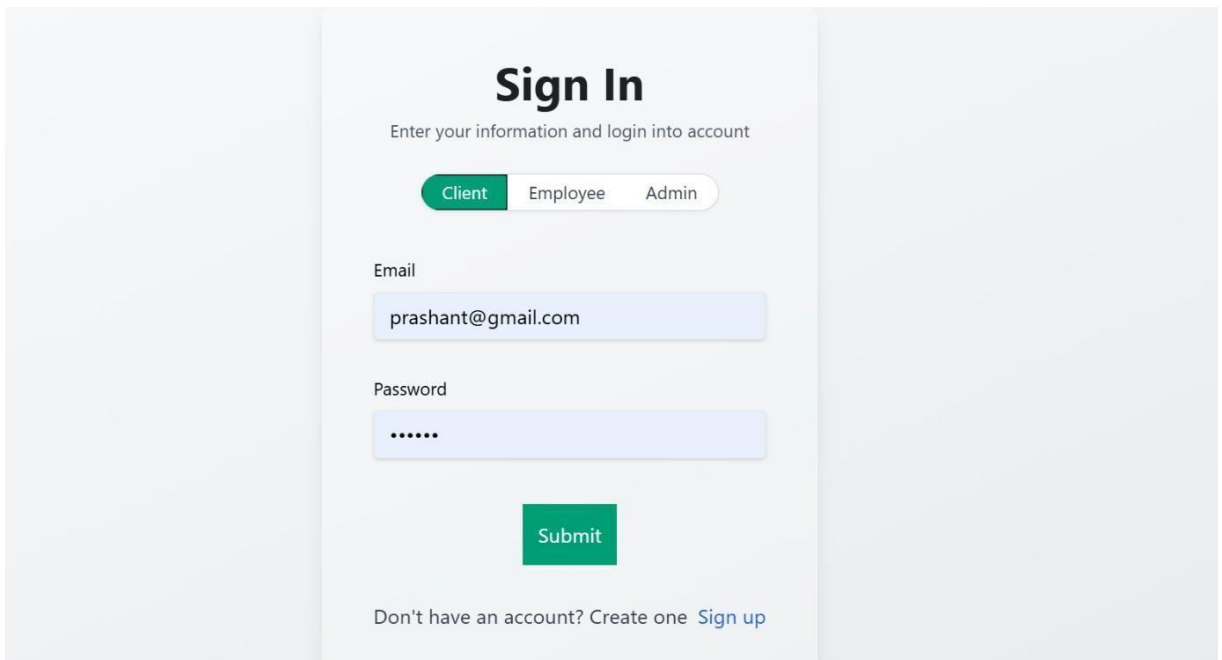
Figure 5: Landing Page

SIGN UP page:

Figure 6: Sign Up Page

SIGN IN PAGE:
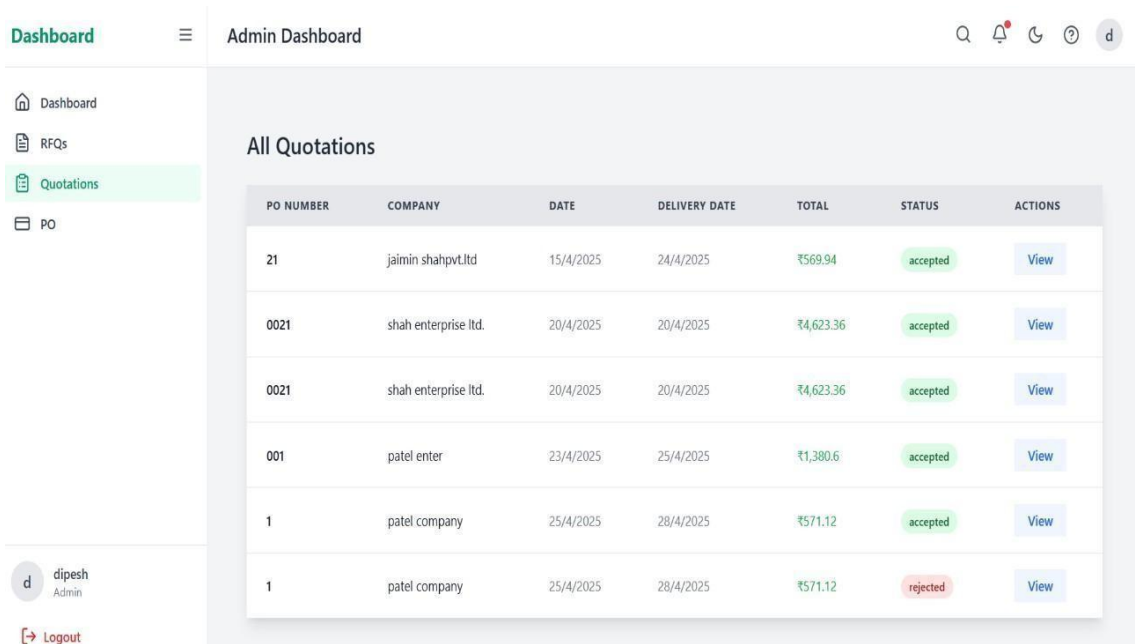
Figure 7: Sign In Page

ADMIN DASHBOARD:

Figure 8: ADMIN DASHBOARD

CUSTOMER DASHBOARD:

Figure 9: CUSTOMER DASHBOARD

EMPLOYEE DASHBOARD:

Figure 10: EMPLOYEE DASHBOARD

QUOTATION PAGE:

Figure 11: QUOTATION PAGE

PO PAGE:

Figure 12: PO PAGE

RFQ PAGE:

Figure 13: RFQ PAGE

DATABASE:
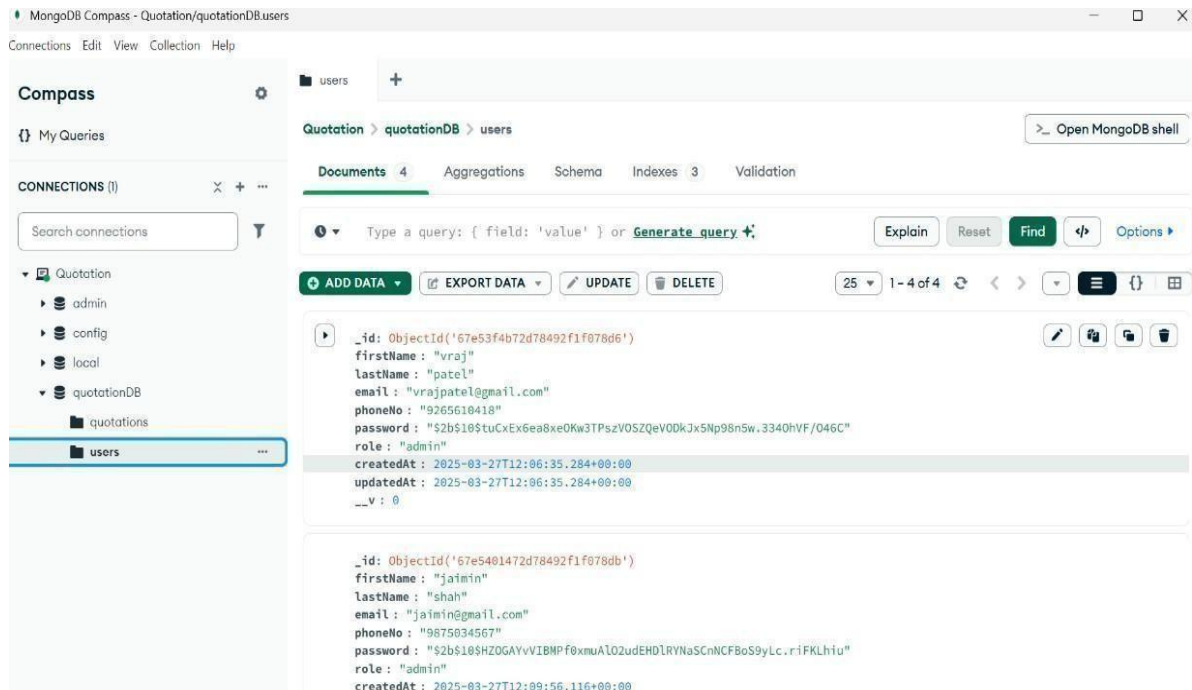
Figure 14: DATABASE

INVOICE PAGE:

Figure 15: INVOICE PAGE

Chapter 7: Testing

7.1 Testing Plan / Strategy

To ensure the reliability, accuracy, and security of the Quotation Management System (QMS), a well-structured testing strategy was followed. Both manual testing and unit testing were performed across all modules.

The primary testing objectives were:

- Validate functionality of each module (Client, Admin, Employee dashboards)

- Ensure secure and proper access based on roles using JWT

- Verify form validation, data integrity, and user interactions

- Test API responses, error handling, and database communication

Testing Types Used:

- Unit Testing – For individual functions/modules

- Integration Testing – For combining frontend with backend

- System Testing – End-to-end workflow from RFQ to invoice

- User Acceptance Testing (UAT) – Conducted to ensure real-world usage satisfaction

7.1 Test Results and Analysis

The following table summarizes key test cases executed during the development and deployment of the project:

7.1.1 Sample Test Cases

Test ID	Module	Test Condition	Expected Output	Actual Output	Remarks
TC_01	Client Login	Valid email & password	Dashboard access	Success	Pass
TC_02	Client Login	Invalid password	Error message–	Error shown	Pass
TC_03	RFQ Form	All fields filled	RFQ submitted	Submitted	Pass
TC_04	RFQ Form	Missing mandatory fields	Warning shown	Warning shown	Pass
TC_05	Admin Login	Role-based routing	Admin Dashboard	Dashboard Loaded	Pass
TC_06	Quotation Submission	Valid details	Quotation saved	Data saved	Pass
TC_07	PO Form	Valid PO data	PO created	Success	Pass
TC_08	Deliverables Review	Request revision	Deliverable updated	Update visible	Pass
TC_09	Invoice View	Click on Invoice	Invoice shown with PDF	Success	Pass
TC_10	Unauthorized Access	Try admin route without token	Redirect to login	Success	Pass

Table 2 : List of Test cases

7.2 Bug Tracking and Fixes

During development, issues like form validation failures, JWT expiration errors, and delayed API responses were detected and fixed. GitHub Issues and Trello were used for tracking and resolving bugs systematically.

7.3 Tools Used for Testing

- Postman – For API testing and response validation

- Chrome DevTools – UI and console error tracking

- MongoDB Compass – Database verification and integrity

- React Testing Library (Partial) – Component-level behavior testing

7.4 Summary

The system passed all critical test cases successfully. Minor UI issues were resolved during user acceptance testing. The project is now stable for real-world deployment with consistent and secure role-based operations.

CHAPTER 8: Conclusion and Discussion

8.1 Overall Analysis of Internship / Project Viability

The Quotation Management System (QMS) project proved to be both technically and functionally viable. It successfully digitizes the entire workflow of Request for Quotation (RFQ), Quotation generation, Purchase Order (PO) processing, Deliverables review, and Invoice management between clients, employees, and admin users. The use of a full-stack JavaScript framework (React.js, Node.js, and MongoDB) ensured a modern and scalable solution.

This system reduces manual effort, minimizes delays, and improves accuracy and transparency across the process.

8.2 Internship Progress Review with Industry Guide

Regular project review meetings were conducted with the industry mentor using platforms like Google Meet and Zoom. Weekly updates, progress tracking, and discussions on design and implementation were part of the internship workflow.

8.3 Dates of Continuous Evaluation (CE-I and CE-II)

- **CE-I Evaluation Date**: [7 march 2025]
- **CE-II Evaluation Date**: [19 april 2025]

Evaluations were based on project understanding, implementation progress, and mentor feedback.

8.4 Problems Encountered and Solutions

Problem Faced	Solution Implemented
JWT token expiry during session	Added token refresh logic and proper logout handling
Database schema mismatch	Introduced schema validation and structure standardization
UI responsiveness issues	Applied Tailwind utility classes and tested on multiple screen sizes

40

Role-based route issues	Added protected routes with role verification on frontend
Problem Faced	**Solution Implemented**
Form input validation errors	Used controlled form components and real-time validation

8.5 Summary of Internship / Project Work

Over the 4-month internship duration, the following milestones were achieved:

- Completed requirement analysis and system design

- Developed frontend (React.js) and backend (Node.js + Express)

- Integrated MongoDB Atlas for cloud database management

- Implemented secure login with JWT and role-based access

- Built complete RFQ to Invoice system with admin and employee dashboards

- Performed detailed testing and debugging

- Documented and presented the project with results and improvements

The internship not only improved technical knowledge but also enhanced project management, communication, and collaboration skills.

8.6 Limitations and Future Enhancements

Limitations:

- No real-time notifications or email alerts integrated.

- Limited analytics and reporting functionality.

- No mobile app version is developed.

- UI/UX can be further polished with animations or feedback messages.

41

REFERENCES

1. https://www.techelecon.com/
2. https://vite.dev/guide/
3. https://react-icons.github.io/react-icons/
4. https://cloud.google.com/docs
5. https://www.nodemailer.com/
6. https://github.com/
7. https://www.mongodb.com/docs
8. https://reactjs.org/docs/getting-started.html
9. https://tailwindcss.com/docs

CVM UNIVERSITY   MBIT