

Trace Register and Web Portal for Dental Clinics Management Solutions

SAI DINESHB
Roll No. 20PW30

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

**FIVE YEAR INTEGRATED
M.Sc. SOFTWARE SYSTEMS**

OF ANNA UNIVERSITY



NOVEMBER 2023

DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004.

PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)
COIMBATORE – 641 004.

Seventh Semester
Project Work

Trace Register and Web Portal for Dental Clinics Management Solutions

Bona fide record of work done by

SAI DINESHB
Roll No. 20PW30

Submitted in partial fulfilment of the requirements for the degree of

FIVE YEAR INTEGRATED
M.Sc. SOFTWARE SYSTEMS
of Anna University

NOVEMBER 2023

Academic Guide

Head of the Department

Submitted for the Viva-Voce Examination held on _____

Internal Examiner

External Examiner

CONTENTS

PARTICULARS	PAGE NO.
ACKNOWLEDGEMENT	i
SYNOPSIS	ii
1. INTRODUCTION	1
1.1 ORGANIZATION PROFILE	1
1.2 SYSTEM ENVIRONMENT	4
1.3 TOOLS AND TECHNOLOGIES USED	5
2. TRACE REGISTER	11
2.1 DASHBOARD FOR SUSTAINABILITY	11
2.2 DASHBOARD FOR DATA INTEGRATION	12
2.3 REFACTORING CODE OF BACKEND API	13
3. WEB PORTAL FOR DENTAL CLINICS MANAGEMENT	16
3.1 COMMON COMPONENTS DEVELOPMENT	16
3.2 STORYBOOK FOR EVERY COMPONENTS	20
3.3 UI FOR DIFFERENT TYPE OF USERS	28
3.4 INTEGRATION FOR DIFFERENT PORTALS	31
3.5 TESTING	33
4. CONCLUSION	34
BIBLIOGRAPHY	34

ACKNOWLEDGEMENT

I am extremely grateful to **Dr. K. Prakasan**, Principal, PSG College of Technology, for giving me this opportunity to do my project at **Foxsense Innovations Pvt Ltd., Coimbatore**.

I further extend my gratitude to **Dr. R. Nadarajan**, Director, PSG Institute of Applied Mathematics and Computational Sciences, for his continual support and encouragement.

I am deeply indebted to **Dr. Shina Sheen**, Professor and Head, Department of Applied Mathematics and Computational Sciences, PSG College of Technology, for her continual support and ardent motivation.

I extend my heartfelt gratitude to my internal guide **Dr. K. Mohan**, Assistant Professor (Sl. Gr), Department of Applied Mathematics and Computational Sciences, PSG College of Technology, for his continual support and enduring guidance throughout my project tenure.

I also express my sincere thanks to **Mr. Ramanuja Naidu Gorle**, Chief Executive Officer, **Foxsense Innovations Pvt Ltd.**... for providing me a great opportunity to pursue my internship at Foxsense Innovations Pvt Ltd.

I also express my sincere thanks to **Mr. Ramprasath Balakrishnan**, Chief Technical Officer, **Foxsense Innovations Pvt Ltd.** for helping me with suggestions in various aspects to do this project work.

I further extend my gratitude to **Mr. Lakshmi Tarun Kodali**, Project Manager, **Foxsense Innovations Pvt Ltd.**, at various phases of this project work.

I further extend my gratitude to my mentor **Mr. Praneeth Narasimha**, Senior Software Developer, **Foxsense Innovations Pvt Ltd.**, at various phases of this project work.

I wish to acknowledge all the other members of my team, who made work challenging and were very helpful in growing my learning curve.

Finally, I express my gratitude to all the faculty and staff of the Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore, and my family and friends for their encouragement and support.

SYNOPSIS

Foxsense Innovations was established in 2019 by three partners with a goal to bring innovative solutions for businesses to meet ongoing demands in the tech industry. The company ensure a standard of excellence and commitment in our services with a major focus on relationship with the clients. All the services that Foxsense Innovations provide to companies strictly adhere to enterprise- level security, and any mandatory compliance. The company provide all engineering capabilities to customers that enable them to create world-class products under one roof.

Trace Register, established in 2005, strategically addresses the information gap between fish catchers and consumers. By amalgamating expertise in the seafood industry and technology, the company offers a transformative solution. Serving as a pivotal tool, Trace Register enhances traceability, thereby revolutionizing supply chain management and business operations. With a commitment to transparency, it acts as a crucial bridge, allowing stakeholders to manage and improve their processes effectively in the dynamic seafood industry.

A unified web portal for dental clinic management presents an exceptional business solution, consolidating a range of critical activities within the centralized framework of a leading parent dental clinic. This comprehensive platform seamlessly integrates key management tasks, including the acquisition of other dental clinics, streamlined recruitment processes for multiple clinics, and facilitating job applications. By leveraging this sophisticated web portal, the parent dental clinic can efficiently navigate and execute strategic initiatives, fostering growth through the seamless management of diverse operations. From expanding the clinic network through acquisitions to managing personnel recruitment across various branches, this innovative solution ensures a cohesive and efficient approach to elevate the overarching success of the dental clinic enterprise.

CHAPTER 1

INTRODUCTION

A detailed description of the organization for which the applications are being developed is presented in this chapter. It also specifies the system environment used in the development and it gives a brief introduction to the tools and various terminologies used in the development of the applications.

1.1 ORGANIZATION PROFILE

The Foxsense Innovations was established in 2019 by three partners with a goal to bring innovative solutions for businesses to meet ongoing demands in the tech industry. The company ensure a standard of excellence and commitment in our services with a major focus on relationship with the clients.

All the services that Foxsense Innovations provide to companies strictly adhere to enterprise- level security, and any mandatory compliance. The company provide all engineering capabilities to customers that enable them to create world- class products under one roof.

1.1.1 Processes

DEMOS / DISCOVERY

A dedicated team collaborates with pre-sales to conduct stakeholder interviews, client meetings, and product demo sessions. With a focus on seamless coordination, this specialized group ensures a comprehensive understanding of client needs through insightful engagements. They play a crucial role in shaping successful client interactions, contributing to the effectiveness of pre-sales efforts. Through adept facilitation of product demo sessions, the team enhances the overall client experience, fostering clarity and confidence in presented solutions. Committed to aligning seamlessly with pre-sales functions, they fortify the foundation for successful client engagements.

DESIGN, DEVELOP & DEPLOY

A collaborative team comprising UX, front-end, and back-end experts, alongside architects, will orchestrate the end-to-end process of designing, developing, testing, and deploying the product. The team's collective proficiency will be harnessed to ensure a seamless

transition from concept to implementation. With a keen focus on user experience, the team aims to create an intuitive and efficient product. Utilizing their diverse skill sets, they will meticulously craft and refine the product's design and functionality. The final product will be rigorously tested to guarantee reliability and performance. Leveraging an optimized Cloud architecture, the team is poised to deliver a scalable and resilient solution that meets the highest standards of quality and efficiency.

SUPPORT

In addition to core services, the company provide comprehensive post-delivery technical support, ensuring a seamless experience for the clients. The commitment extends to offering tailored product training sessions, empowering users with in-depth knowledge and proficiency. With a focus on continuous improvement, our team specializes in DevOps practices, optimizing the product's operational efficiency. Beyond the initial release, Foxsense Innovations is dedicated to facilitating ongoing feature enhancements, adapting to evolving needs and market dynamics. This holistic approach underscores the commitment to long-term partnerships and client success, reinforcing the role as a reliable and adaptive technology partner.

1.1.2 Services Offered

COMPREHENSIVE TECHNOLOGY CONSULTING

Comprehensive Technology Consulting is the cornerstone of the project, providing strategic guidance, customized solutions, and expert advice for seamless digital transformation. The approach includes risk mitigation, performance optimization, and a commitment to continuous innovation. The company offers end-to-end solutions, ensuring the business stays ahead in the dynamic tech landscape.

WEB APPLICATION DEVELOPMENT

Web Applications Development is the bedrock of the company, encompassing custom solutions, strategic planning, and expert execution for a seamless online presence. The approach prioritizes user experience, scalability, and security, ensuring digital success. With a commitment to innovation, mitigate risks and optimize performance, delivering end-to-end solutions that keep the web presence at the forefront of technology.

MOBILE APPLICATION DEVELOPMENT

Mobile Applications Development forms the core of our company, emphasizing custom solutions, strategic planning, and expert execution for a seamless mobile presence. Our approach prioritizes user experience, scalability, and security, ensuring success in the dynamic mobile landscape. With a commitment to innovation, Foxsense Innovations mitigate risks and optimize performance, delivering end-to-end solutions that keep the mobile applications at the forefront of technology.

DATA SERVICES: MIGRATION | INTEGRATIONS | SCRAPING | ANALYTICS

Data Services, comprising Migration, Integrations, Scraping, and Analytics, is the cornerstone of our project. The company specialize in crafting custom solutions and strategic planning to seamlessly handle data across the ecosystem. Our approach ensures a smooth transition during migrations, seamless integrations, and valuable insights through advanced analytics. With a commitment to innovations, the company employ scraping techniques for data acquisition, enabling harness actionable intelligence. Our end-to-end solutions mitigate risks, optimize performance, and position the data operations at the forefront of technology.

1.2 SYSTEM ENVIRONMENT

The Windows environment has been used throughout the development. The hardware and software configurations used are given below.

1.2.1 Hardware Configurations

Processor	: Intel Core i5-1135G7 11 th Gen
Hard Disk	: 512GB
RAM	: 8GB
System Type	: 64-bit Operating System

1.2.2 Software Configurations

OS	: Windows 10
Languages	: Javascript, C#
IDE	: Visual Studio Code, Visual studio

1.3 TOOLS AND TECHNOLOGIES USED

The technology stack used to develop the tool is shown in Figure 1.1. The tools and technologies used in the project are briefly discussed in this section.



Figure 1.1 Technology Stack

1.3.1 Next.js Framework

Next.js is a powerful and versatile React framework renowned for its simplicity and efficiency in building robust web applications. Leveraging server-side rendering and automatic code splitting, it optimizes performance and enhances user experiences. The framework's seamless integration with React components, coupled with its support for both client and server-side rendering, makes it a preferred choice for scalable projects. With built-in routing and a strong emphasis on developer-friendly features, Next.js accelerates development cycles and facilitates easy deployment. Its extensibility allows for the incorporation of various plugins, ensuring flexibility in functionality. Overall, Next.js stands out as an intuitive, performance-driven solution for modern web development.

Key features and components of NEXTJS Framework:

Next.js, a React framework, comprises several key components that contribute to its functionality and flexibility:

Pages

Next.js organizes applications around the "pages" directory, where each file corresponds to a route, simplifying the creation of dynamic, SEO-friendly web pages.

File-based Routing:

The framework uses a file-based routing system, allowing developers to create routes by merely adding files to the "pages" directory, streamlining the development process.

Server-side Rendering (SSR):

Next.js supports SSR, enabling faster page loads by rendering pages on the server and sending HTML to the client, which is particularly beneficial for SEO and initial page performance.

Automatic Code Splitting:

Next.js automatically splits code into small chunks, delivering only the necessary JavaScript for each page. This feature enhances performance and facilitates efficient use of network resources.

API Routes

Next.js simplifies the creation of API endpoints with its dedicated API Routes, allowing developers to handle server-side logic within the same project.

Static Site Generation (SSG):

Alongside SSR, Next.js supports SSG, generating static HTML files during build time, which can be served directly from a CDN for improved performance and scalability.

CSS-in-JS Support:

Next.js seamlessly integrates with popular CSS-in-JS libraries like styled-components and Emotion, providing a convenient way to style React components.

1.3.2 NestJS

NestJS is a powerful and extensible Node.js framework that follows a modular architecture, promoting scalable and maintainable server-side applications. Leveraging TypeScript, NestJS enhances developer productivity and code maintainability by providing strong typing and object-oriented programming features. Its modular structure, inspired by Angular, encourages the use of decorators and dependency injection, facilitating code organization and testing. With built-in support for GraphQL and REST APIs, NestJS offers flexibility in designing APIs to meet diverse project requirements. The framework's emphasis on convention over configuration streamlines development, while its compatibility with various databases and middleware simplifies integration. Overall, NestJS stands out as a robust and developer-friendly framework for building enterprise-level, server-side applications in the Node.js ecosystem.

Key aspects of NESTJS:

Modular Architecture:

NestJS promotes a modular structure, allowing developers to organize code into cohesive and reusable modules. This modularity enhances scalability and maintainability.

TypeScript Support:

NestJS is built with TypeScript, providing strong typing, object-oriented features, and improved code maintainability. TypeScript brings static typing to JavaScript, making code more robust and less error-prone.

Decorators and Dependency Injection:

Inspired by Angular, NestJS utilizes decorators and dependency injection to streamline code organization and improve testability. This design pattern enhances code readability and maintainability.

Built-in Support for GraphQL and REST:

NestJS supports both GraphQL and REST APIs out of the box. This flexibility allows developers to choose the API paradigm that best suits their project requirements.

Convention over Configuration:

NestJS follows the principle of convention over configuration, reducing the need for manual setup. This makes development more intuitive and accelerates the coding process.

Compatibility and Integration:

NestJS is compatible with various databases, middleware, and external services, making it versatile and suitable for integration with different technologies.

1.3.3 Storybook

Storybook is an open-source tool widely used for developing UI components in isolation, providing a dedicated environment for component testing and showcasing. It supports multiple frameworks like React, Vue, and Angular, offering a versatile solution for diverse web development projects. With Storybook, developers can create interactive and documented component libraries, streamlining collaboration between design and development teams. Its hot-reloading feature allows for real-time updates during the development process, enhancing efficiency. Storybook's addon ecosystem further extends its functionality, enabling integration with various tools and services for a seamless workflow. Overall, Storybook serves as a valuable resource for building, testing, and documenting UI components across different web development frameworks.

KEY COMPONENTS OF STORYBOOK

Isolation of Components

Storybook allows developers to develop and test UI components in isolation, providing a dedicated environment for each component.

Support for Multiple Frameworks:

It supports various frontend frameworks such as React, Vue, and Angular, making it versatile and suitable for a range of projects.

Interactive Component Development:

Developers can create interactive stories for components, allowing for real-time development and testing, improving the efficiency of the development process.

Addon Ecosystem:

Storybook has a rich ecosystem of addons that extend its functionality. These addons enable integration with various tools and services, enhancing the overall development workflow.

Documentation:

Storybook serves as a documentation platform for UI components, making it easier for developers and designers to understand, test, and use components across projects.

Hot Reloading:

The hot-reloading feature allows for real-time updates as developers make changes to components, providing immediate feedback and speeding up the iteration process.

Collaboration:

Storybook promotes collaboration between design and development teams by providing a visual representation of components and their variations, fostering a shared understanding of the UI elements being developed.

1.3.4 PostgreSQL

PostgreSQL is an open-source, object-relational database management system (ORDBMS). It is known for its robustness, extensibility, and advanced features, making it a popular choice for a wide range of applications, from small projects to large, enterprise-level systems.

PostgreSQL comes with many features aimed to help developers build applications, administrators to protect data integrity and build fault-tolerant environments, and help us manage our data no matter how big or small the dataset. In addition to being free and open source, PostgreSQL is highly extensible.

CHAPTER 2

TRACE REGISTER

There was a vast information gap between the people who catch and process fish and those who buy and eat it. Traceability was the solution to fill that information gap and Trace Register was started in 2005, bringing together a team with a depth of experience in the seafood industry and technology. It is a critical tool for managing and improving supply chains and business operations.

2.1 Dashboard for Sustainability

The sustainability dashboard empowers seafood companies to ensure responsible sourcing from fisheries. Users can easily verify product compliance with set rules and update information, distinguishing between wild and farmed products. The intuitive interface features insightful pie charts and bar graphs illustrating the volume of products meeting sustainability criteria versus those that do not, enhancing transparency in the seafood supply chain.

2.1.1 Bar Graphs and Pie Charts

Bar graphs and pie chart are used to display the amount of the volume of the records of fishing products which have passed the rules and have failed the rules.

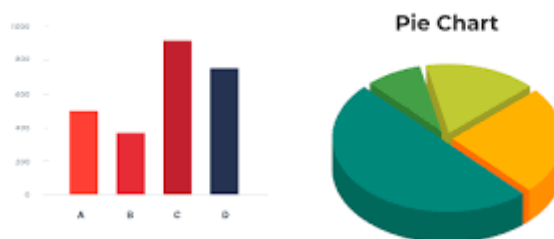


Figure 2.1 Bar Graphs and Pie Charts

2.1.2 CSV EXPORTS FOR DETAILED INFORMATION

Users will have an option to export their products which have failed the sustainability rules and in which category it has failed.

Figure 2.2 CSV exports

2.1.3 INBUILT TABLE FOR ANALYTICS

Apart from having an exclusive option for exporting as a CSV file, it has an inbuilt table to view the products which have failed the sustainability rules and in which category it has failed.

Figure 2.3 Table

2.1.4 OPTION FOR SHARING WITH OTHER PARTNERS

There is an option for sharing details of the sustainability rules of one's products to other traders for better cooperation between them.

2.1.5 OPTION TO VIEW ALL THE SUSTAINABILITY RULES

Within the dashboard, users have a dedicated option to access and review all sustainability rules governing seafood sourcing. This feature enables a comprehensive understanding of the criteria guiding responsible fishing practices. Users can navigate through a user-friendly interface to explore detailed information about each rule, ensuring transparency

and informed decision-making. This centralized repository fosters awareness and adherence to sustainability guidelines, promoting ethical choices in sourcing seafood products.

2.1.6 Filter Option

The dashboard facilitates seamless record management with an advanced filtering system. Users can easily sort through diverse records based on criteria such as sustainability rules, distinguishing between wild and farmed products, and categorizing by source and final products. This filtering functionality ensures a tailored view of data, enhancing efficiency in decision-making and allowing users to focus on specific aspects of their sustainability initiatives. The intuitive interface empowers users to swiftly navigate and analyze records, promoting transparency and responsible sourcing within the seafood industry.

2.2 DASHBOARD FOR DATA INTEGRATION

The DI dashboard provides users with real-time insights into the transaction status of key company files, including Bill of Material and Advanced Shipment Notice. Users can easily upload files locally and verify successful processing, with detailed reports available for any failed uploads. The dashboard offers analytics on missing fields, enhancing user understanding and facilitating collaboration with partners. A dynamic pie chart visually represents the absent information in uploaded files, providing a quick overview for informed decision-making in the transactional process.

2.2.1 BAR CHARTS AND PIE CHARTS

The DI dashboard features dynamic data visualization through impactful pie charts and bar graphs. These graphical representations offer users a quick and intuitive understanding of missing information in uploaded files. The pie chart vividly illustrates the distribution of absent data, providing a visual summary for efficient analysis. Additionally, bar graphs showcase analytics on fields lacking in uploaded files, empowering users to pinpoint specific areas for improvement. These visual elements not only enhance the dashboard's aesthetics but also serve as powerful tools for users to make informed decisions and optimize their transactional processes.

2.2.2 CSVEXPORT OPTION

Enhances the user to export their information of the uploaded files and processed files which are all lacking records and passed records.

2.3 REFACTORING CODE OF BACKEND API

The entire legacy code in native C# language has been completely refactored to accommodate the design principle of SOLID principles and enhances the code for better testing.

2.3.1 IMPLEMENTATION OF SOLID PRINCIPLES

The code base implementation integrates sophisticated design patterns such as the Strategy and Chain of Responsibility, enhancing Flexibility and Extensibility. By adhering to SOLID principles—ensuring Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, and Dependency Inversion—The development approach prioritizes maintainability and scalability. The Strategy pattern facilitates dynamic algorithm selection, enabling seamless adaptability to evolving requirements. Meanwhile, the Chain of Responsibility enhances code modularity by decoupling senders and receivers of requests. This meticulous incorporation of design patterns and adherence to SOLID principles not only fortify the software's architecture but also foster a foundation for agile development and long-term sustainability.

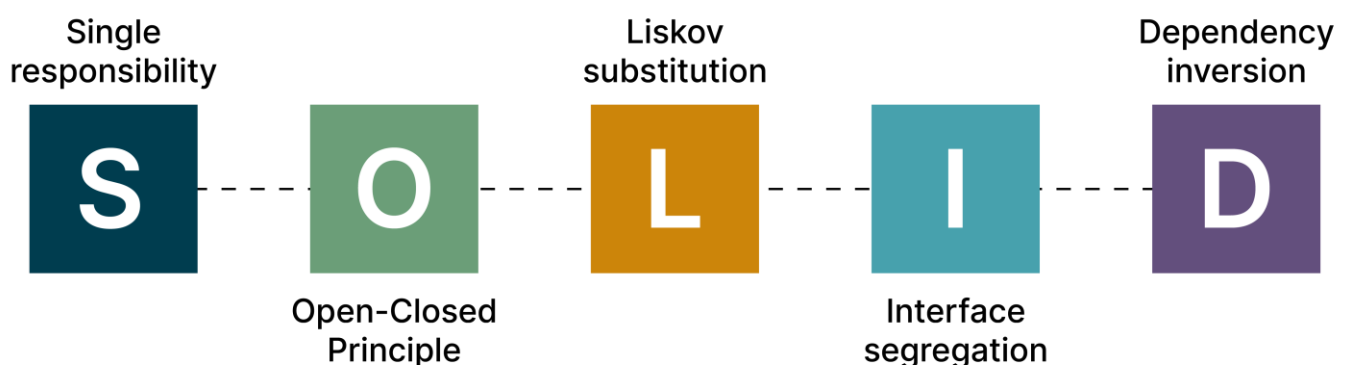


Figure 2.4 SOLID principles

2.3.2 TEST DRIVEN DEVELOPMENT

Embracing Test Driven Development (TDD), I've adopted a mindset centered on bugless software development, reinforcing the reliability of our backend APIs for the dashboards. Through TDD, unit test cases are meticulously crafted and executed prior to actual code implementation, ensuring robust functionality. This proactive approach not only identifies potential issues early in the development cycle but also establishes a comprehensive suite of tests for ongoing code maintenance. TDD acts as a catalyst for improved code quality, reducing the likelihood of post-implementation bugs and enhancing the overall stability of the software.



Figure 2.4 Test Driven Development

CHAPTER 3

WEB PORTAL FOR DENTAL CLINICS MANAGEMENT SOLUTIONS

A unified web portal for the various management of activities of a parent dental like acquisition of various other clinics, recruitment for clinics and application of various jobs in the clinics.

3.1 COMMON COMPONENTS DEVELOPMENT

Common components are the various basic and generic components of the entire application which have been used in the web portal.

3.1.1 Alert messages

This custom alert message is used everywhere in the application for notifying the users for their action.

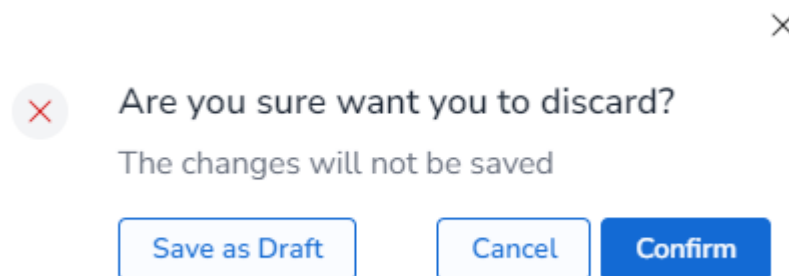


Figure 3.1 CSV Alert message

3.1.2 Dropdown

This dropdown component is a customized one which is not built using any library. This enables the user to select any option from the listed options.

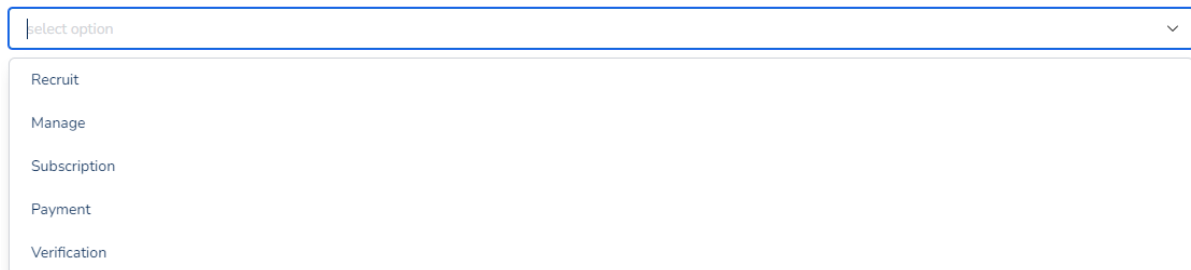


Figure 3.2 Dropdown

3.1.3 Generic Card for Information Display

This generic card has been used in the application everywhere where there is a need to display some basic information of the user.



Figure 3.3 Generic card

3.1.4 Guide Card

This guide card enables the user to understand what a particular part of the application is used for.

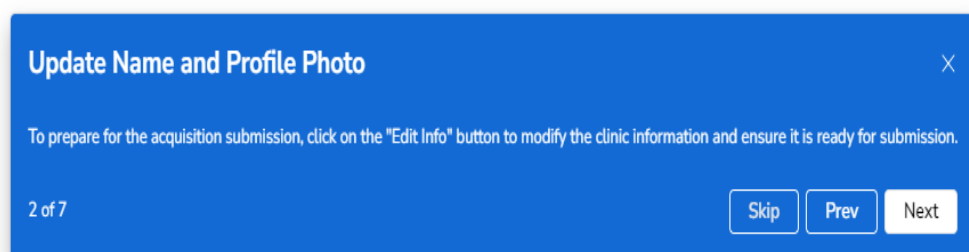


Figure 3.4 Guide card

3.1.5 Job Cards

This enables the users to view information about the job that has been posted in the application.

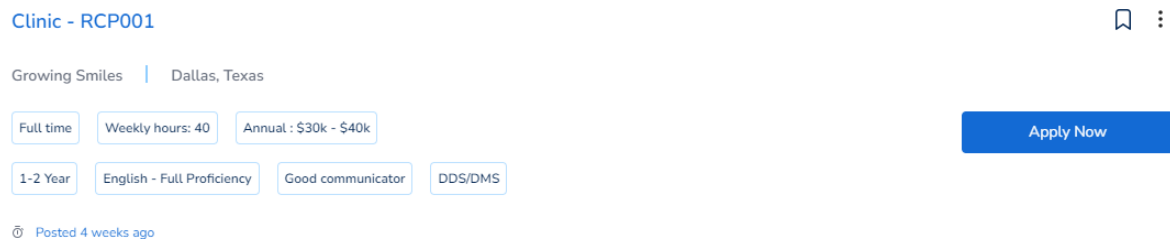


Figure 3.5 Job card

3.1.6 Step Dot Progress bar

This component is used for letting the user know what is the current status of some task in the application.



Figure 3.6 Step dot progress bar

3.1.7 Three Dots Menu

This enables the users to view more actionable items in particular parts of the application.

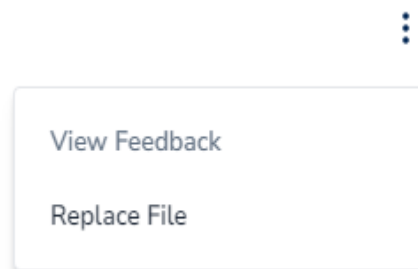


Figure 3.7 Three dots menu

3.1.8 Toast Messages

The variant of toast messages will be displayed after every action that has been performed by the user. It includes success, warning, failure or basic info toast.

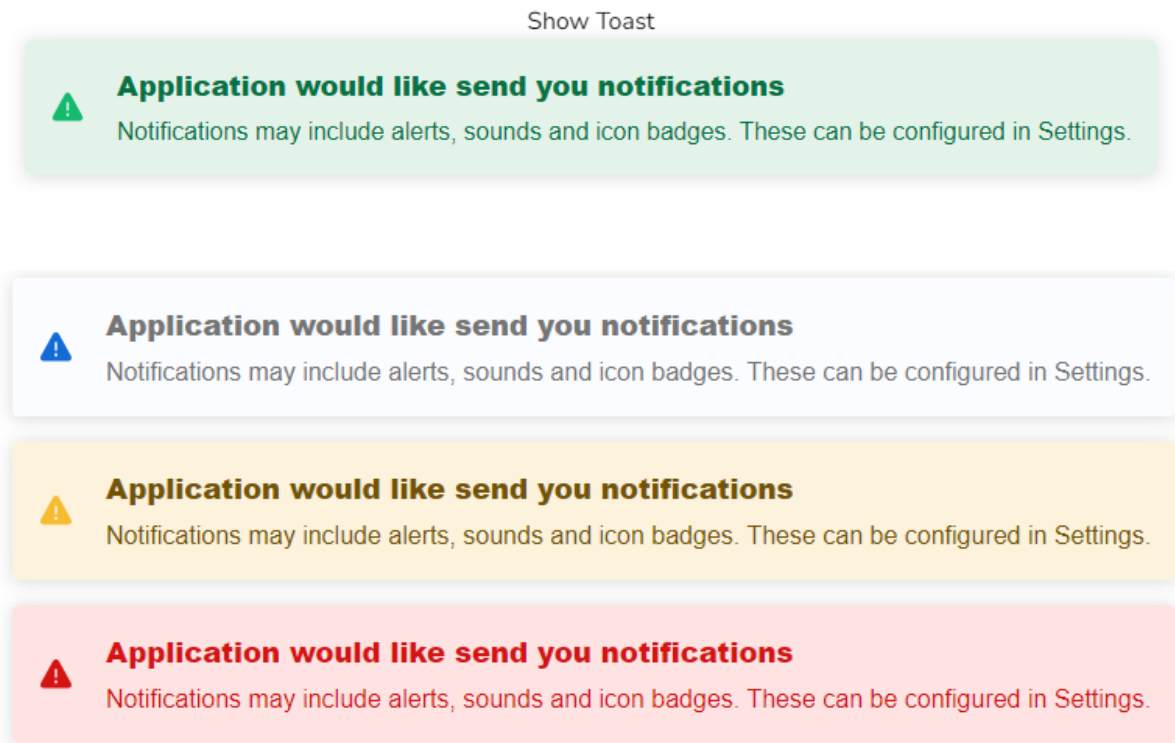


Figure 3.8 Toast messages

3.1.9 Typography

This typography component is used everywhere in the application to make the UI consistent for the text displayed in the application.

The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog
The quick brown fox jumps over the lazy dog

Figure 3.9 Typography

3.1.10 Pill Tabs

This pill tab enables the user to select different sections of a particular page.



Figure 3.10 Pill tabs

3.1.11 Table

This table component is used where there is need to display a group of information in the application.

#	Clinic Name	Location	Type	
	Abstergo Ltd.	Detroit, MI	Full-time	Remove
	Barone LLC.	Detroit, MI	Full-time	Remove
	Big Kahuna Ltd.	Detroit, MI	Full-time	Remove
	Biffco Enterprises Ltd.	Detroit, MI	Full-time	Remove
	Acme Co.	Detroit, MI	Full-time	Remove

Figure 3.11 Table

3.2 STORYBOOK FOR EVERY COMPONENTS

An open-source tool for UI development, Storybook is essentially a library or a playground for UI components. The tool enables developers to showcase the interaction of the components in an isolated development environment, making it easier to test and debug the components and collaborate with fellow developers.

3.2.1 Problem

- The web's universality is pushing more complexity into the frontend. It began with responsive web design, which turned every user interface from one to 10, 100, 1000 different user interfaces. Over time, additional requirements piled on like devices, browsers, accessibility, performance, and async states.
- Component-driven tools like React, Vue, and Angular help break down complex UIs into simple components but they're not silver bullets. As frontends grow, the number of components swells. Mature projects can contain hundreds of components that yield thousands of discrete variations.
- To complicate matters further, those UIs are painful to debug because they're entangled in business logic, interactive states, and app context.
- The breadth of modern frontends overwhelms existing workflows. Developers must consider countless UI variations, yet aren't equipped to develop or organize them all. Developer end up in a situation where UIs are tougher to build, less satisfying to work on, and brittle.

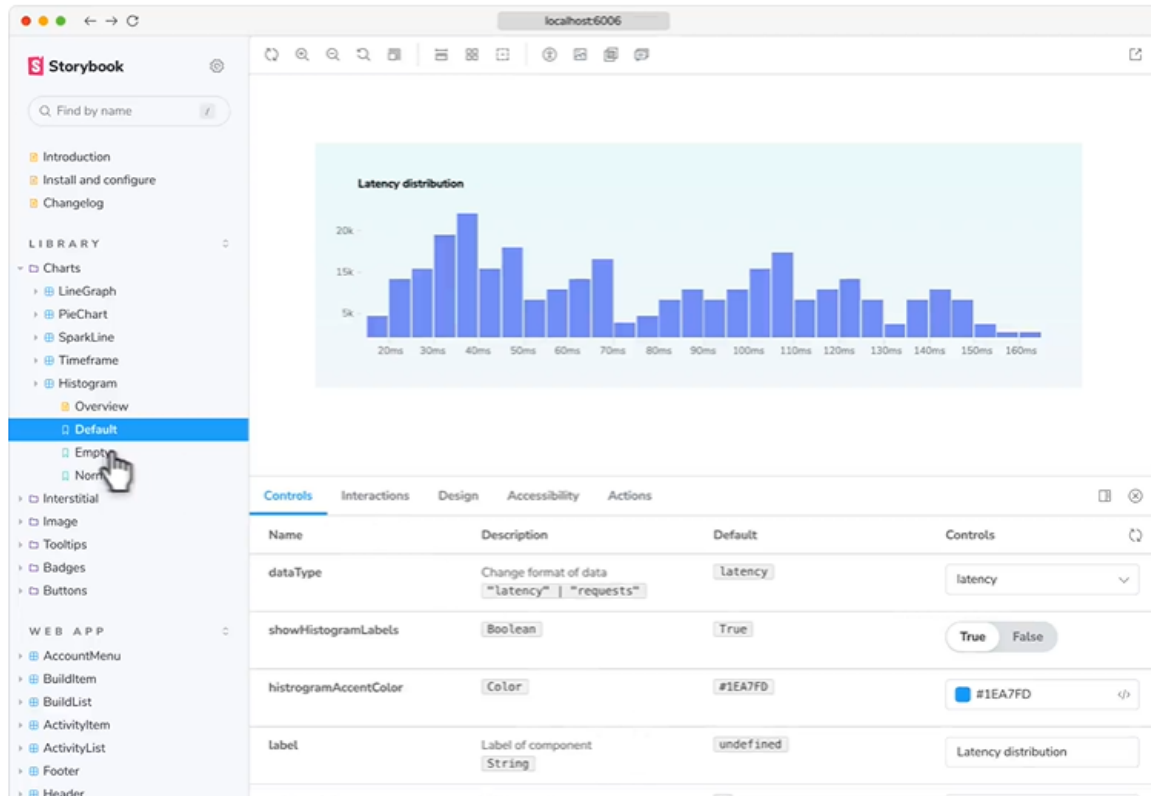
3.2.2 NEED FOR STORYBOOK

- Every piece of UI is now a component. The superpower of components is that developers don't need to spin up the whole app just to see how they render. Developers can render a specific variation in isolation by passing in props, mocking data, or faking events.
- Storybook is packaged as a small, development-only, workshop that lives alongside in the app. It provides an isolated iframe to render components without interference from app business logic and context. That helps developers focus development on each variation of a component, even the hard-to-reach edge cases.



Figure 3.12 Need for storybook

- It is used to isolate a component of a UI separately. The developers can see the list of components available and check every state of the component and properties that can be applicable to the component.
- It is also used to check the compatibility of the component in various browsers and network state.
- It is also used to check the responsiveness of the component in various view ports and devices. According to this, the developer can easily develop the component in such a way it is portable for every device.



. Figure 3.13 Sample storybook UI

3.2.3 Benefits of Storybook

Develop UIs that are more durable

Isolate components and pages and track their use cases as stories. Verify hard-to-reach edge cases of UI. Use addons to mock everything a component needs—context, API requests, device features, etc.

Test UIs with less effort and no flakes

Stories are a pragmatic, reproducible way of tracking UI states. Use them to spot-test the UI during development. Storybook offers built-in workflows for automated Accessibility, Interaction, and Visual testing. Or use stories as test cases by importing them into other JavaScript testing tools.

Document UI for the team to reuse

Storybook is the single source of truth for the UI. Stories index all the components and their various states, making it easy for the team to find and reuse existing UI patterns.

Storybook also auto-generates documentation from those stories.

Share how the UI actually works

Stories show how UIs actually work, not just a picture of how they're supposed to work. That keeps everyone aligned on what's currently in production. Publish Storybook to get sign-off from teammates. Or embed them in wikis, Markdown, and Figma to streamline collaboration.

Automate UI Workflows

Storybook is compatible with the continuous integration workflow. Add it as a CI step to automate user interface testing, review implementation with teammates, and get signoff from stakeholders.

3.2.4 Story

- A story captures the rendered state of a UI component. Developers write multiple stories per component that describe all the “interesting” states a component can support.
- The CLI created example components that demonstrate the types of components that can build with Storybook: Button, Header, and Page.
- Each example component has a set of stories that show the states it supports. Developers can browse the stories in the UI and see the code behind them in files that end with `.stories.js` or `.stories.ts`. The stories are written in Component Story Format (CSF)--an ES6 modules-based standard--for writing component examples.

3.2.5 Stories of Component

In the application there are various components with multiple rendering states.

For example, there is a button with two variants 'Dark' and 'Light' and also with icons rendering inside the components. Figure 3.14 shows the 'Dark' variant and Figure 3.15 shows the 'Light' variant.

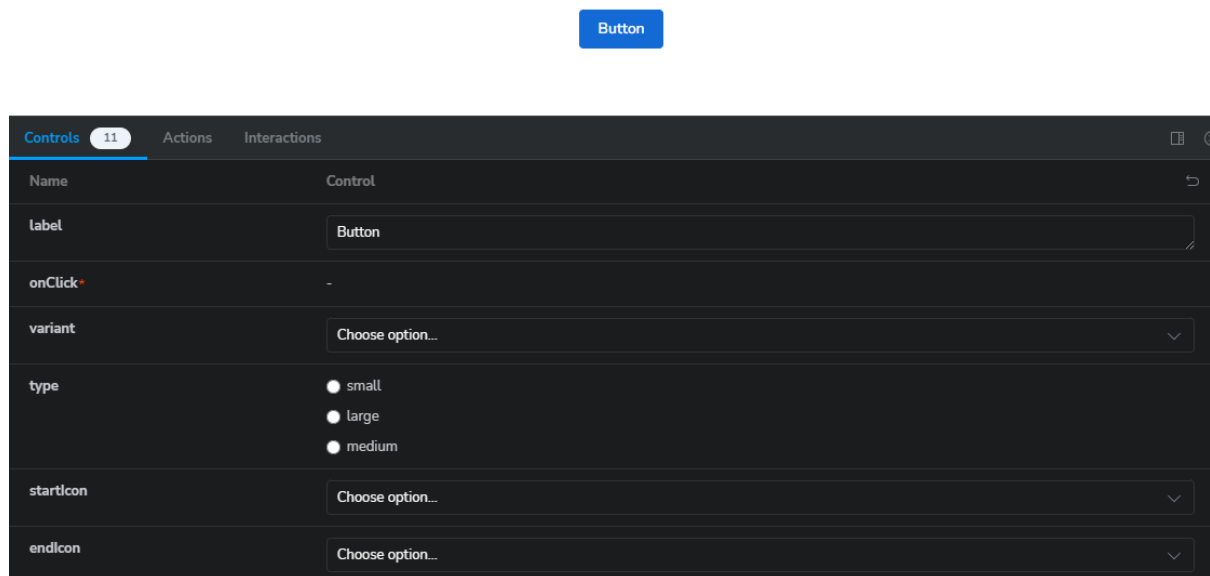


Figure 3.14 Dark variant

The above figure shows the exact UI of the storybook. There are various control options for a single component. From the UI itself, the developers are able to control the rendering state of a component. In this case, the developers can control the variant of the button as 'dark' and 'light', add a **handler function** to the button, add an icon to the button based the position as start or end, control the size of the button as small, large and medium as per the standard design requirements.



Controls 11 Actions Interactions

Name	Control
label	Button
onClick	-
variant	light
type	<input checked="" type="radio"/> small <input type="radio"/> large <input type="radio"/> medium
starticon	Choose option...
endicon	Choose option...

Figure 3.15 Light variant

State of button with start icon



Controls 11 Actions Interactions

Name	Control
label	Button
onClick	-
variant	light
type	<input checked="" type="radio"/> small <input type="radio"/> large <input type="radio"/> medium
starticon	Choose option...
endicon	Choose option...

Figure 3.16 Button with icon at start

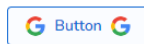
State of the button with end icon



Controls 11		Actions	Interactions
Name	Control		
label	Button		
onClick	Setup controls >		
variant	light		
type	<input type="radio"/> small <input type="radio"/> large <input type="radio"/> medium		
starticon	None		
endicon	Choose option...		

Figure 3.17 Button with icon at end

State of the button with both icons



Controls 11		Actions	Interactions
Name	Control		
label	Button		
onClick	-		
variant	light		
type	<input type="radio"/> small <input type="radio"/> large <input type="radio"/> medium		
starticon	Choose option...		
endicon	Choose option...		

Figure 3.18 Button with icons at both side

The above states of the button are written as each separate story in javascript with file named “*.stories.tsx/jsx”.

3.3 UI for Different Type of Users

This project involves various types of end users like clinic owners, back office team, recruitment team, applicant of jobs etc.

The UI for each and different users varies according to their perspective of their role, technical requirements and nature of their responsibilities.

3.3.1 UI for Clinic Profile

Clinic owners are the end users who can manage their clinics through the dedicated portal. The kind of activities they can perform through this portal are:

- Add a new clinic
- Delete an existing clinic
- Update the information of an existing clinic
- Able to view the details of multiple clinics under their portal

Information of clinics stored

Each and every clinic contains the following fields to be saved.

PRIMARY DETAILS

- Practice name
- Phone number
- Email
- Website
- City
- State

OTHER DETAILS

- Number of partners
- Number of associates
- Number of non-clinical employees
- Number of owners
- Type of clinic
- Additional information

Clinic Profile

[Edit Info](#)

Practice Name*

Website Address

https://

Email*

Phone Number*

 US ▾

City*

State*

Figure 3.19 Primary details

Other Details

[Edit Info](#)

Number of Ops

Number of Associates

Total Number of Nonclinical Employees

Number of Partners

Number of Owners

Number of Hygienists

Type

Additional Relevant Info

0 / 300

Figure 3.20 Other details

3.3.2 UI for Owner Profile

Similarly, the owner can also manage their profile under the portal.

Information of owner stored:

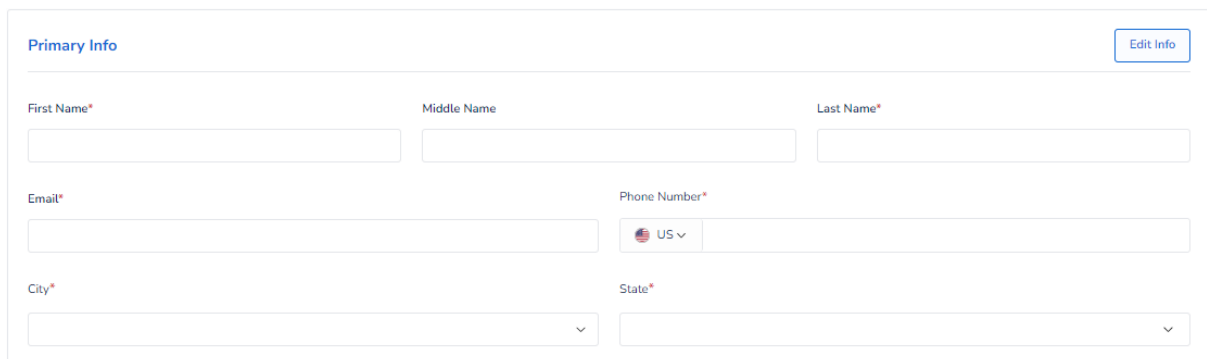
UI for the owner contains the following to be saved.

Primary Details:

- Owner name
- Email
- Website
- Phone number
- City
- State

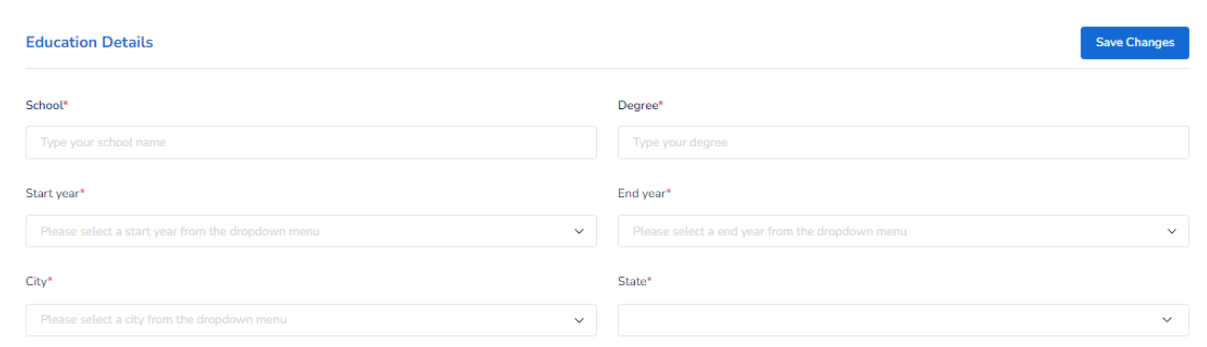
Education Details:

- School name
- Degree
- Start year
- End Year
- City
- State



The form is titled "Primary Info" in blue text at the top left. In the top right corner, there is a blue button labeled "Edit Info". The form contains several input fields: "First Name*" (required), "Middle Name", and "Last Name*" (required), each with a corresponding text input box. Below these, there is an "Email*" (required) field and a "Phone Number*" (required) field. The phone number field includes a country code dropdown menu currently set to "US" with a flag icon. At the bottom, there are "City*" (required) and "State*" (required) dropdown menus.

Figure 3.21 Primary Details



The form is titled "Education Details" in blue text at the top left. In the top right corner, there is a blue button labeled "Save Changes". The form contains several input fields: "School*" (required) with a text input box and placeholder "Type your school name"; "Degree*" (required) with a text input box and placeholder "Type your degree"; "Start year*" (required) with a dropdown menu and placeholder "Please select a start year from the dropdown menu"; "End year*" (required) with a dropdown menu and placeholder "Please select an end year from the dropdown menu"; "City*" (required) with a dropdown menu and placeholder "Please select a city from the dropdown menu"; and "State*" (required) with a dropdown menu.

Figure 3.22 Education Details

3.4 Integration of Different Portals with Backend

The Database used for the project is **POSTGRESQL** and the backend code has been written in **NESTJS** framework along with **GRAPHQL**.

Instead of calling **REST API** like **GET/POST/PUT/DELETE**, in GraphQL, **Queries** and **Mutation** along with service functions are used

QUERIES

GraphQL has its own syntax for querying data. Instead of multiple endpoints with fixed responses, developer send a single query to the server, specifying the data that one's need. This query can be shaped to request only the fields interested in, reducing over-fetching of data.

Sample Query:

```
query GetUser($userId: ID!) {  
  user(id: $userId) {  
    name  
    email  
  }  
}
```

For the project, querying the clinic's information and owner information from the database involved these types of GraphQL queries along with service layer functions.

From the frontend, the respective query files will be created and called in the respective pages for retrieving the information. The API calls will be made and the values are stored in store using **MOBX** - state management library.

Instead of managing the states of react locally, the **MOBX** global stores are used and the components of each and every page are made observable to this global store. The components get rendered each time when the values in the store get changed.

MUTATION

While queries are used for reading data, mutations are used for writing or modifying data on the server.

Sample mutation

```
mutation {  
  createUser(name: "John Doe", email: "john@example.com") {  
    id  
    name  
    email  
  }  
}
```

After every mutation, the UI has to be updated seamlessly for the better experience for the end users. So the values of the global store are managed properly for the updation of UI.

The API calls are also optimized in both the logic and technical way to enhance faster operation between frontend and backend.

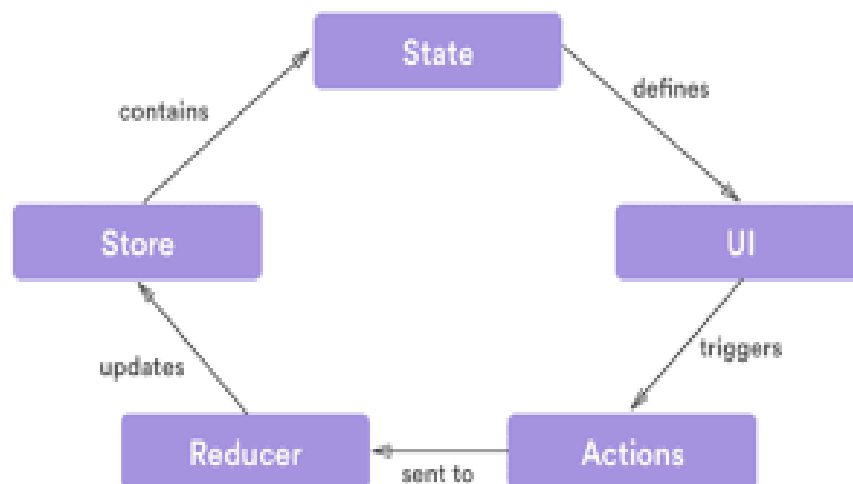


Figure 3.23 Global state management

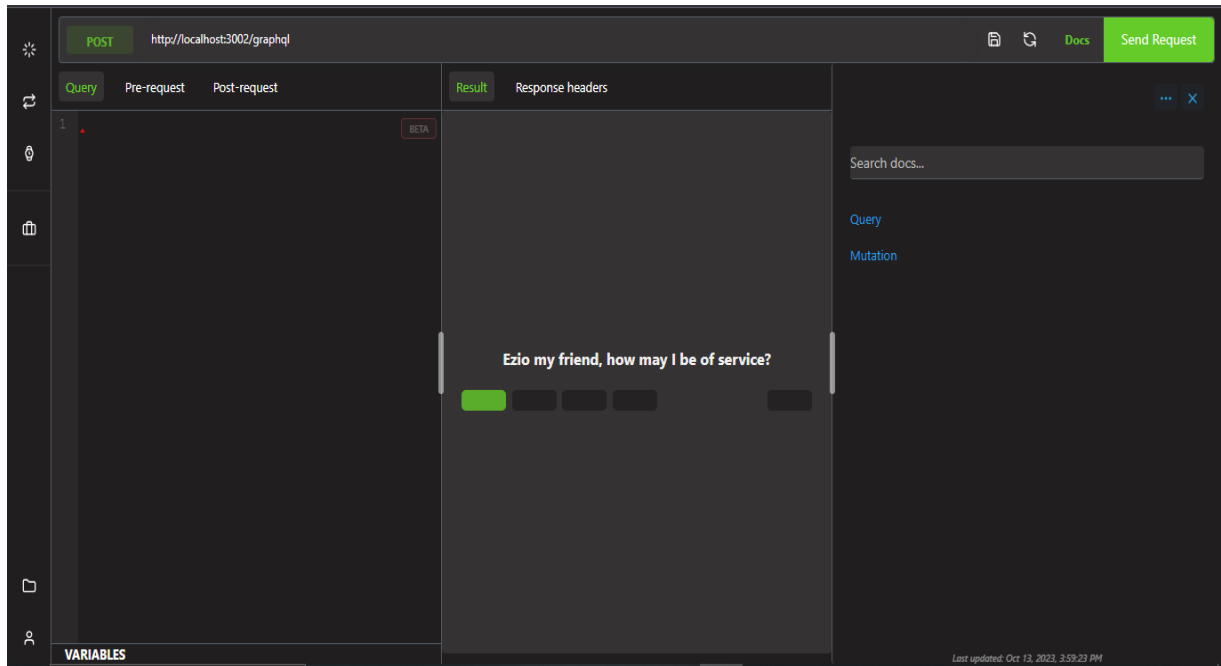


Figure 3.24 UI of “Altair” software for backend API exploration

3.5 TESTING

Testing is a very important phase in the development life cycle of an application. It ensures that the application is of high quality. Testing requires careful planning and execution. It is also the most time-consuming phase of the development.

3.5.1 UI Testing

The UI of each and every page is tested based on the design provided through the **FIGMA**. The attributes like padding, margin, gap between sections, colors, fonts, borders, border colors, font color, font size, font weight, line height, icons size are tested seamlessly to ensure a smooth UI experience. The bugs will be posted in ticket management tools like **Monday.com** once after the developers finished with their development.

3.5.2 Backend Testing

Backend testing is the most important phase of testing as it involves the most logic-based implementation of database manipulation. The tester of the team crafts the test plan execution along the test cases for each and every queries and mutations. The bugs will be identified through this rigorous testing method to help developers to make their code and logic clearer.

3.5.3 Integration Testing

Integration testing involves both UI and backend testing along with external components of the application like database, web service, etc., It simulates or mocks the real environment to find nearly all bugs, but it is the most complicated process. It also involves the testing of the complete scenario of a user activity in the application starting from the log in, activities in the application of the portal and logging out.

CHAPTER 4

CONCLUSION

The basic use of NextJS framework is to develop web applications with minimal development effort. The hot reload functionality in NextJS allows developers to quickly and easily experiment, build UIs, add features, and fix bugs. The server side rendering of NextJS enables the application to load faster than other modern web development frameworks. So, the choice of framework for the application development is NextJS for UI and NestJS for backend with GraphQL.

Over the course of the internship, several UI challenges and new work models were explored and implemented in NextJs. The functionalities are tested using integration testing and made sure to work in all corner cases. State management-based architecture was explored and configured according to the needs of the application. Each and every aspect of the project was accomplished by following the Software Development Lifecycle. Testing the application in the production environment will be the future work and thus incorporating newer requirements that might come in between and changes will be made accordingly.

BIBLIOGRAPHY

WEBSITES

1. <https://nextjs.org/>
2. <https://graphql.org/>
3. <https://www.postgresql.org/>
4. <https://nestjs.com/>
5. <https://storybook.js.org/>
6. <https://stackoverflow.com/>
7. <https://altairgraphql.dev/>
8. <https://monday.com/>