REACT DASHBOARD AND ANALYTTICS PLATFORM

AN INTERNSHIP PROJECT

Submitted by

PATEL KRINAL JANAKBHAI

191210107047

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

In

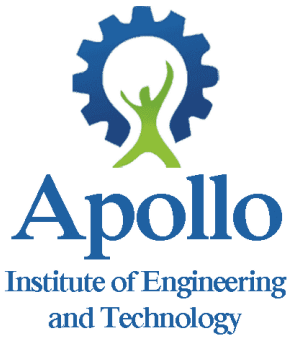
Computer Engineering

Apollo Institute Of Engineering And Technology

Ahmedabad

Gujarat Technological University, Ahmedabad

May, 2023



Apollo Institute of Engineering and Technology

Ahmedabad

CERTIFICATE

This is to certify that the project entitled **"React Dashboard and Analytics Platform"** has been carried out by **"PATEL KRINAL JANAKBHAI (191210107047)"** under my guidance in partial fulfillment for the Internship Project Report (3180701)Subject of Bachelor of Engineering in **Computer Engineering**, 8th Semester of Gujarat Technological University, Ahmadabad during the academic year 2022-23.

Name of Internal Guide

Prof. Krupa Suthar

Head of the Department

Prof. Richa Bhadauria

GUJARAT TECHNOLOGICAL UNIVERSITY

CERTIFICATE FOR COMPLETION OF ALL ACTIVITIES AT ONLINE PROJECT PORTAL

B.E. SEMESTER VIII, ACADEMIC YEAR 2022-2023

Date of certificate generation : 12 May 2023 (22:13:34)

This is to certify that, ***Patel Krinal Janakbhai*** (Enrolment Number - 191210107047) working on project entitled with ***React Dashboard and analytics platform*** from ***Computer Engineering*** department of ***APOLLO INSTITUTE OF ENGINEERING & TECHNOLOGY, AHMEDABAD*** had submitted following details at online project portal.

Internship Project Report	**Completed**

Name of Student : Patel Krinal Janakbhai

Name of Guide : Miss. Krupa B Suthar

Signature of Student : _____

*Signature of Guide : _____

Disclaimer :
This is a computer generated copy and does not indicate that your data has been evaluated. This is the receipt that GTU has received copy of the data that you have uploaded and submitted as your project work.

*Guide has to sign the certificate only all above activities has been completed.

INFOLABZ IT SERVICES PVT. LTD.
WEB DEVELOPMENT | APP DEVELOPMENT | DATA SCIENCE | IOT

Date: 21 / 04 / 2023

TO WHOM IT MAY CONCERN

This is to certify that Krinal Janakbhai Patel, a student of Apollo Institute of Engineering and Technology has successfully completed her internship in the field of Web Development from 23 January 2023 to 15 April 2023 (Total number of Weeks: 12) under the guidance of Mr. Chintan Nagrecha.

Her internship activities include front-end web development in React, API development and Integration.

During the period of her internship program with us, she had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish her every success in her life and career.

Ms. Zarna Shah,
Human Resources Department,
INFOLABZ IT SERVICES PVT. LTD.

91 8866 2662
1 8141236662
contact@infolabz.in
www.infolabz.in
405 Vraj Avenue, Above SAM'S Pizza
Nr. Commerce Six Rd, Navrangpura,
Ahmedabad, Gujarat 380009

Apollo Institute Of Engineering and Technology

Ahmedabad 382330

DECLARATION

We hereby declare that the Internship report submitted along with the Internship Project entitled **React Dashboard and Analytics Platform** 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 INFOLABZ IT SERVICES PVT. LTD. under the supervision of Chintan Nagrecha 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.

Candidate's Name **Student's Signature**

PATEL KRINAL

Acknowledgment

First of all, I sincerely thank to my External guide Mr. CHINTAN NAGRECHA for continuously guiding me at the company and answering all my doubts with patience. I am grateful for his prolonged interest in my work and excellent guidance. He has been a constant source of motivation to me. I also thank my parents, friends and all the members of the family for their precious support and encouragement which they had provided in completion of our work. In addition to that, I would also like to mention the company personals who gave me the permission to use and experience the valuable resources required for the internship.

Thus, in conclusion to the above said, I once again thank the staff members of Infolabz IT Services Pvt. Ltd. for their valuable support in completion of the project.

PATEL KRINAL
191210107047

Abstract

The Dashboard and Analytics Platform is a comprehensive solution designed to provide businesses with real-time insights and analytics. The platform is built using React and integrates with various data sources to provide a centralized view of business metrics and KPIs. The platform's intuitive dashboard displays up-to-date information on key performance indicators and trends, enabling businesses to quickly identify areas of improvement and track progress towards their goals. The platform also features advanced analytics capabilities, such as data visualization and predictive analytics that allow businesses to gain deeper insights into their data and make informed decisions. With its user-friendly interface and real-time updates, the Dashboard and Analytics Platform is an essential tool for businesses looking to stay ahead of the curve and make data-driven decisions.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENT

1. OVERVIEW OF THE COMPANY

1.1 History

[Fig 1.1.1 Infolabz IT Services PVT.LTD]

Established in 2016, incorporation with our parent IT company, INFOLABZ IT SERVICES PVT. LTD. has managed to make its own position in IT Sector. We are involved in Web Development, App Development, Progressive Web Application Development, IOT solutions, Graphics & Designing, Digital Marketing, Domain & Hosting services, SMS services etc.

In the span of six years we have managed to deliver all projects on time with utmost accuracy to our clients across the globe. We have dedicated teams of experienced and hardworking developers. Our developers, who are always willing to take new challenges and looking forward to learn new things, are heart of this company.

Our objective is to sustain with exponential growth in IT industry. Our mission is to deliver the best with top notch quality every quarter and vision is to develop a product with one of its kind concept which could be used by millions of people.

1.2 Different Products

➢ Web Development

Our team of dedicated developers is well-versed in web and software development. We provide state of the art frontend and backend services using the best technologies and frameworks in web development. With insights about the latest technology trends, knowledge and expertise in web development, our dedicated developers optimize your web applications for maximum benefit. We create web applications for our clients to help their business grow by streamlining business processes, internal communication, organizational efficiency, increasing revenue growth and gaining an edge over competitors.

➢ Mobile Development

With the right knowledge and skills about the latest technology in mobile app development, Infolabz creates mobile app solutions for our clients across sectors. Mobile apps have become a major point of attention for users given the huge leap in technology, internet speed, and access. Today, mobile apps are the main medium of digital interaction. Therefore, mobile app development is an important factor in consumer retention and business growth. Our expert mobile app development team has the expertise to develop globally popular Android and iOS apps so that you can scale and sustain your business.

➢ Data Science

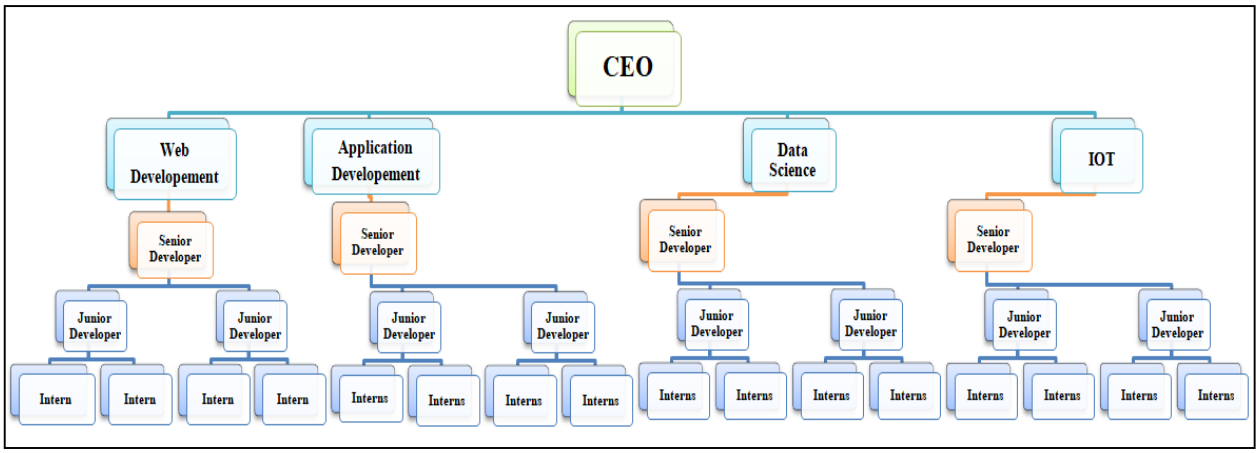
Infolabz specializes in discovering new business opportunities and solving real-time problems using data science and machine learning. Our data science services include data analysis, predictive modeling, machine learning, and data visualization. We utilize advanced data science tools and technologies to deliver high-quality results to clients, and we have experience working with a variety of data types and formats, including structured and unstructured data. Our data science services are customized to meet the specific needs of each client, ensuring that they receive tailored solutions to their business problems.

➢ Internet Of Things

From small businesses to large corporations, the Io[illegible] around us

We use IoT to make your business stronger and more successful.

1.3 Organization Chart



[Fig 1.3.1 Organization Chart of Infolabz IT Services]

1.4 Capacity of Company

Area: 3450 sqFt

Employee Capacity: 30 Employees

Intern Capacity: 30 Interns

Total Capacity: 60

Web Development Department: 12

App Development Department: 8

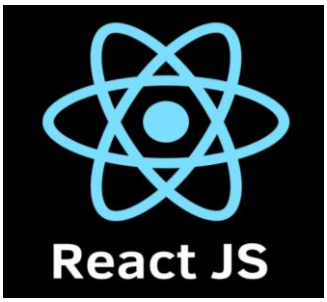
Machine Learning / Data Science: 4

IoT: 3

Ongoing Projects : 7 (As on 15 April 2023)

2. LANGUAGE INTRODUCTION

2.1 Introduction



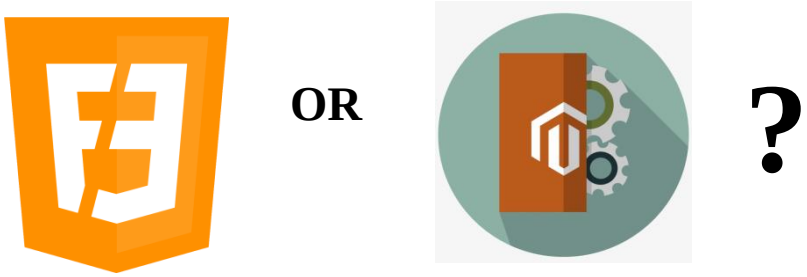
[Fig 2.1.1 React JS logo]

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front end library responsible only for the view layer of the application. It was created by Jordan Walke, who was a software engineer at Facebook. It was initially developed and maintained by Facebook and was later used in its products like WhatsApp & Instagram. Facebook developed ReactJS in 2011 in its newsfeed section, but it was released to the public in the month of May 2013 ReactJS uses virtual DOM based mechanism to fill data in HTML DOM. The virtual DOM works fast as it only changes individual DOM elements instead of reloading complete DOM every time.

Today, many JavaScript frameworks are available in the market(like angular, node), but still, React came into the market and gained popularity amongst them. The previous frameworks follow the traditional data flow structure, which uses the DOM (Document Object Model). DOM is an object which is created by the browser each time a web page is loaded. It dynamically adds or removes the data at the back end and when any modifications were done, then each time a new DOM is created for the same page. This repeated creation of DOM makes unnecessary memory wastage and reduces the performance of the application.

Therefore, a new technology ReactJS framework invented which remove this drawback. ReactJS allows you to divide your entire application into various components. ReactJS still used the same traditional data flow, but it is not directly operating on the browser's Document Object Model (DOM) immediately; instead, it operates on a virtual DOM. It means rather than manipulating the document in a browser after changes to our data, it resolves changes on a DOM built and run entirely in memory. After the virtual DOM has been updated, React determines what changes made to the actual browser's DOM. The React Virtual DOM exists entirely in memory and is a representation of the web browser's DOM. Due to this, when we write a React component, we did not write directly to the DOM; instead, we are writing virtual components that react will turn into the DOM.

2.2 Frontend or Backend Information



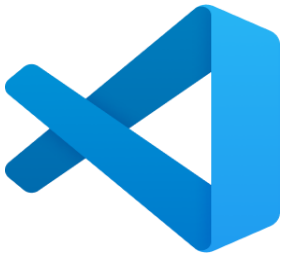
[Fig. 2.2.1 Frontend OR Blackened]

Frontend and Backend are the two most popular terms used in web development. The front end is what users see and interact with and the backend is how everything works. Each side needs to communicate and operate effectively with the other as a single unit to improve the website's functionality. React has recognized itself as the most commonly used Front-end JS framework in the world. It is popular with both software developers and project sponsors. React is not a backend development. When React Developers build a project they mainly focus on what the user's experience and see is called Frontend. It is used for client-side programming building things. The user interacts directly with the website. React is a solid and strong framework. It is commonly used. It is easy to develop and has a large community.

2.3 Tools & Technology used in React JS

Modern computers are very complex and in order to productively program them, various abstractions are needed. For example, rather than writing down a program's binary representation a programmer will write a program in a programming language like C, Java or Python. Programming tools like assemblers, compilers and linkers translate a program from a human write-able and readable source language into the bits and bytes that can be executed by a computer. Interpreters interpret the program on the fly to produce the desired behavior.

2.3.1 Integrated development environment (VS CODE)



[Fig 2.3.1 Visual Studio Code logo]

Visual Studio Code is “a free-editor that helps the programmer write code, helps in debugging and corrects the code using the intelli-sense method ”. In normal terms, it facilitates users to write the code in an easy manner. Visual Studio code is a source-code editor that can be used with a variety of programming languages including C , C++ , Java , JavaScript , Node.js , React JS , Python . Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a code editor for any language. It supports many programming languages and a set of features that differs per language. Unwanted files and folders can be excluded from the project tree via the settings. Many Visual Studio Code features are not exposed through menus or the user interface but can be accessed via the command palette.

2.3.2 JavaScript Object Notation(JSON)

JavaScript Object Notation is part of the JavaScript language, not ReactJS. Since React is a JS library, it can access everything that the language supports. This data format is universal and

can be used in any programming language or platform for data exchange from Python, PHP, Kotlin to Ruby, you name it.

2.3.3 ECMAScript 6

ES6 or ECMAScript 2015 is the 6th version of the ECMAScript programming language. ECMAScript is the standardization of Javascript which was released in 2015, and subsequently renamed as ECMAScript 2015.ECMAScript and Javascript are both different in nature.React uses ES6, and you should be familiar with some of the new features like:**Arrow Functions,Variables (let, const, var),Array Methods,String New .**

2.3.4 Bootstrap

React-Bootstrap replaces the Bootstrap JavaScript. Each component has been built from scratch as a true React component, without unneeded dependencies like jQuery.As one of the oldest React libraries, React-Bootstrap has evolved and grown alongside React, making it an excellent choice as your UI foundation.

2.3.5 HTML or HTML5

HTML stands for *Hyper Text Markup Language*. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages. This language is used to annotate (at the note for computer) text so that a machine can understand it and manipulate text accordingly. Most of the markup (e.g. HTML) languages are human readable. The language uses tags to define what manipulation has to be done on the text. It is used for structuring and presenting the content on the web pages. HTML5 is the fifth version of HTML. Many elements are removed or modified from HTML5.

2.3.6 CS CSS3

CSS stands for Cascading Style Sheet. It is a styling language used for designing web pages where we can structure the styles. It is a stylesheet programming language used for describing the format and interface the elements of a markup language (usually HTML) will take. CSS3 is a more advanced version of CSS. Also known as Cascading Style Sheet Level 3, CSS is mainly responsible for the structure and formatting of web pages. One of the biggest benefits of implementing CSS3 is that it is supported by almost all modern browsers.

2.3.7 JavaScript

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved.

3. APPLICATION PROGRAMMING INTERFACE & ES6

3.1 Introduction of API

API is an abbreviation for Application Programming Interface which is a collection of communication protocols and subroutines used by various programs to communicate between them. A programmer can make use of various API tools to make its program easier and simpler. Also, an API facilitates the programmers with an efficient way to develop their software programs. Client-side JavaScript, in particular, has many APIs available to it — these are not part of the JavaScript language itself, rather they are built on top of the core JavaScript language, providing you with extra superpowers to use in your JavaScript code.

JSON in API:

JSON (JavaScript Object Notation) is a lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate. It is an open standard format that uses human-readable text to transmit data objects consisting of attribute–value pairs. It is used primarily to transmit data between a server and a web application, as an alternative to XML. In JSON, data is represented as a collection of key-value pairs, where each key is a string and each value can be a string, number, Boolean, null, array, or another JSON object.

Object Object Format:

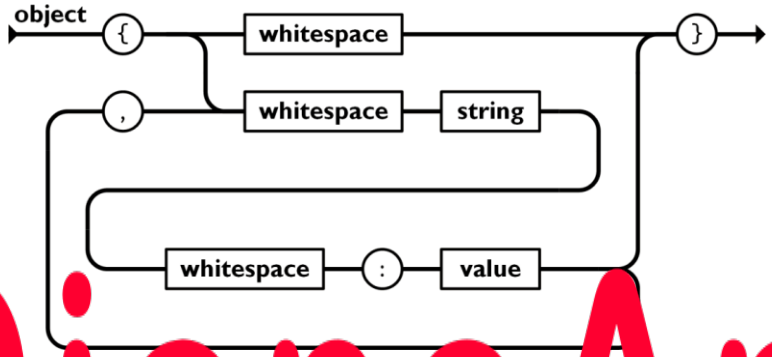
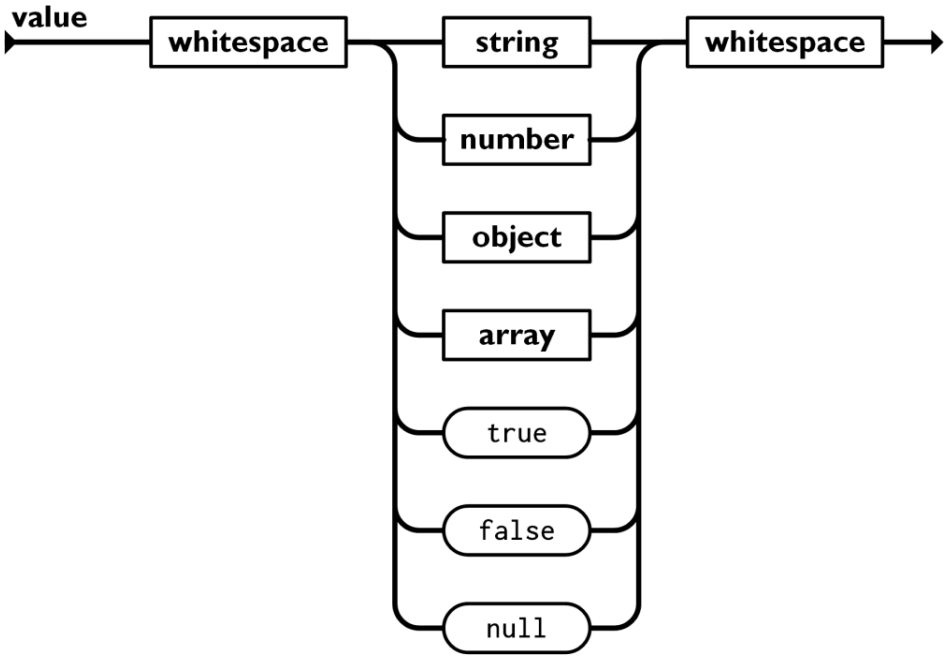


Fig 3.1.1 JSON format]

Json values:



[Fig 3.1.2 JSON values]

Fetching data from API:

async:

In JavaScript, async is a keyword used to define an asynchronous function. An asynchronous function is a function that executes its code asynchronously, meaning that it runs in the background, separately from the main program flow.

An asynchronous function returns a special type of object called a Promise, which represents a value that may not be available yet. The Promise can be used to specify what to do when the result of the asynchronous operation becomes available, either by calling a then method or by using the await operator.

```
async function load(){
    let url = "https://api.coindesk.com/v1/bpi/currentprice.json";
    let obj = await ( await fetch(url)).json();
    ment.wr te(obj["bpi"]["USD"]["rate"]);
}
```

await:

await is a JavaScript operator that is used to wait for a Promise to be resolved or rejected before moving on to the next statement in your code. It can only be used within an async function.

The await operator works by pausing the execution of the async function until the Promise that it is waiting for is either resolved (meaning that the asynchronous operation has completed successfully) or rejected (meaning that the asynchronous operation has failed).

3.2 API of Spacecraft data

Fetch API to get data as well as display some specific data as per user input data and work with Live API.

Spacecraft API: https://isro.vercel.app/api/spacecrafts

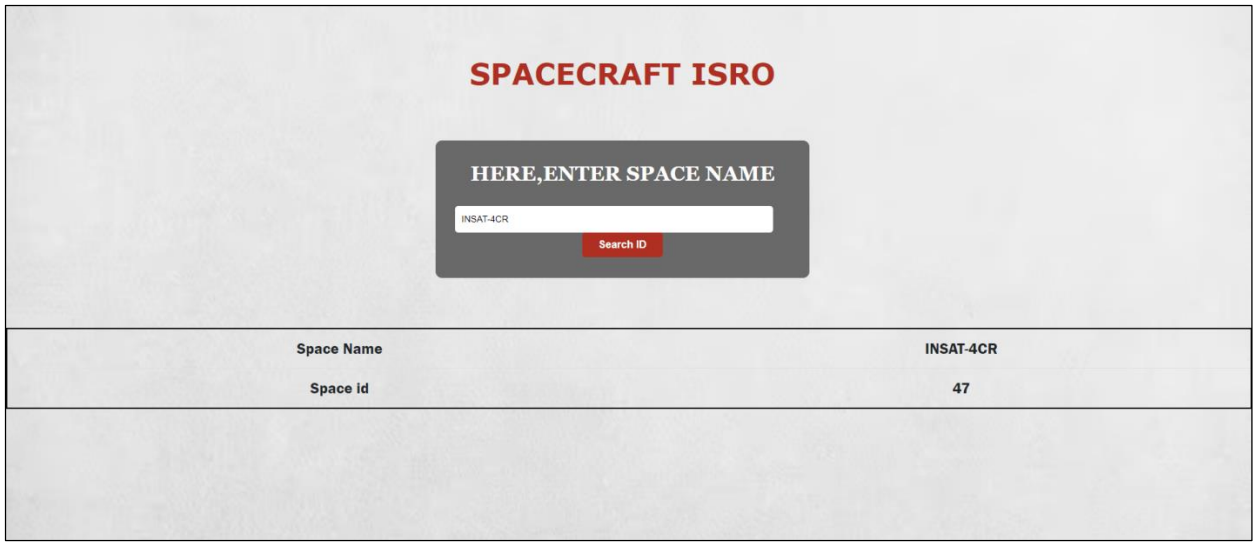
- Allow user to input Space name in text box.

- Search Space ID from this API and print Space ID.

```
<body>
    <script>
        function check(form){
            var space = form.sname.value;
            async function load(){
                let url ='https://isro.vercel.app/api/spacecrafts';
                let obj = await(await fetch(url)).json();

                for(var i=0;i<obj['spacecrafts'].length;i++){
                    if(space==obj['spacecrafts'][i]['name']){
                        document.getElementById("msg").innerHTML="<table class='table table-dark'><tr><td>
                        break;
                    }
                    else{
                        document.getElementById("msg").innerHTML="Space name is Invalid";
                    }
                }
            }
        load();
    }
    </script>
<div class="header">
    <h1 style="font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif ;color: whitesmoke
```

[Fig 3.2.1 API fetch data]

OUTPUT



[Fig. 3.2.2 Output of Valid data]

[Fig 3.2.3 Output of Invalid data]

3.3 Dictionary Search using API

A simple web application that allows the user to search for the definition of a word. The application has a user interface with a text input field and a search button. When the user enters a word and clicks the search button, the application makes an HTTP request to the Dictionary API to retrieve the definition of the word. If the word is not found in the dictionary, a message is

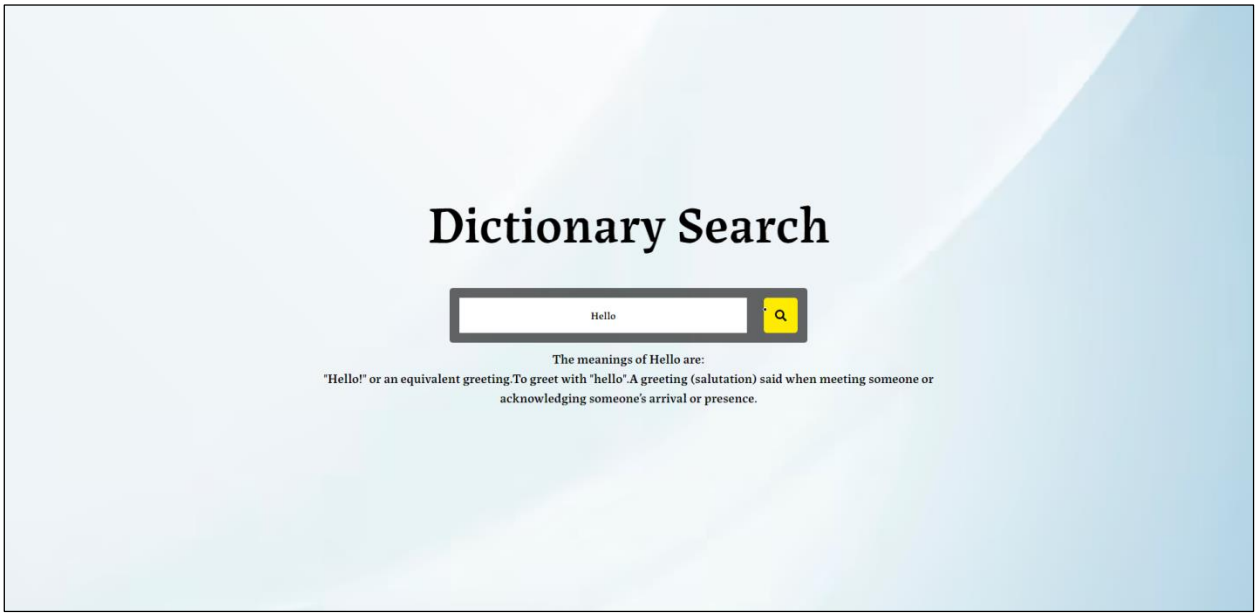
displayed indicating that no definition was found. If the word is found, the definition is displayed to the user.

The HTML code includes links to CSS and JavaScript files that provide the styling and functionality of the application. The CSS file includes styles for the centered heading, input field, search button, and message text. The JavaScript code includes a function that retrieves the user input, makes the API request, and displays the definition or error message.

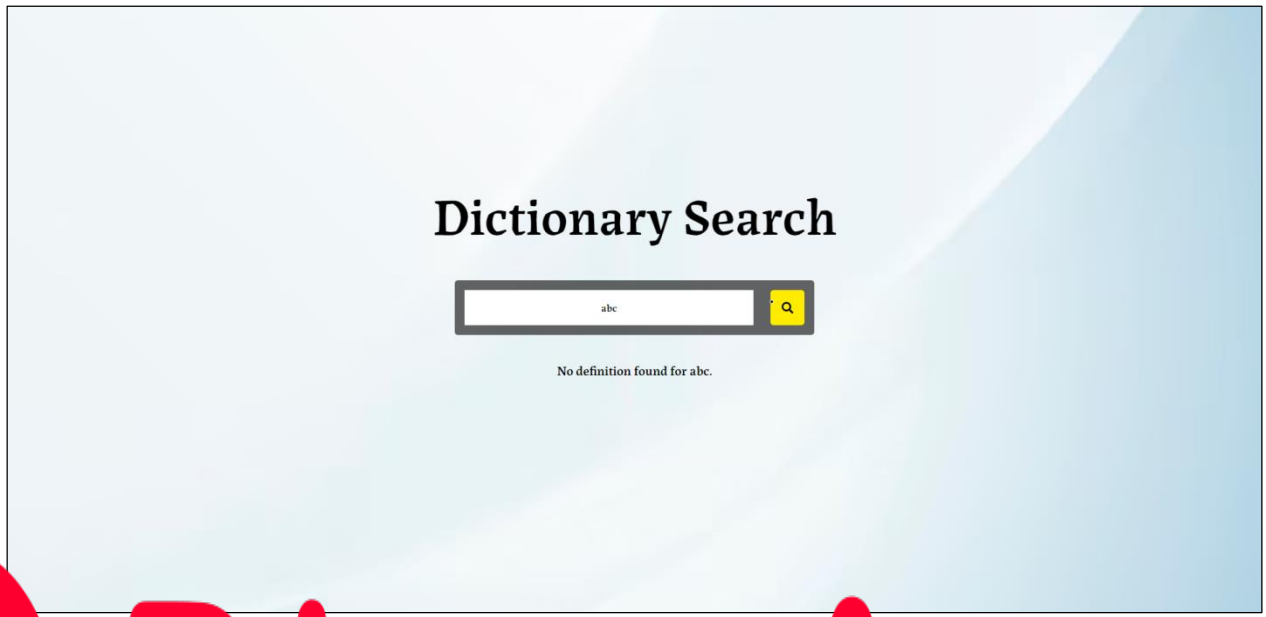
```
function load(form){
    var word = form.word.value;
    if (word==""){
        document.getElementById("msg").innerHTML="Please Enter a Word";
    }
    else{
    async function load(word){
        let url = `https://api.dictionaryapi.dev/api/v2/entries/en/${word}`;
        let response = await fetch(url);
        let data = await response.json();
        if (data.title) {
            document.getElementById("msg").innerHTML = `No definition found for ${word}.`;
        } else {
            let meanings = data[0].meanings;
            let message = `The meanings of ${word} are:<br/>`;
            for (let i = 0; i < meanings.length; i++) {
                let definition = meanings[i].definitions[0].definition;
                message += `${definition}`;
            }
            document.getElementById("msg").innerHTML = message;
        }
    }
    load(word);
```

[Fig 3.3.1 Code of Dictionary Search using API]

OUTPUT:



[Fig 3.3.2 Output of Valid Dictionary Search]



[Fig 3.3.3. Output of Invalid Dictionary Search]]

3.4 Introduction to ES6

ES6 (ECMAScript 2015) is a version of the ECMAScript programming language that introduced several new features to JavaScript, such as arrow functions, classes, and template literals. React is a JavaScript library for building user interfaces, and it can take advantage of many of the new features introduced in ES6.

Some of the key ES6 features used in React include:

1. **Arrow Functions:** Arrow functions provide a concise way to write functions in JavaScript. They use a shorter syntax than traditional function expressions and do not have their own this value.

2. **Classes:** ES6 classes provide a cleaner syntax for defining objects in JavaScript. They can be used to define constructor functions, methods, and inheritance.

3. **Template Literals:** Template literals provide an easier way to concatenate strings in JavaScript. They use backticks (`) instead of single or double quotes and can contain placeholders for variable interpolation.

4. **Destructuring:** Destructuring is shorthand syntax for assigning values from an object or array to variables. It can be used to simplify code by extracting values from complex data structures.

5. **Rest and Spread Operators:** The rest and spread operators allow for more flexible handling of function arguments and array/object manipulation. The rest operator allows a function to take an indefinite number of arguments and store them in an array, while the spread operator allows an array or object to be spread into individual arguments or properties.

6. **Default Parameters:** Default parameters allow function arguments to have default values, which are used if the argument is not provided or is undefined.

7. **Pr** Prom es provide a cleaner syntax for handli asynchronous operations in Jav Script They allow for chai in and er handlin i a mo e i uitive w than tra ional llba ks.

8. **Modules:** ES6 introduces a new module system for JavaScript, which allows for more organized and reusable code. Modules can be imported and exported, making it easier to share code between different files.

9. **Enhanced Object Literals:** Enhanced object literals provide a cleaner syntax for defining object properties and methods. They can include shorthand property names, computed property names, and method definitions.

10. **Iterators and Generators:** ES6 introduces a new iteration protocol for JavaScript, which allows for custom iteration behavior for objects. Generators are a type of function that allow for the creation of iterators.

3.5 Arrow Functions in ES6

Arrow functions allow us to write shorter function syntax.It gets shorter! If the function has only one statement, and the statement returns a value, you can remove the brackets and the return keyword. If you have parameters, you pass them inside the parentheses.

```
function two(a, b){
    document.write ("function Two Answer:"+(a+b));
}
two (100,200);
```

This JavaScript code defines a function called **two** that takes two arguments, **a** and **b**, and then writes the sum of those two values to the document using the **document.write()** method. The function is then called with the arguments **100** and **200**.

When the function is called, it will output the text "function Two Answer: 300" to the document. This code demonstrates how to define and call a simple function in JavaScript.

Same these code can define in arrow function within two line as below:

```
const two = (a,b) => document.write(a+b);
two(100,200);
```

4. REACT ENVIRONMENT & TASKS

4.1 Steps to follow

Setting up a React environment involves several steps. Here are the general steps to follow:

Step 1: Install NodeJS.

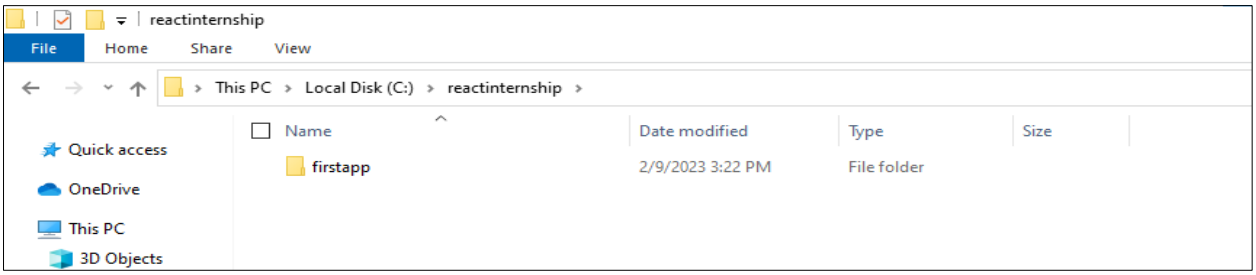
Step 2: now go to **reactinternship** folder which id created by user through command **cd reactinternship.**

```
C:\Users>cd..

C:\>cd reactinternship

C:\reactinternship>npx clear-npx-cache
Need to install the following packages:
  clear-npx-cache@1.0.1
Ok to proceed? (y) y

C:\reactinternship>
```

[Fig 4.1.1 React Launch Project]

Step 3: Run below command to create a React application named **firstapp.**

npm init react-app firstapp



[Fig 4.1.2 Create folder Using Command]

Step 4: Now you are ready to run your first real React application.

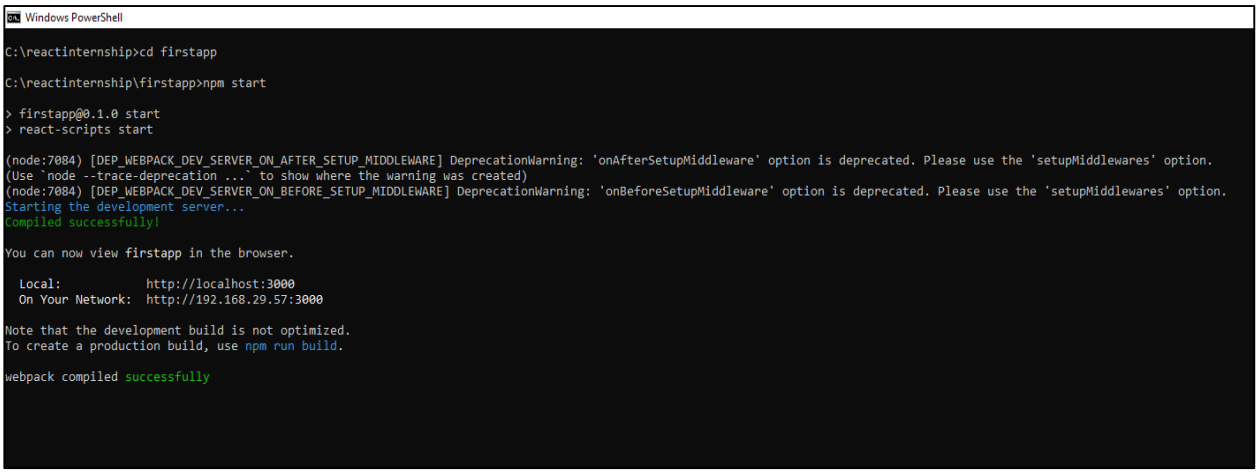
Run this command to move to the firstapp directory:

```
cd firstapp
```

Run this command to run the React application fisrtapp:

```
npm start
```

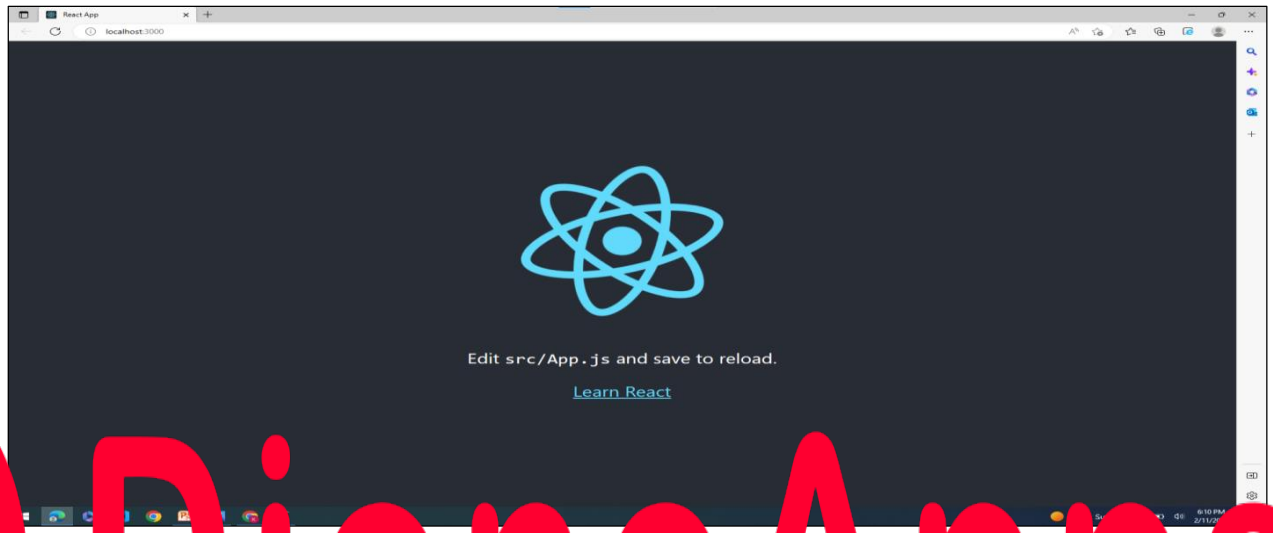
A new browser window will pop up with your newly created React App!



[Fig 4.1.3 Successfully Launch Project in Command Prompt]

The result:

In ReactJS, components are the building blocks of an application's user interface. There are two types of components that you can use in ReactJS: class components and functional components.



[Fig 4.1.4 Successfully Launch Project]

4.2 Class Component

Class components are defined using the ES6 class syntax and extend the **React.Component** class. Class components have access to lifecycle methods and state, which allow them to manage complex application logic. Here's an example of a simple class component:

App.js

```
import React, { Component } from 'react';
class Welcome extends Component {
  render() {
    return (
      <div>
        <h1>Welcome to my app!</h1>
        <p>This is a class component.</p>
      </div>
    );
  }
}
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import Welcome from './App';
ReactDOM.render(
<Welcome />,
ent.g  entBy  t'));
```

In this **index.js** file, we import the **React** and **ReactDOM** libraries, as well as the **Welcome** component that we created in the previous code example. We then use the **ReactDOM.render()** method to render the **Welcome** component and mount it to the HTML element with the **id** of "root".

index.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title> Class Component </title>
  </head>
  <body>
    <div id="root"></div>
    <script src="./index.js"></script>
  </body>
</html>
```

In this **index.html** file, we create a basic HTML document with a **div** element that will serve as the container for our React application. We also include a **script** tag that links to our **index.js** file where we have defined our React component. When the **index.js** file is loaded, it will render our **Welcome** component inside the **div** element with the **id** of "root".

OUTPUT:

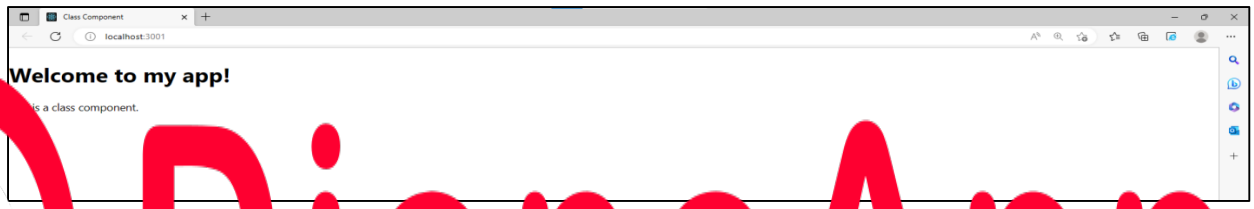


Fig. 2.1 Output of Class Component

4.3 Functional Components

Functional components are defined as functions that return JSX. They are simpler than class components and are used for presenting static UI elements. Functional components are typically used for simple UI elements that don't require state or lifecycle methods. Here's an example of a simple functional component:

App.js

This file contains a React functional component named Welcome.

The first two lines of the file import the logo and CSS files, which are used to style the React application.

The next line imports the React library, which is required to define and use React components.

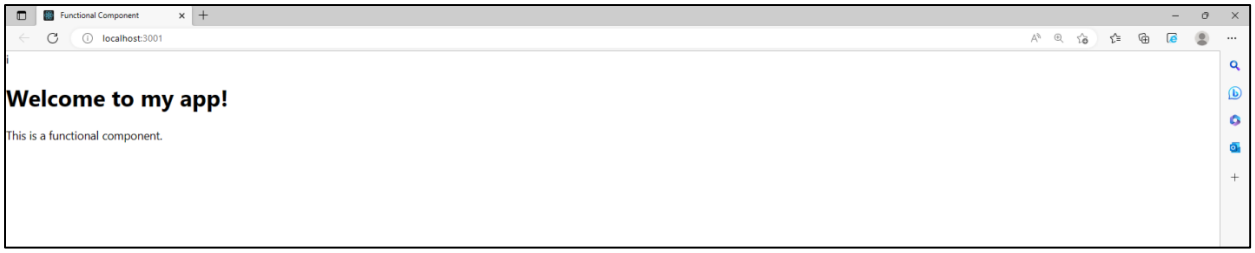
The Welcome component is then defined as a JavaScript function that returns a JSX element. This component renders a div element that contains a h1 and a p element, with some text content.

```
import logo from './logo.svg';
import './App.css';
import React from 'react';
function Welcome() {
 return (
  <div>
   <h1>Welcome to my app!</h1>
   <p>This is a functional component.</p>
  </div>
 );}
export default Welcome;
```

The last line exports the Welcome component using the export default syntax, which allows other components or modules to import and use it.

Index.js and index.html files are same as class component.

OUTPUT:



[Fig 4.3.1 Output of Functional Component]

4.4 React Props

In React, props (short for "properties") are a way to pass data from one component to another. They are a fundamental concept in React and are used to make components more reusable and modular.

A component's props are passed down from its parent component and can be accessed within the component using the props object. The props object contains all the properties that were passed to the component, and the component can then use these properties to render itself and its children.

Props can be used to pass data of any type, including strings, numbers, objects, arrays, and even functions. Props also used for print data in table and print same row 100 or many times.

Example: Print same row multiple times using props.

The **App** component simply renders a heading. The **Rowmultiple** component is defined next and accepts **props**. Within the **Rowmultiple** component, an empty array called **rows** is created. A **for** loop is used to loop through the range of 1 to 100, and in each iteration of the loop, a new **rowdata** component is pushed onto the **rows** array with props such as **id**, **name**, **email**, and **mobile**.

The **Rawdata** component is defined in the **propscomp.js** file, which accepts **props** and returns a table row with data passed in as props.

App.js

```
import logo from './logo.svg';
import './App.css';
import React from 'react';
import Rawdata from './propscomp';
function App() {
  return (
    <div>
      <h1>Print same row Multiple times</h1>
    </div>
  );
}
function Rowmultiple(props) {
  const rows = [];
  for (let i = 1; i <= 100; i++) {
    rows.push(
      <Rawdata
        key={i}
        id={i}
        name="Krinal"
        email="krinal@gamil.com"
        mobile="908756789"
      />
    );
  }
```

```
  return (
    <table border="1" width="100">
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>
          <th>Mobile</th>
        </tr>
      </thead>
      <tbody>
        {rows}
      </tbody>
    </table>
  );
}
export default App;
export {Rowmultiple};
```

[Fig 4.4.1 App.js file code for React Props]

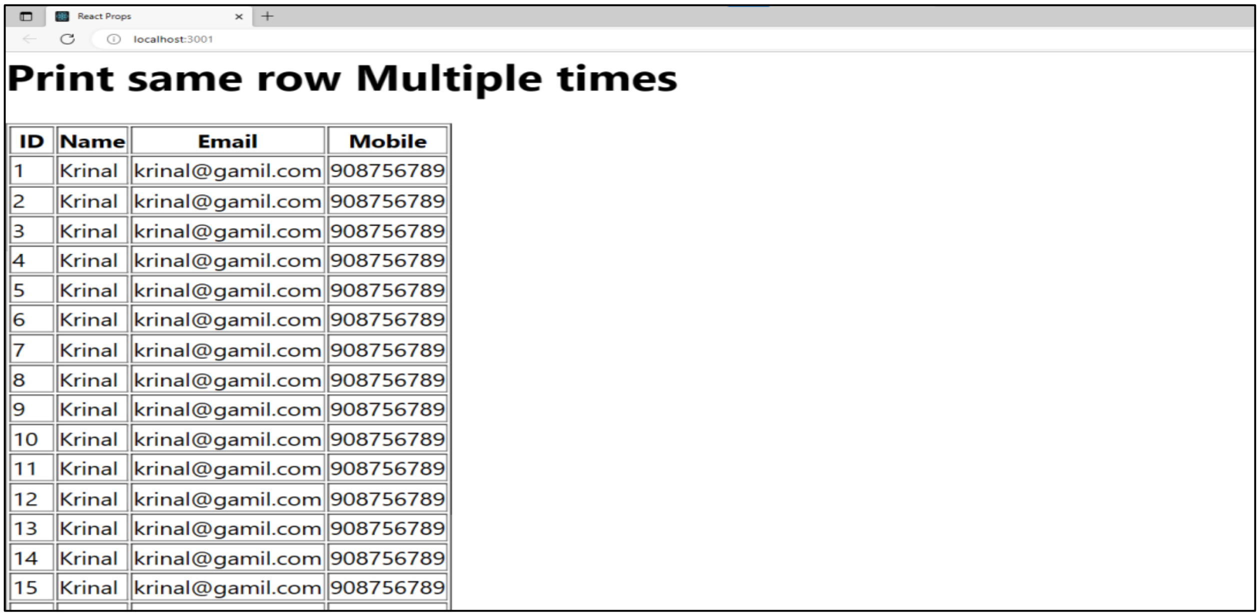
The **Rowmultiple** component renders a table with four columns: **ID**, **Name**, **Email**, and **Mobile**. It then renders the rows array within the **<tbody>** tag, which contains 100 rows of data, each with the same information.

Propscomp.js

```
function Rawdata(props){
    return(
    <tr>
        <td>{props.id}</td>
        <td>{props.name}</td>
        <td>{props.email}</td>
        <td>{props.mobile}</td>
    </tr>
    );
}
export default Rawdata;
```

[Fig 4.4.2 Props file code for React Props]

OUTPUT:



Print same row Multiple times

ID	Name	Email	Mobile
1	Krinal	krinal@gamil.com	908756789
2	Krinal	krinal@gamil.com	908756789
3	Krinal	krinal@gamil.com	908756789
4	Krinal	krinal@gamil.com	908756789
5	Krinal	krinal@gamil.com	908756789
6	Krinal	krinal@gamil.com	908756789
7	Krinal	krinal@gamil.com	908756789
8	Krinal	krinal@gamil.com	908756789
9	Krinal	krinal@gamil.com	908756789
10	Krinal	krinal@gamil.com	908756789
11	Krinal	krinal@gamil.com	908756789
12	Krinal	krinal@gamil.com	908756789
13	Krinal	krinal@gamil.com	908756789
14	Krinal	krinal@gamil.com	908756789
15	Krinal	krinal@gamil.com	908756789

[Fig 4.4.3 Output of React Props]

4.5 React Bootstrap

React Bootstrap is a front-end framework that combines the UI components of Bootstrap with the power of React, a popular JavaScript library for building user interfaces. It allows developers to quickly and easily create responsive, mobile-first web applications using pre-built components such as navigation bars, forms, buttons, alerts, modals, and more.

React Bootstrap is built using React's component-based architecture, which makes it easy to use and customize the individual components. It also includes features like server-side rendering, accessibility, and cross-browser compatibility.

Installation:

The best way to consume React-Bootstrap is via the npm package which you can install with npm.

```
npm install react-bootstrap bootstrap
```

Importing Components:

You should import individual components like: react-bootstrap/Button rather than the entire library.

```
import Button from 'react-bootstrap/Button' ;
```

CSS:

The following line can be included in your src/index.js or App.js file

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

React Bootstrap provides a wide range of UI features, including:

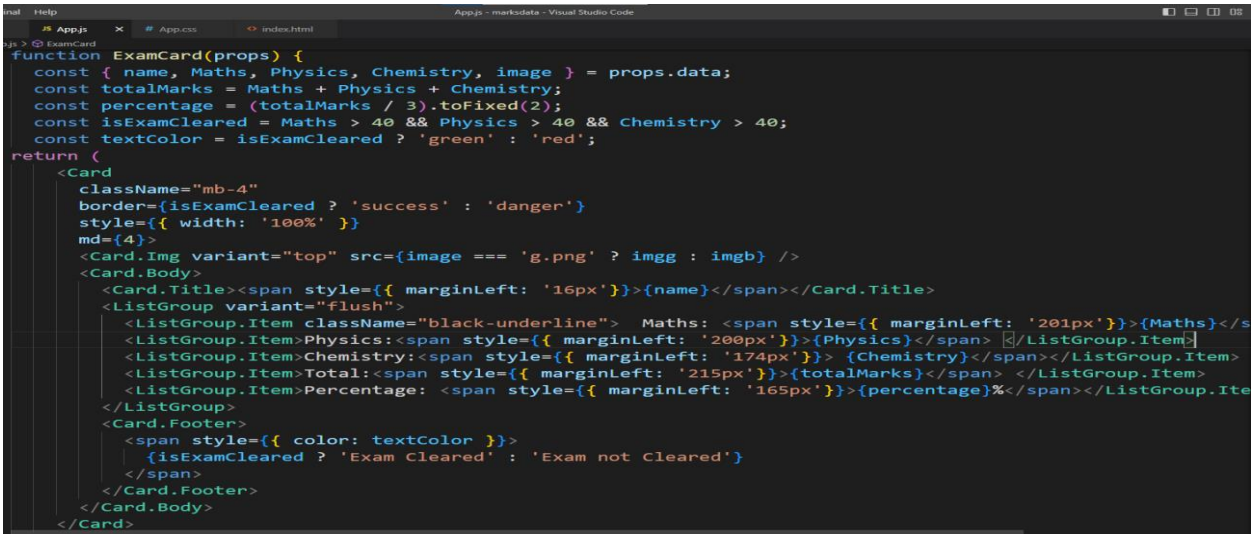
1. Navigation bars
2. Forms
3. Buttons
4. Dropdowns
5. Alerts
6. Modals
7. Tooltips
8. Progress bars
9. Spinners
10. Carousels
11. Accordion
12. Tabs
13. Cards
14. Tables
15. Pagination

16. Badges

17. Icons

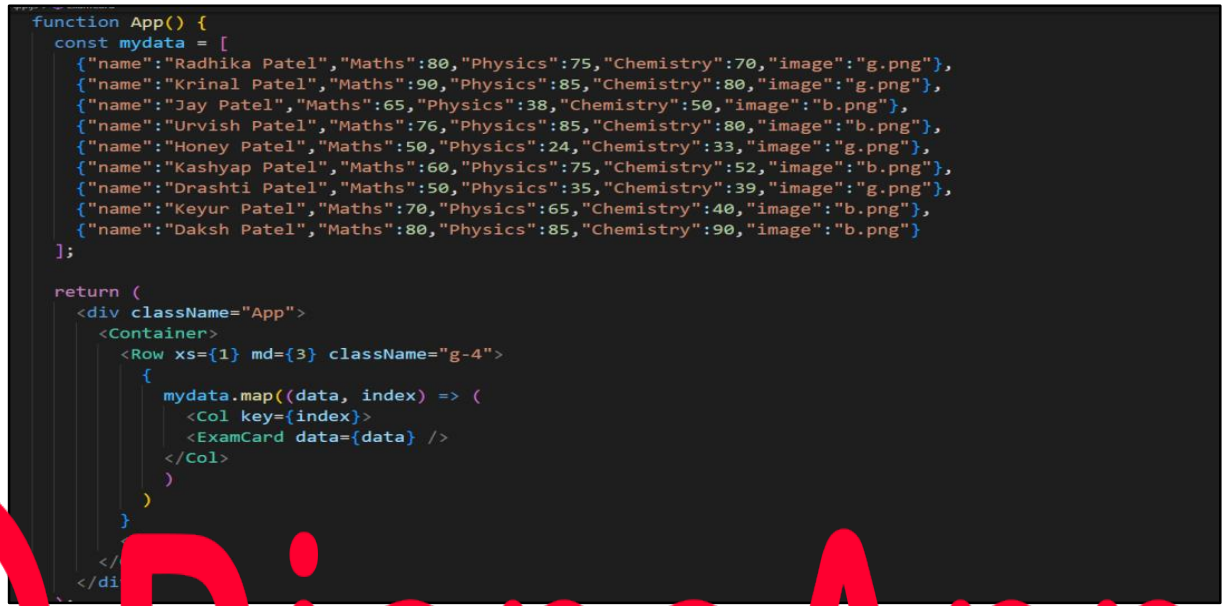
4.6 React Component for Displaying Exam Results in Cards using Bootstrap

This code is a React application that displays a list of exam cards. Each card represents the exam result of a student, including their name, scores in Maths, Physics, and Chemistry, total marks, percentage, and whether the student has cleared the exam or not.



```
function ExamCard(props) {
  const { name, Maths, Physics, Chemistry, image } = props.data;
  const totalMarks = Maths + Physics + Chemistry;
  const percentage = (totalMarks / 3).toFixed(2);
  const isExamCleared = Maths > 40 && Physics > 40 && Chemistry > 40;
  const textColor = isExamCleared ? 'green' : 'red';
return (
    <Card
      className="mb-4"
      border={isExamCleared ? 'success' : 'danger'}
      style={{ width: '100%' }}
      md={4}>
      <Card.Img variant="top" src={image === 'g.png' ? imgg : imgb} />
      <Card.Body>
        <Card.Title><span style={{ marginLeft: '16px'}}>{name}</span></Card.Title>
        <ListGroup variant="flush">
          <ListGroup.Item className="black-underline">  Maths: <span style={{ marginLeft: '201px'}}>{Maths}</s
          <ListGroup.Item>Physics:<span style={{ marginLeft: '200px'}}>{Physics}</span> </ListGroup.Item>
          <ListGroup.Item>Chemistry:<span style={{ marginLeft: '174px'}}> {Chemistry}</span></ListGroup.Item>
          <ListGroup.Item>Total:<span style={{ marginLeft: '215px'}}>{totalMarks}</span> </ListGroup.Item>
          <ListGroup.Item>Percentage: <span style={{ marginLeft: '165px'}}>{percentage}%</span></ListGroup.Ite
        </ListGroup>
        <Card.Footer>
          <span style={{ color: textColor }}>
            {isExamCleared ? 'Exam Cleared' : 'Exam not Cleared'}
          </span>
        </Card.Footer>
      </Card.Body>
    </Card>
```

[Fig 4.6.1 Code for cards]

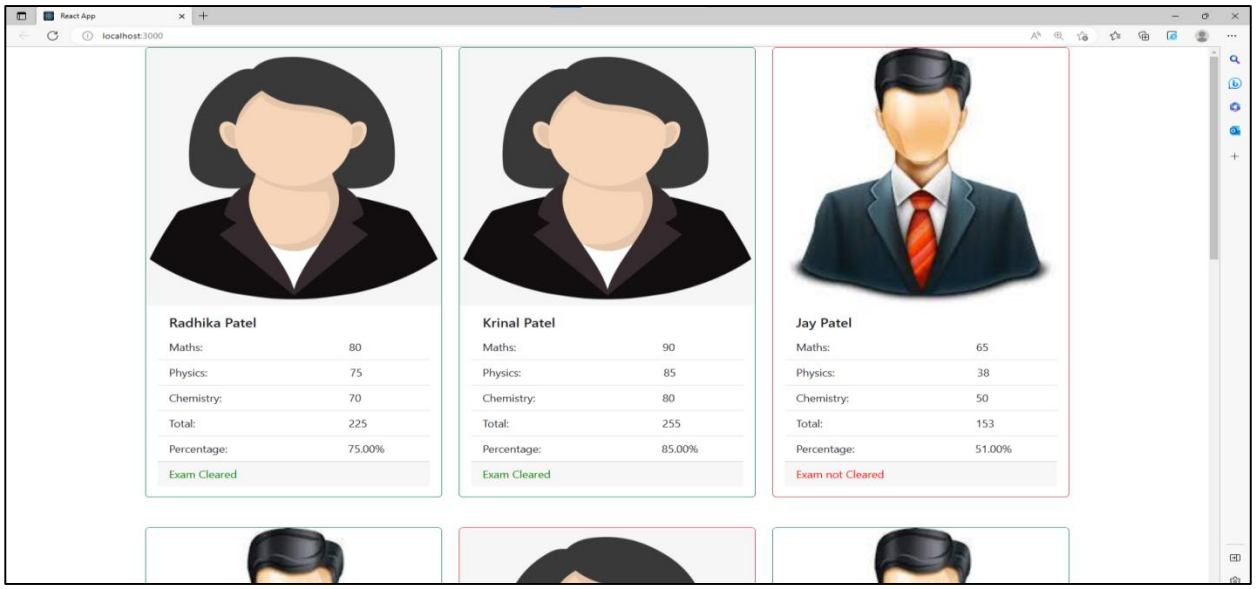


```
function App() {
  const mydata = [
    {"name":"Radhika Patel","Maths":80,"Physics":75,"Chemistry":70,"image":"g.png"},
    {"name":"Krinal Patel","Maths":90,"Physics":85,"Chemistry":80,"image":"g.png"},
    {"name":"Jay Patel","Maths":65,"Physics":38,"Chemistry":50,"image":"b.png"},
    {"name":"Urvish Patel","Maths":76,"Physics":85,"Chemistry":80,"image":"b.png"},
    {"name":"Honey Patel","Maths":50,"Physics":24,"Chemistry":33,"image":"g.png"},
    {"name":"Kashyap Patel","Maths":60,"Physics":75,"Chemistry":52,"image":"b.png"},
    {"name":"Drashti Patel","Maths":50,"Physics":35,"Chemistry":39,"image":"g.png"},
    {"name":"Keyur Patel","Maths":70,"Physics":65,"Chemistry":40,"image":"b.png"},
    {"name":"Daksh Patel","Maths":80,"Physics":85,"Chemistry":90,"image":"b.png"}
  ];

  return (
    <div className="App">
      <Container>
        <Row xs={1} md={3} className="g-4">
          {
            mydata.map((data, index) => (
              <Col key={index}>
              <ExamCard data={data} />
            </Col>
            )
          )
        }
```

[Fig 4. o cod cards]

The application uses React Bootstrap components to style and layout the exam cards. The **ExamCard** function is responsible for rendering a single card, while the **App** function renders the entire list of cards.

OUTPUT:



[Fig 4.6.3 Output of Student Data Card]

In the **ExamCard** function, the props are destructured to extract the necessary data such as the name, scores, and image. The total marks and percentage are computed from the scores. The function also determines whether the student has cleared the exam by checking if their scores in each subject are above 40. If the student has cleared the exam, the card's border color is set to green; otherwise, it is set to red.

In the **App** function, an array of student data is defined and passed to the **ExamCard** component as props. The **map** method is used to iterate through the data array and generate a card for each student. The cards are laid out in a grid using the **Row** and **Col** components from React Bootstrap.

Overall, this application demonstrates the use of React components and Bootstrap styling to create a dynamic and responsive user interface.

4.7 Data map

data.map() function is used to iterate over each element in the **data** array and return a new array with transformed elements. In this case, it returns a new array of **<h1>** elements, each containing an element from the **data** array.

Example:

```
const data = [10,20,30,40,50,60];
function Datashow(){
  return(
    <div>
      {
        data.map(
          (mydata) => {
            return <h1>{mydata}</h1>
          }
        )

      }
    </div>
  );
}
```

[Fig 4.7.1 Data Map Code]

Here's an explanation of how the **data.map()** function works:

- **data**: the original array to be mapped over
- **.map()**: a built-in JavaScript method for iterating over an array
- **(mydata) => {...}**: an arrow function that takes a single argument, **mydata**, which represents the current element being processed in the iteration
- **return <h1>{mydata}</h1>**: returns a new **<h1>** element with the **mydata** value interpolated into it
- **data.map(...)**: returns a new array of **<h1>** elements with each element from the **data** array

In summary, the **data.map()** function is a useful method for transforming array elements and returning a new array. It is often used in React to dynamically render components based on data.

4.8 Object map

The **Object.keys(data)** method is used to get an array of keys from the **data** object, and then the **map()** method is used to iterate over each key and generate a heading element that displays the key and its corresponding value.

Example: This code defines a React component called **Mutualfund**, which renders a table of mutual fund data. The mutual fund data is stored in an array called **mutualdata**, which contains objects with properties **schemeCode** and **schemeName**.

The **Mutualfund** component uses the **map()** method to loop through the **mutualdata** array and render a row for each object in the array. For each object, a **<tr>** element is created with two **<td>** elements inside it, one for the **schemeCode** and one for the **schemeName**.

```
import logo from './logo.svg';
import './App.css';
import 'bootstrap/dist/css/bootstrap.min.css';
import Table from 'react-bootstrap/Table';

function App() {
  return (
    <div className="App">
      <h1>Mutual Fund Data</h1>
    </div>
  );
}
const mutualdata = [{"schemeCode":100027,"schemeName":"Grindlays Super Saver Income Fund-GSSIF-Half Yearly
Dividend"},{"schemeCode":100028,"schemeName":"Grindlays Super Saver Income Fund-GSSIF-Quaterly Dividend"},
{"schemeCode":100029,"schemeName":"Grindlays Super Saver Income Fund-GSSIF-Growth"},{"schemeCode":100030,
"schemeName":"Grindlays Super Saver Income Fund-GSSIF-Annual Dividend"},{"schemeCode":100031,
"schemeName":"Grindlays Super Saver Income Fund-GSSIF - ST-Dividend"},{"schemeCode":100032,
"schemeName":"Grindlays Super Saver Income Fund-GSSIF - ST-Growth"},{"schemeCode":100033,
"schemeName":"Aditya Birla Sun Life Equity Advantage Fund - Regular Growth"},{"schemeCode":100034,
"schemeName":"Aditya Birla Sun Life Equity Advantage Fund -Regular Dividend"},{"schemeCode":100034,
"schemeName":"Aditya Birla Sun Life Equity Advantage Fund -Regular - IDCW"},{"schemeCode":100035,
```

[Fig 4.8.1 Stored data for Object Map]

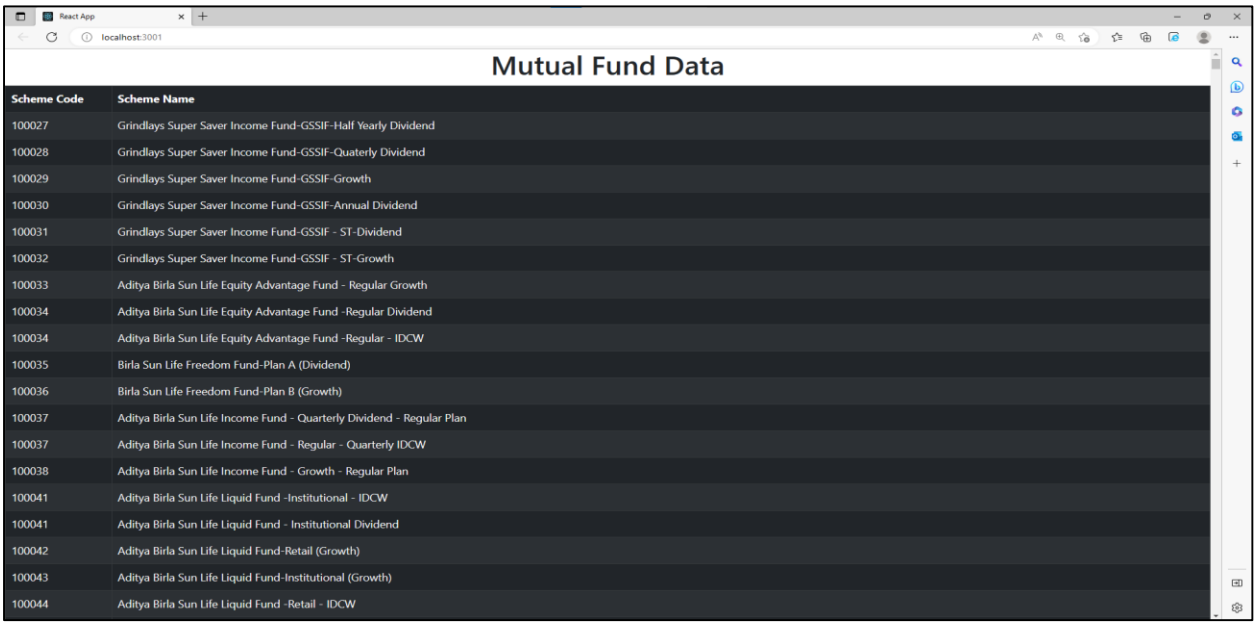
```
"schemeName":"HDFC FMP 2638D February 2023 - Growth Option - Direct Plan"}];
function Mutualfund(){
  return(
<div>
<Table striped bordered hover variant="dark">
          <thead>        <tr>
            <th>Scheme Code</th>
            <th>Scheme Name</th>
          </tr>
          </thead>
          <tbody>
  {
  mutualdata.map(
    (mdata) =>(
    // <h1 key={mdata.schemeCode}>{mdata.schemeCode}{mdata.schemeName}</h1>
    <tr><td>{mdata.schemeCode}</td><td>{mdata.schemeName}</td></tr>
    )
  )
}
  </tbody>
  </Table>
  div>
```

[Fig 4.8.2 Table display for Object Map code]

The **<Table>** element is a Bootstrap component that adds some styling to the table. It has a striped, bordered and hover effect on the table rows. The table has a header row that contains the column names "Scheme Code" and "Scheme Name".

When the **Mutualfund** component is rendered, it will display the table with the mutual fund data in it.

OUTPUT:



[Fig 4.8.3 Output of Table]

4.9 Map Props

A functional component called Share that renders a table displaying information about different stock prices. The stock data is stored in an array called shareinfo, which contains objects with information about the stock name, current market price (CMP), previous closing price (prev_close), and group.The table has six columns: Sr No, Stock Name, Current Price (CMP), Movement, Change in %, and Prev close. Each row in the table displays information for one stock.

T **share** a is mapped over using the **map** function, wh generates a new array of JSX ele ents t be ren red n th Ins map on, fo ea obje are ew ow s crea d wit he ev t dat fro that bje

```
const shareinfo = [
  {"stock_name" :"Reliance","cmp":"2500","prev_close":"2400","group":"A"},
  {"stock_name" :"HDFC","cmp":"2200","prev_close":"2500","group":"B"},
  {"stock_name" :"TATA","cmp":"5500","prev_close":"5467","group":"A"},
  {"stock_name" :"ICICI","cmp":"8769","prev_close":"9000","group":"A"},
  {"stock_name" :"Lenskart","cmp":"3500","prev_close":"5500","group":"A"},
  {"stock_name" :"Nirma","cmp":"2200","prev_close":"2500","group":"B"},
  {"stock_name" :"Sugar","cmp":"6700","prev_close":"6000","group":"A"},
  {"stock_name" :"Mamaearth","cmp":"4500","prev_close":"4000","group":"B"},
  {"stock_name" :"Flipkart","cmp":"20567","prev_close":"15000","group":"A"},
  {"stock_name" :"Amazon","cmp":"18500","prev_close":"15000","group":"B"},
]
function Share() {
  return (
    <div>
      <Table striped>
        <thead>
          <tr>
            <th> Sr No</th>
            <th>Stock Name</th>
            <th>Currenr Price</th>
            <th>Movement</th>
            <th>Change in %</th>
            <th>Prev close</th>
```

[Fig 4.9.1 Stored Data for Map Props]

The **movement** variable is calculated as the difference between the current market price **(value.cmp)** and the previous closing price **(value.prev_close).** If the movement is positive, it is displayed in green; if negative, it is displayed in red.

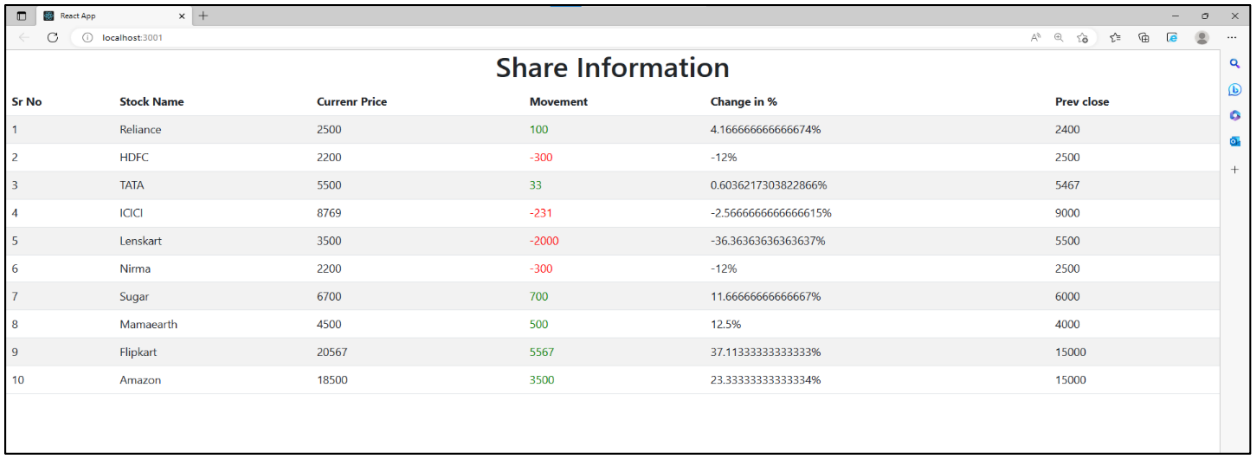
```
          </thead>
          <tbody>
            {
            shareinfo.map(
              (value, index) => {
              const movement = value.cmp - value.prev_close;
              const percentageChange = ((value.cmp / value.prev_close) - 1) * 100

              return (
                <tr>
                  <td>{index + 1}</td>
                  <td>{value.stock_name}</td>
                  <td>{value.cmp}</td>
                  <td style={{color: movement >= 0 ? 'green' : 'red'}}>{movement}</td>
                  <td>{percentageChange}%</td>
                  <td>{value.prev_close}</td>
                </tr>
              );
            }
          )
        }
          </tbody>
        </Table>
      </div>
    );
  }
```

[Fig 4.9.2 Map Code for Map Props]

The **percentageChange** variable is calculated as the percentage change from the previous closing price to the current market price. It is displayed with a % symbol. Finally, the **return**

statement renders the table with the generated rows, using the **Table** component from the React Bootstrap library.

OUTPUT:



[Fig 4.9.3 Output of Map Props]

4.10 React Hooks

React Hooks is a feature introduced in React version 16.8. Hooks allow functional components to use state and other React features without having to convert them into class components.

The most commonly used React Hooks are:

- **useState:** This Hook is used to manage state in a functional component. It returns an array with two elements, the current state and a function to update the state.
- **useEffect:** This Hook is used to handle side effects in functional components, such as fetching data from an API or updating the DOM. It takes two arguments, a function to run on component mount and a dependency array to track changes that may cause the function to run again.
- **useContext:** This Hook is used to access a context object created by a Provider com[illegible]t. It ta[illegible]s a context object as an argument and re[illegible]rns its current value.
- **use[illegible]ef:** Th[illegible] H[illegible]k is[illegible]o cr[illegible]muta[illegible]ferenc[illegible]to[illegible] DOM[illegible]ent [illegible]a[illegible]. It ret[illegible]ns a re[illegible] obj[illegible]t th[illegible] can[illegible] at[illegible]ched[illegible]o a[illegible]lem[illegible]t.

- **useCallback:** This Hook is used to memoize a function so that it only re-renders when its dependencies change. It takes a function and an array of dependencies as arguments and returns a memoized function.
- **useMemo:** This Hook is used to memoize a value so that it only re-computes when its dependencies change. It takes a function and an array of dependencies as arguments and returns a memoized value.

Hooks provide a cleaner and more functional programming approach in React. They allow for more flexible and reusable code and make it easier to manage component state and lifecycle.

4.11 React Modals

A React application that displays sports news articles in a responsive grid layout using Bootstrap.

The news articles are stored in the **newsdata** object which contains a category property and a data property that is an array of objects representing individual news articles.

The application uses React hooks, specifically the **useState** hook, to maintain the state of the modal windows that display the full article content when the "Read more" button is clicked. The **handleShow** function is called when the button is clicked, and it sets the modal state for the corresponding article to true. The **handleClose** function is called when the modal window is closed, and it sets the modal state for the corresponding article to false.

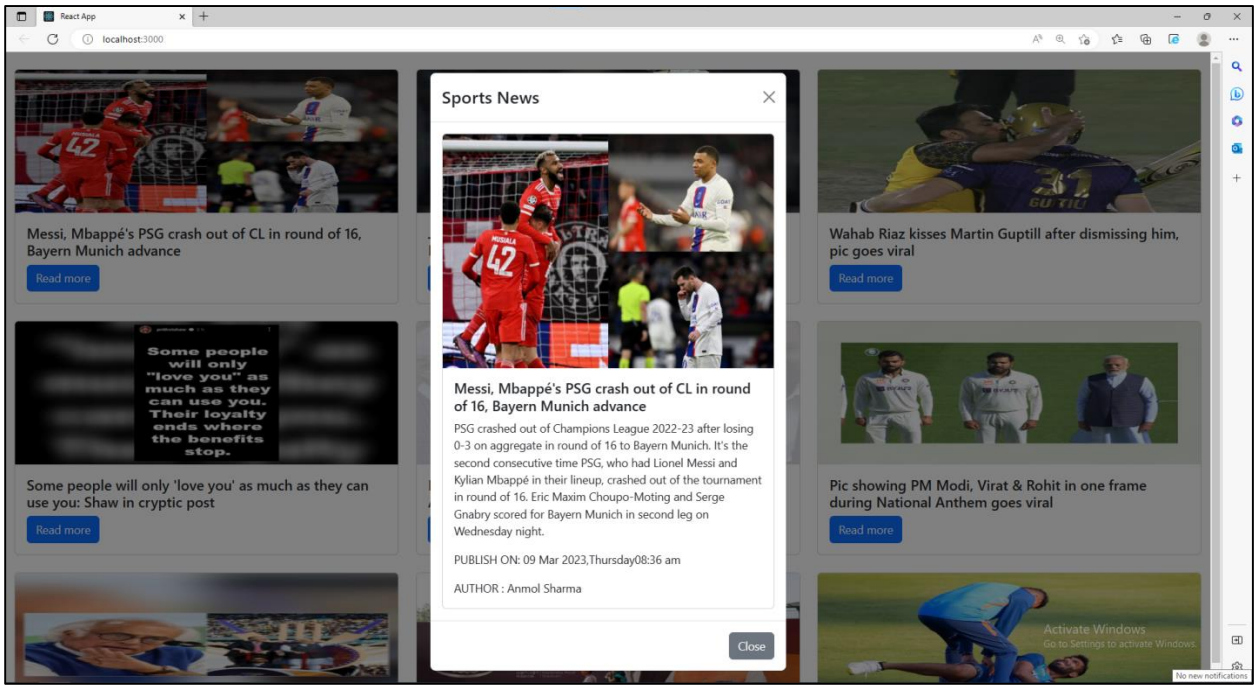
```
function App() {
  const [modalStates, setModalStates] = useState(newsdata.data.map(() => false));
  const handleClose = (index) => {
    const newStates = [...modalStates];
    newStates[index] = false;
    setModalStates(newStates);
  };
  const handleShow = (index) => {
    const newStates = [...modalStates];
    newStates[index] = true;
    setModalStates(newStates);
  };
  return (
    <Container fluid>
      <Row xs={1} md={3} classname = "g-2">
        {
          newsdata.data.map(
            (value,index)=>{
              return(
                <Col className="container-fluid mt-4">
                <Card key ={index} >
                <Card.Img variant="top" src={value.imageUrl} width="100%" height="200px" />
                <Card.Body>
                <Card.Title>{value.title}</Card.Title>
                <Button variant="primary" onClick={() => handleShow(index)
                <Modal show={modalStates[index]} onHide={() => handleClose
                <Modal.Header closeButton>
                <Modal.Title>Sports News</Modal.Title>
```

[Fig 11 Code for React Modals]

```
                <Row xs={1} md={1} className="g-4">
                  <Col className="container-fluid mt-4">
                    <Card>
                      <Card.Img variant="top" src={value.imageUrl} />
                      <Card.Body>
                        <Card.Title>{value.title}</Card.Title>
                        <Card.Text>
                          {value.content}
                        </Card.Text>
                        <Card.Text>
<span className="font-weight-bold">PUBLISH ON:</span> {value.date}{value.time}
                        </Card.Text>
                        <Card.Text>
                          AUTHOR : {value.author}
                        </Card.Text>
                      </Card.Body>
                    </Card>
                  </Col>
                </Row>
                </Modal.Body>
                <Modal.Footer>
                <Button variant="secondary" onClick={() =>handleClose(index)}>
                  Close
                </Button>
                </Modal.Footer>
                </Modal>
                </Card.Body>
                </Card>
                </Col>
```

[Fig 4.11.2 Code for React Modals]

OUTPUT:



[Fig 4.11.3 Output of React Modals]

4.12 Live API data fetch

To **fetch live API** data in ReactJS, you can use the fetch method, which is a built-in function in modern browsers that allows you to make HTTP requests. The **fetch** method returns a Promise that resolves to the Response object representing the response to the request.

To use **fetch** in a React component, you can create a new function that calls the API and updates the component's state with the response data. You can use the **useState** hook to define state variables for the data you want to fetch, and the **useEffect** hook to fetch the data and update the state whenever the component mounts or when the data changes.

Here is an example of how to use **fetch** to fetch **Bitcoin** price data from the **Coindesk API** and display it in a React component:

A React application that fetches the current Bitcoin prices in USD, GBP, and EUR from the Coindesk API and displays them in Bootstrap cards.

The App component simply renders a header that says "Bitcoin Price".

The FetchAPI component is where most of the functionality is. It uses the useState and useEffect hooks provided by React to manage state and handle side effects, respectively. The useState hook creates three state variables: usdRate, gbpRate, and eurrate, all initialized to 0. These variables will be used to store the current Bitcoin prices in USD, GBP, and EUR, respectively.

```
import logo from './logo.svg';
import './App.css';
import { useState, useEffect } from "react";
import 'bootstrap/dist/css/bootstrap.min.css';
import Card from 'react-bootstrap/Card';
import { Container, Row, Col } from 'react-bootstrap';
import './FetchAPI.css';
export default function App() {
  return (
    <div className="App" style={{ backgroundColor:"darkgrey", padding: "20px" }}>
    <h1 style={{ color: "darkblue", fontSize: "48px", textAlign: "center" }}>
      Bitcoin Price
    </h1>
  </div>
  );
}
export function FetchAPI() {
  const [usdRate, setUsdRate] = useState({});
  const [gbpRate, setGbpRate] = useState({});
  const [eurRate, setEurRate] = useState({});
  const apiGet = () => {
    fetch('https://api.coindesk.com/v1/bpi/currentprice.json')
      .then((response) => response.json())
      .then((json) => {
        setUsdRate(json.bpi.USD);
        setGbpRate(json.bpi.GBP);
```

Fig 4.12.1 Code for Live Bitcoin API data fetch]

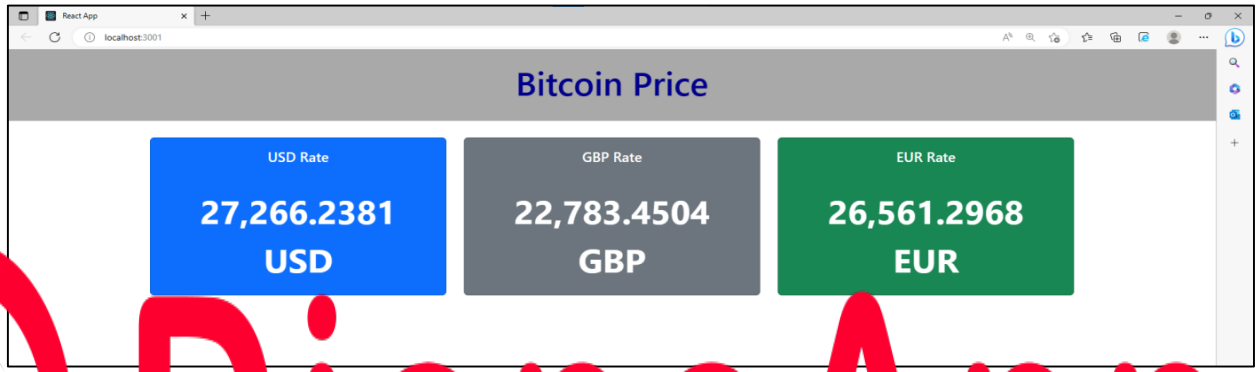
The useEffect hook is used to fetch the data from the Coindesk API and update the state variables every second. It calls the apiGet function, which fetches the data and updates the state variables with the new prices. The useEffect hook also returns a cleanup function that clears the interval timer when the component is unmounted.

The fetch function is used to make a GET request to the Coindesk API endpoint for the current Bitcoin prices. The response is in JSON format, which is parsed and used to update the state variables with the current prices.

```
  useEffect(() => {
  apiGet();
  const interval = setInterval(apiGet, 1000);
  return () => clearInterval(interval);
}, []);
  return (
  <Container className="mt-4">
    <Row className="row-cols-1 row-cols-md-3 g-4">
      <Col>
        <Card bg="primary" text="white" className="text-center h-100">
          <Card.Body>
            <Card.Title>USD Rate</Card.Title>
            <Card.Text className="rate">{usdRate.rate} {usdRate.code}</Card.Text>
          </Card.Body>
        </Card>
      </Col>
      <Col>
        <Card bg="secondary" text="white" className="text-center h-100">
          <Card.Body>
            <Card.Title>GBP Rate</Card.Title>
            <Card.Text className="rate">{gbpRate.rate}{gbpRate.code}</Card.Text>
          </Card.Body>
        </Card>
      </Col>
      <Col>
        <Card bg="success" text="white" className="text-center h-100">
```

[Fig 4.12.2 Code for Live Bitocoin API data fetch]

OUTPUT:



4.12 ut of in Da

4.13 Web Page of Covid-19 data

React web application that displays **COVID-19** data for India. The application fetches the data from an API and displays it in two sections - a summary section and a table section.

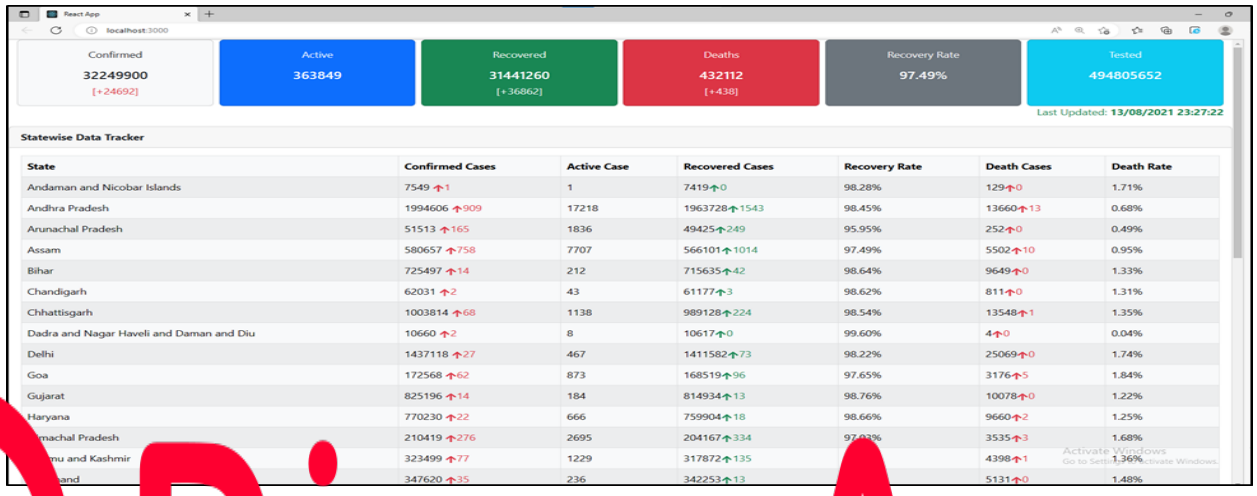
The data is fetched from an external API and stored in a variable called **mydata**. The mydata variable contains an object with two properties - **statewise** and **tested.**

The code uses the Bootstrap framework to style the cards and format the layout.

The table displays state-wise COVID-19 data, including the number of confirmed cases, active cases, recovered cases, deaths, recovery rate, and death rate.

```
function CovidTable() {
  return (
    <Card>
     <Card.Header style={{ fontWeight: 'bold' }}>Statewise Data Tracker</Card.Header>
       <Card.Body>
       <Table striped bordered hover variant="light">
         <thead>
           <tr>
             <th>State</th>
             <th>Confirmed Cases</th>
             <th>Active Case</th>
             <th>Recovered Cases</th>
             <th>Recovery Rate</th>
             <th>Death Cases</th>
             <th>Death Rate</th>
           </tr>
         </thead>
         <tbody>
           {
           mydata.statewise.slice(1).map((data) => (
             <tr>
               <td>{data.state}</td>
               <td>
                 {data.confirmed}{' '}
                 <span className="text-danger">
                     <FaArrowUp />
                     {data.deltaconfirmed}
                   </span>
```

[Fig. 4.13.1 Code for Covid-19 web Page]



[Fig. 4.13.2 Output for Covid-19 Page]

4.14 What is React Routing?

React Routing is a library used in React applications to manage the navigation and rendering of different components based on the current URL or user interactions. It allows you to create single-page applications (SPAs) where the content is dynamically loaded and updated without refreshing the entire page.

In React, the most popular routing library is React Router. It provides a set of components and utilities that enable you to define routes, map them to specific components, and handle navigation between them. The key component in React Router is the <BrowserRouter>, which wraps your application and provides the necessary functionality for routing.

The main component used for defining routes is <Route>. It specifies a path and the corresponding component to render when that path matches the current URL. For example, <Route path="/about" component={About} /> will render the About component when the URL matches "/about". You can also use the exact prop to ensure an exact match.

React Router also provides other components like <Switch>, <Link>, and <Redirect> to enhance the routing experience. <Switch> is used to render only the first <Route> that matches the current URL, preventing multiple components from rendering simultaneously. <Link> is used to create navigation links that can be clicked to update the URL and trigger a route change. <Redirect> is used to redirect the user to a different URL.

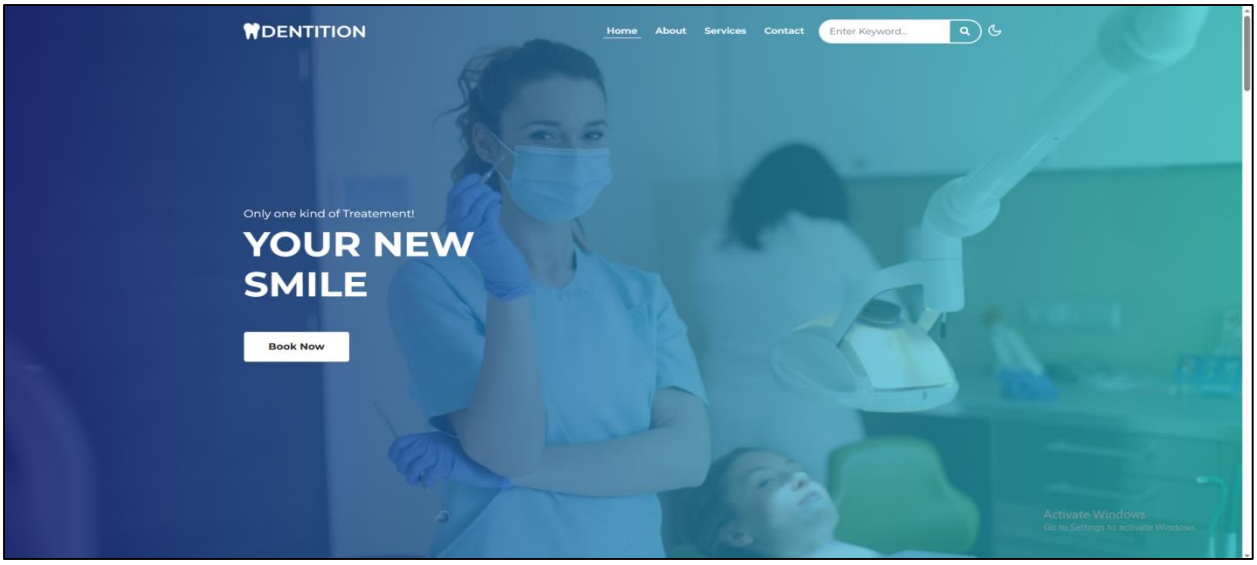
Additionally, React Router allows you to pass URL parameters and query strings as part of the route path, enabling dynamic rendering based on user input or data. It also supports nested routes and provides hooks and APIs for programmatic navigation and accessing route-related information.

4.15 Webpage with Multiple Pages With Routing

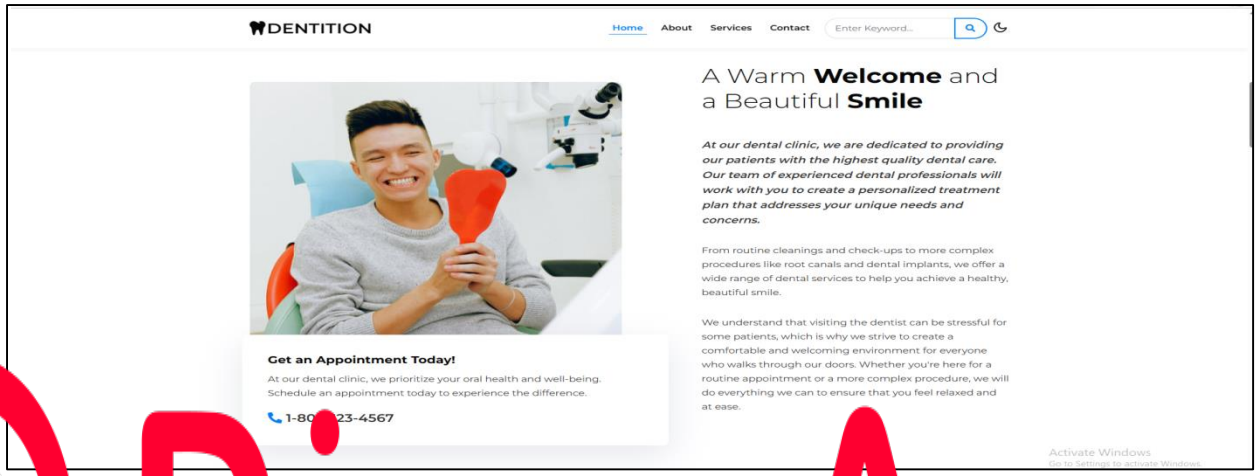
Input:

[Fig. 4.15.1 Routing in Dentist Template]

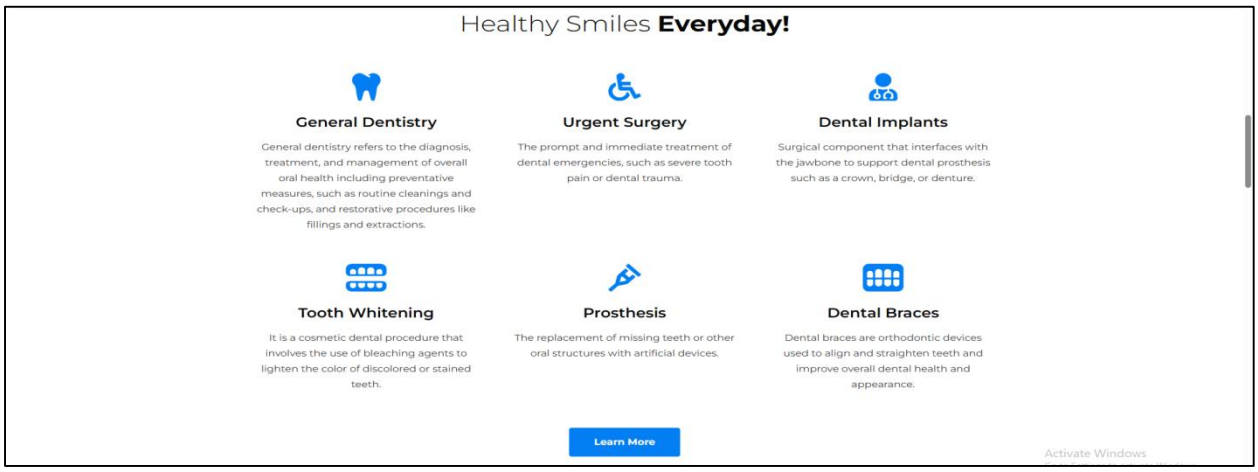
OUTPUT:



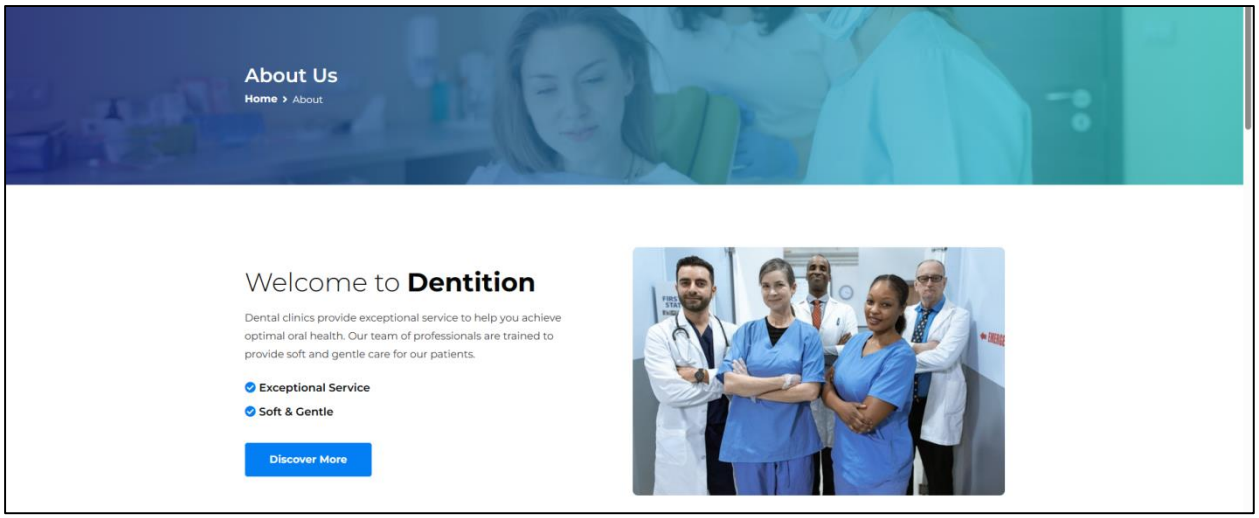
[Fig. 4.15.2 Home Page of Dentist Template]



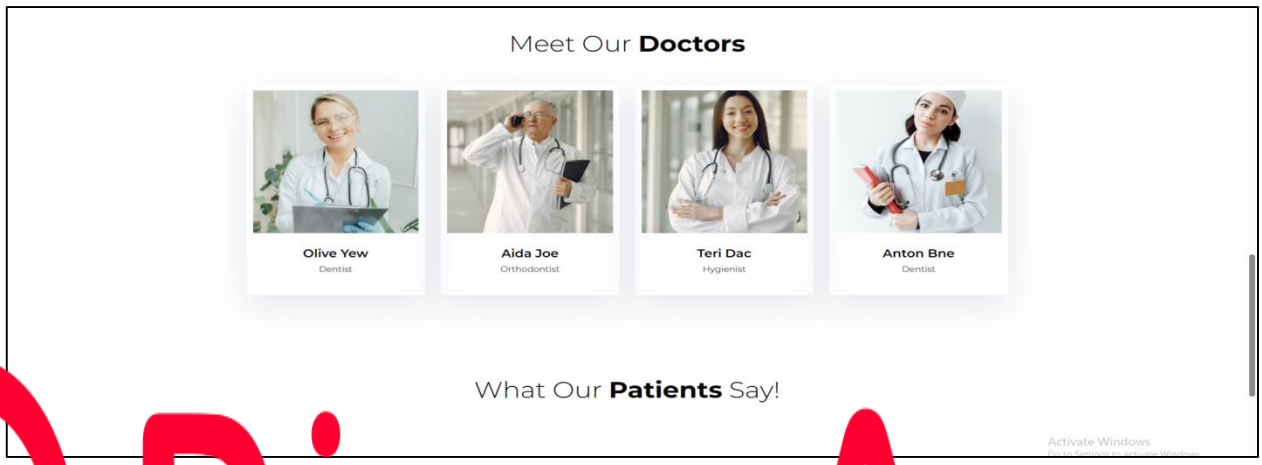
[Fig. 4.15.3 Home Page of Dentist Template]



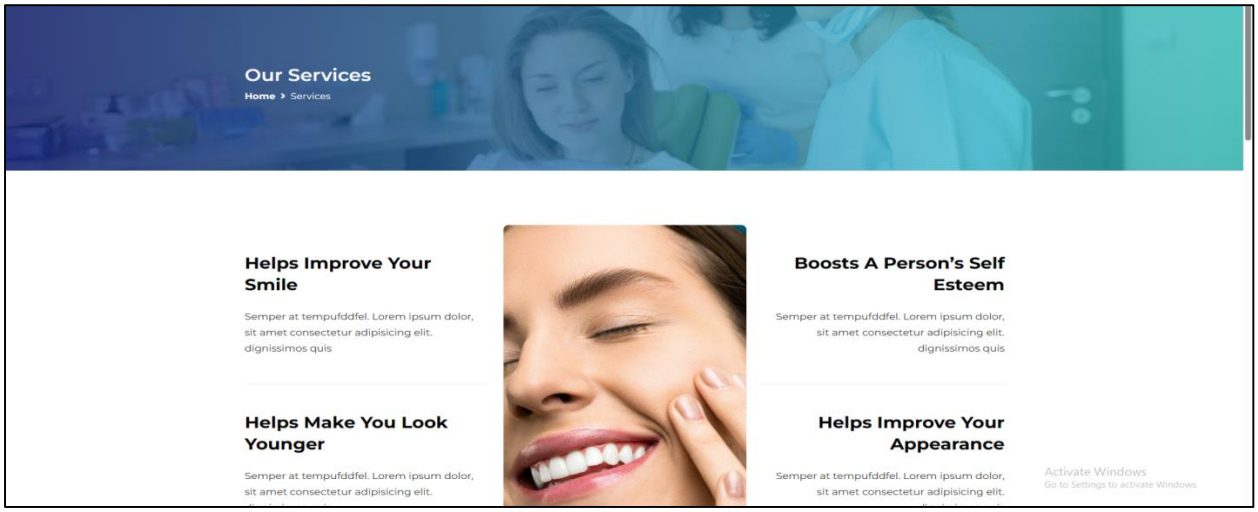
[Fig. 4.15.4 Home Page of Dentist Template]



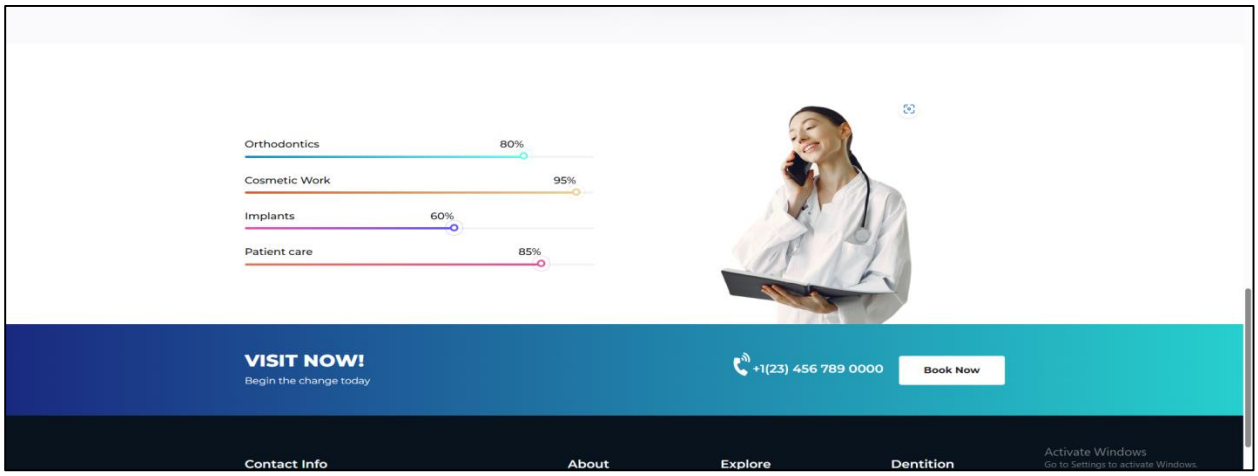
[Fig. 4.15.5 About Page of Dentist Template]



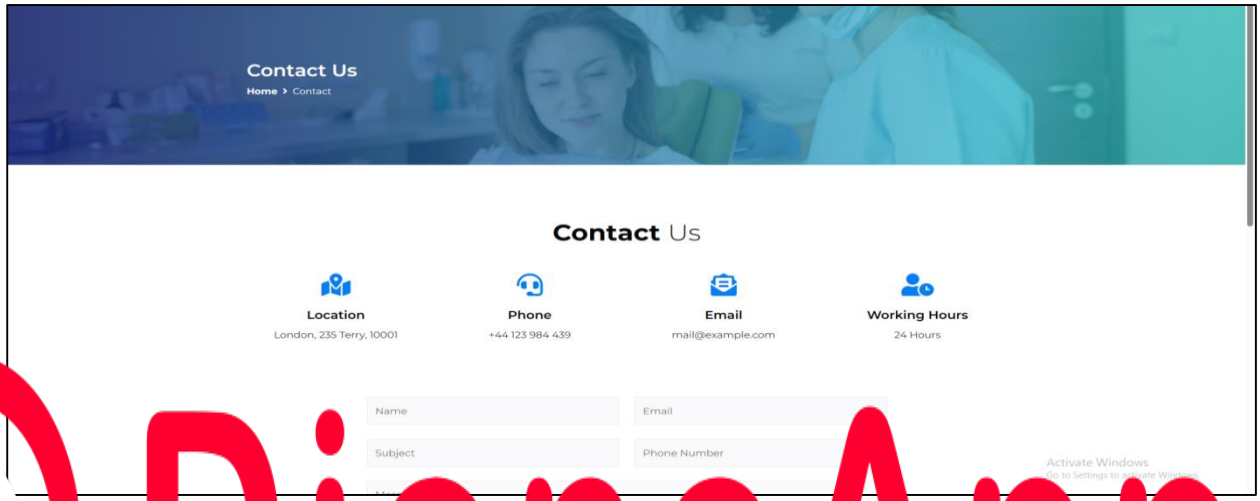
[Fig. 4.15.6 About Page of Dentist Template]



[Fig. 4.15.7 Service Page of Dentist Template]



[Fig. 4.15.8 Service Page of Dentist Template]



[Fig. 4.15.9 Contact Page of Dentist Template]

5. INTRODUCTION TO PROJECT

5.1 Project Summary

The React Dashboard and Analytics Platform is a powerful tool for businesses that provides real-time insights and analytics in an easy-to-use dashboard. It is built using React and can integrate with various data sources to provide a complete view of a business's performance metrics and key indicators.

The platform allows businesses to quickly identify areas of improvement and track progress towards their goals by providing up-to-date information and trends. It also offers advanced analytics capabilities, such as data visualization and predictive analytics, to help businesses gain deeper insights into their data and make informed decisions.

This web-based application is an admin template that is customizable and can be adapted to meet specific business needs. It is designed to help businesses stay ahead of the competition by providing them with the necessary insights and data to make data-driven decisions and achieve their objectives.

5.2 Purpose

The purpose of the React Dashboard and Analytics Platform project is to develop a comprehensive and user-friendly solution for businesses that provides real-time insights and analytics. By leveraging React technology and integrating with various data sources, this platform aims to centralize business metrics and KPIs to provide a complete view of a business's performance.

The platform's intuitive dashboard provides up-to-date information on key performance indicators and trends, enabling businesses to quickly identify areas of improvement and track progress towards their goals. It also offers advanced analytics capabilities such as data visualization and predictive analytics to help businesses gain deeper insights into their data and make informed decisions.

This project's ultimate goal is to provide businesses with a powerful and customizable admin template that can be adapted to meet specific business needs. By offering real-time updates and a user-friendly interface, the Dashboard and Analytics Platform aims to be an essential tool for businesses looking to stay ahead of the curve and make data-driven decisions. The purpose of this project is to create a platform that is reliable, efficient, and can help businesses achieve their objectives.

5.3 Objective

Provide businesses with a comprehensive tool for gaining insights into their key performance indicators and metrics.

Solve the problem of data fragmentation by integrating with various data sources.

Provide a centralized view of business metrics and KPIs.

Offer advanced analytics capabilities to allow businesses to make informed decision.

5.4 Scope

The scope of the React Dashboard and Analytics Platform project is to develop a web-based application that provides businesses with real-time insights and analytics. The platform will be built using React and will integrate with various data sources to provide a centralized view of business metrics and KPIs.

The project will include developing an intuitive dashboard that displays up-to-date information on key performance indicators and trends, as well as advanced analytics capabilities such as data visualization and predictive analytics.

The goal is to provide businesses with a user-friendly platform that enables them to make data-driven decisions and stay ahead of the competition. The project will also involve creating an admin template that is customizable to meet specific business needs.

5.5 Tech[illegible] and Literature Review

Frontend:

The Dashboard and Analytics Platform will be built using React, which is a popular frontend JavaScript library for building user interfaces. React offers numerous benefits, such as high performance, reusability of code components, and a large community of developers. The platform's intuitive dashboard will display up-to-date information on key performance indicators and trends, and will be designed using modern UI/UX principles to ensure a user-friendly experience.

Backend:

The Dashboard and Analytics Platform will require a backend to handle data processing and storage. The backend will integrate with various data sources, such as databases and APIs, to provide a centralized view of business metrics and KPIs.

Code Editor:

The code editor used for the Dashboard and Analytics Platform project will depend on the developer's preference. VS Code, a widely used code editor, will be used to write and edit the code for the Dashboard and Analytics Platform.VS Code provides useful features such as syntax highlighting, debugging, and extensions that can enhance the development process. These code editors offer numerous features, such as code highlighting, auto-completion, and debugging, to make development easier and more efficient. The code editor will be used to write and edit the frontend and backend code for the platform.

5.6 Project Planning

- First We Will discuss the definition of Project
- Choose the technology
- Then define the software requirement of the system
- Define the role in the system
- Draw the ER diagram and System flow diagram
- Define the module
- De e E n Entities
- Pla the U In rfac sys

6. SYSTEM ANALYSIS

6.1 Study of Current System

The current system for business dashboards reveals that many businesses face challenges in gaining real-time insights and analytics into their operations. The current system often relies on manual data collection and analysis, which can be time-consuming and error-prone. This leads to delays in decision-making and a lack of visibility into business performance.

Furthermore, the current system often lacks integration with various data sources, leading to a fragmented view of business metrics and key performance indicators (KPIs). This makes it difficult for businesses to gain a holistic view of their operations and identify areas of improvement.

The current system also has limited capabilities for data visualization and predictive analytics, making it challenging for businesses to gain deeper insights into their data and make informed decisions.

6.2 Problem and Weaknesses Of Current System

1. **Data fragmentation:** Existing systems may have data scattered across various sources, making it difficult to get a comprehensive view of business metrics and KPIs. The Dashboard and Analytics Platform can integrate with these sources and provide a centralized view of the data.
2. **Lack of real-time updates:** Traditional reporting systems may provide data on a periodic basis, which may not be sufficient for businesses that need to make quick decisions. The Dashboard and Analytics Platform provides real-time updates, enabling businesses to respond to changes in their metrics in real-time.
3. **Limited analytics capabilities:** Existing systems may not provide advanced analytics capabilities such as data visualization and predictive analytics, which are essential for

gaining deeper insights into the data. The Dashboard and Analytics Platform offers these capabilities, allowing businesses to make informed decisions based on their data.

6.3 Proposed System Requirement

Functional Requirements:

1. **Real-time data updates:** The platform should be able to display real-time updates on the dashboard, enabling businesses to make informed decisions in a timely manner.
2. **Integration with multiple data sources:** The platform should be able to integrate with various data sources, providing a centralized view of business metrics and KPIs.
3. **User-friendly interface:** The platform should have an intuitive and user-friendly interface that is easy to navigate.
4. **Customizable dashboard:** The platform should allow users to customize their dashboard, adding or removing metrics based on their business needs.
5. **Advanced analytics capabilities:** The platform should provide advanced analytics capabilities, such as data visualization and predictive analytics, to help businesses gain deeper insights into their data.

Non-Functional Requirements:

1. **Performance:** The platform should be able to handle large amounts of data and provide real-time updates without any performance issues.
2. **Security:** The platform should have robust security measures in place to ensure the confidentiality and integrity of sensitive business data.
3. **Scalability:** The platform should be able to scale up or down based on the changing needs of the business.
4. **Compatibility:** The platform should be compatible with different browsers and operating systems.

5. **Usability:** The platform should be easy to use and understand, even for users with minimal technical expertise.

6.4 Function Of The System

- Customizable dashboard
- Data visualization
- Alert system
- Predictive analytics
- Trending Products
- Add Products
- Manage Products
- Orders
- User Management
- Feedback
- Contact us

6.5 Software Specification

Designing frontend	: Html , CSS, JavaScript, ES6, JSON
Framework	: Bootstrap framework
IDE	: Vs code
Backend	: 000Webhost, Postman
Browser	: Chrome, V8 Engine
Core Language	: ReactJs

7. SYSTEM DESIGN

7.1 System Design

Admin side:

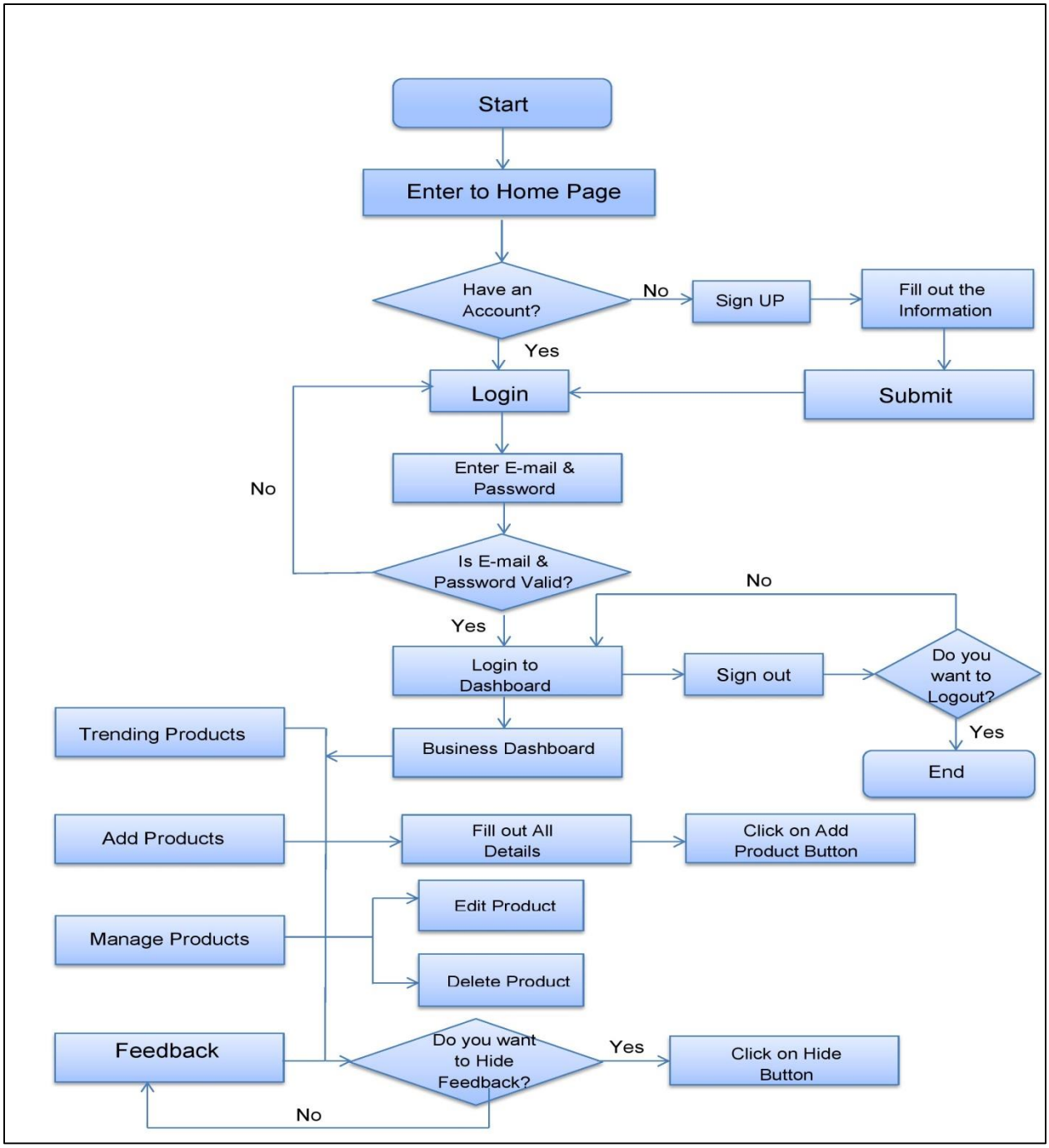
- User authentication and authorization system for secure access to the platform
- Data source integration to pull in data from various sources and store it in a centralized database
- Data processing and analysis system to generate real-time insights and analytics
- Customizable dashboard builder to create personalized views for different users and departments
- Tools for managing user accounts, permissions, and data sources
- System monitoring and maintenance tools for ensuring optimal performance and uptime

User side:

- User authentication and authorization system for secure access to the platform
- Dashboard with real-time updates on key performance indicators and trends
- Data visualization tools for deeper insights into data
- Predictive analytics capabilities for forecasting future trends
- Customizable views based on user preferences and roles
- Ability to export data and reports for further analysis and sharing

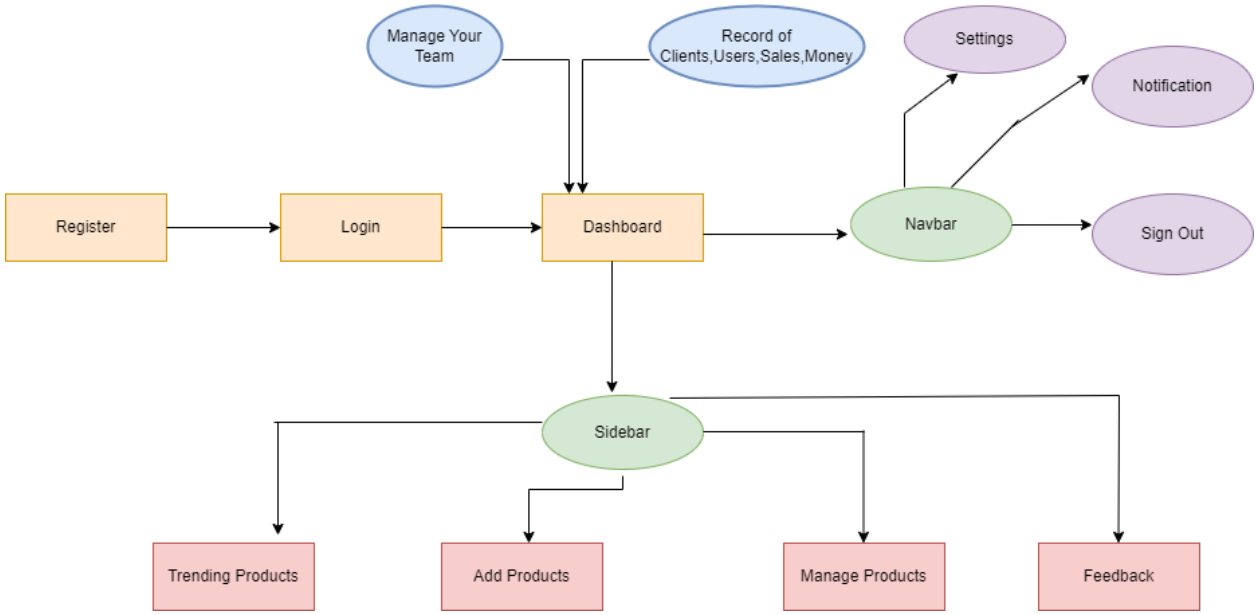
Both the admin and user sides will require a robust and scalable system architecture, with considerations for security, data privacy, and performance. The use of React for the front-end and API for the back-end will help to ensure a responsive and reliable platform for users.

7.2 System Flow Diagram



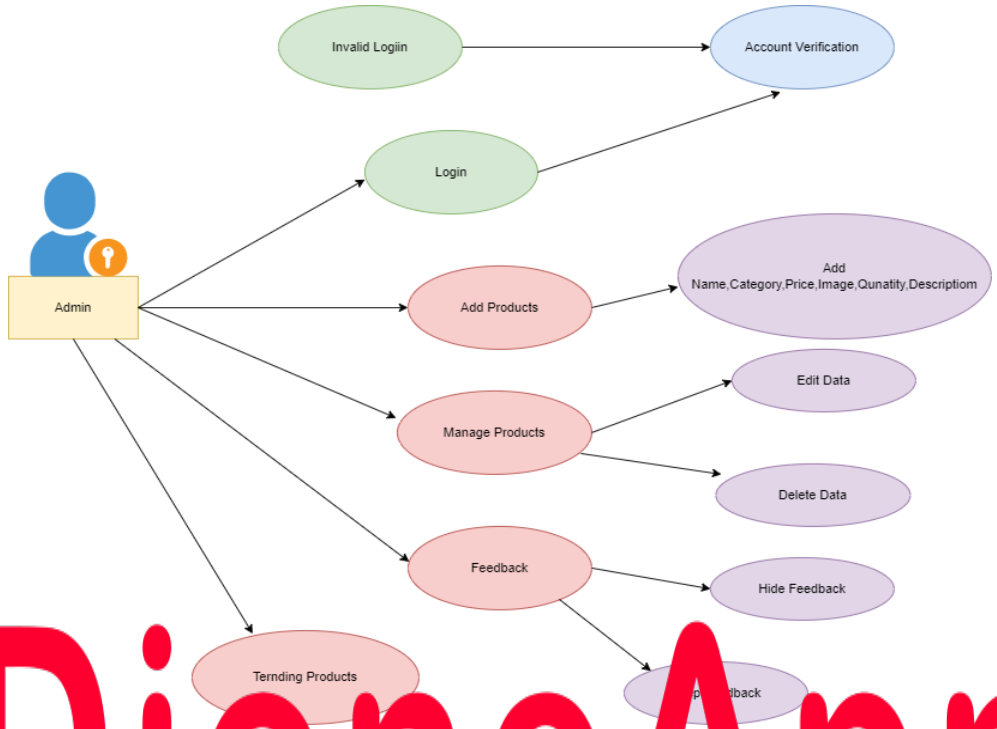
[Fig. 7.2.1 System Flow Diagarm]

7.3 ER Diagram



[Fig. 7.3.1 ER Diagram]

7.4 Use Case Diagram



[Fig. 7.4.1 Use case Diagram]

8. IMPLEMENTATION

8.1 Technologies and Implementation Environment

Designing frontend	: Html , CSS, JavaScript, ES6, JSON
Framework	: Bootstrap framework
IDE	: Visual Studio code
Backend	: 000Webhost, Postman
Version Control	:Github
Browser	: Chrome, V8 Engine
Core Language	: ReactJs

Visual Studio Code (VS Code) is a popular integrated development environment (IDE) used by many developers for coding. An IDE is a software application that provides a comprehensive development environment for software developers, including code editors, debugging tools, build automation, and other features to aid in the development process.

VS Code supports a wide range of programming languages including JavaScript, Python, C#, Java, HTML, CSS, and many others. This makes it a popular choice for developers who work with multiple programming languages.

VS Code provides powerful code completion and error detection features called IntelliSense, which helps developers write code faster and with fewer errors.

VS Code's extensions system allows developers to customize and extend the editor's functionality with a wide range of plugins and add-ons. There are thousands of extensions available, covering everything from debugging tools to language support to themes and icons.

Overall, VS Code is a powerful and versatile code editor that has become a popular choice among developers due to its speed, ease of use and extensibility. It provides a range of powerful

features and supports a wide range of programming languages, making it an ideal choice for developers working on a variety of projects.

8.2 Security Features

1. **User authentication:** This feature ensures that only authorized users can access the platform by requiring them to enter login credentials, such as a username and password.
2. **Access Control:** Only authorized Person should be able to physically access the servers where the data is stored.
3. **Strong Password Policy:** The system could enforce a strong password policy that requires users to create a complex password that includes a mix of upper and lowercase letters, numbers, and special characters.
4. **Data encryption:** This feature encrypts sensitive data so that it cannot be accessed by unauthorized users, even if they manage to breach the platform's security measures.
5. **Audit logs:** This feature tracks all Admin activity on the platform, including logins, data access, and changes made to the system, providing a record of events that can be used for auditing and compliance purposes.
6. **Regular software updates:** This feature ensures that the platform is always up-to-date with the latest security patches and fixes for any known vulnerabilities, reducing the risk of exploitation by hackers or other malicious actors.

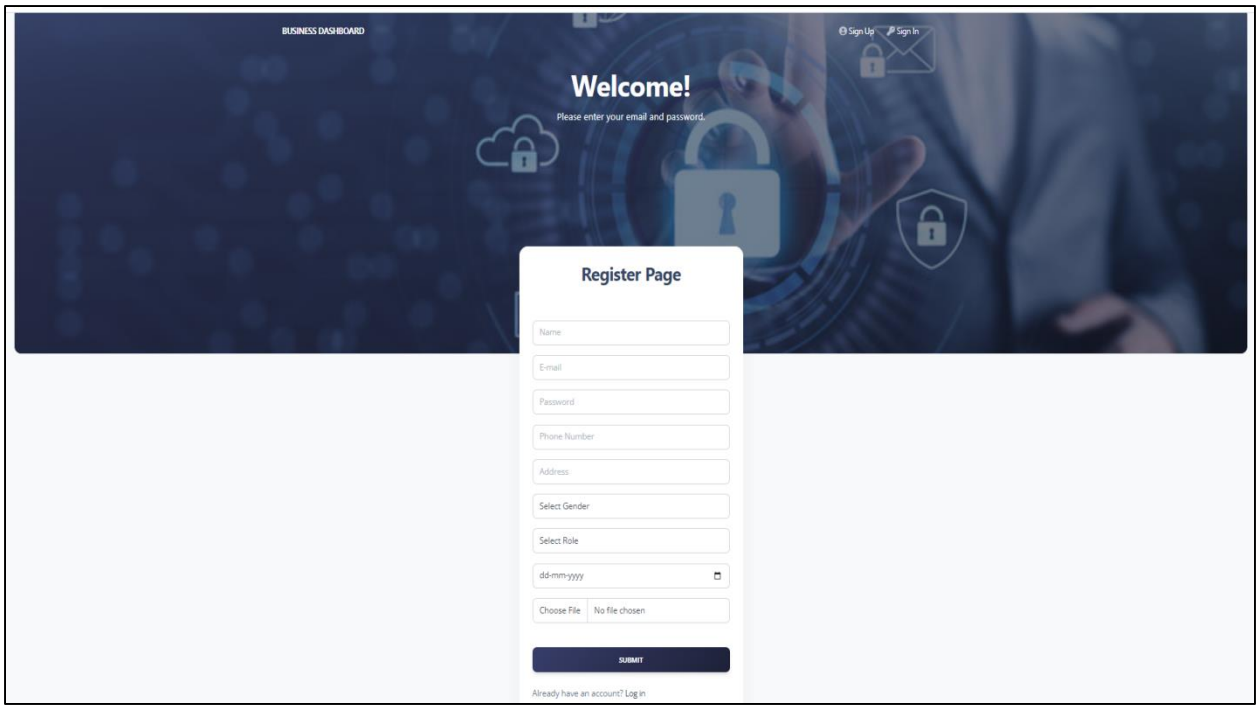
8.3 Coding Standards

As the Dashboard and Analytics Platform is built using React, some coding standards that can be followed for this project include:

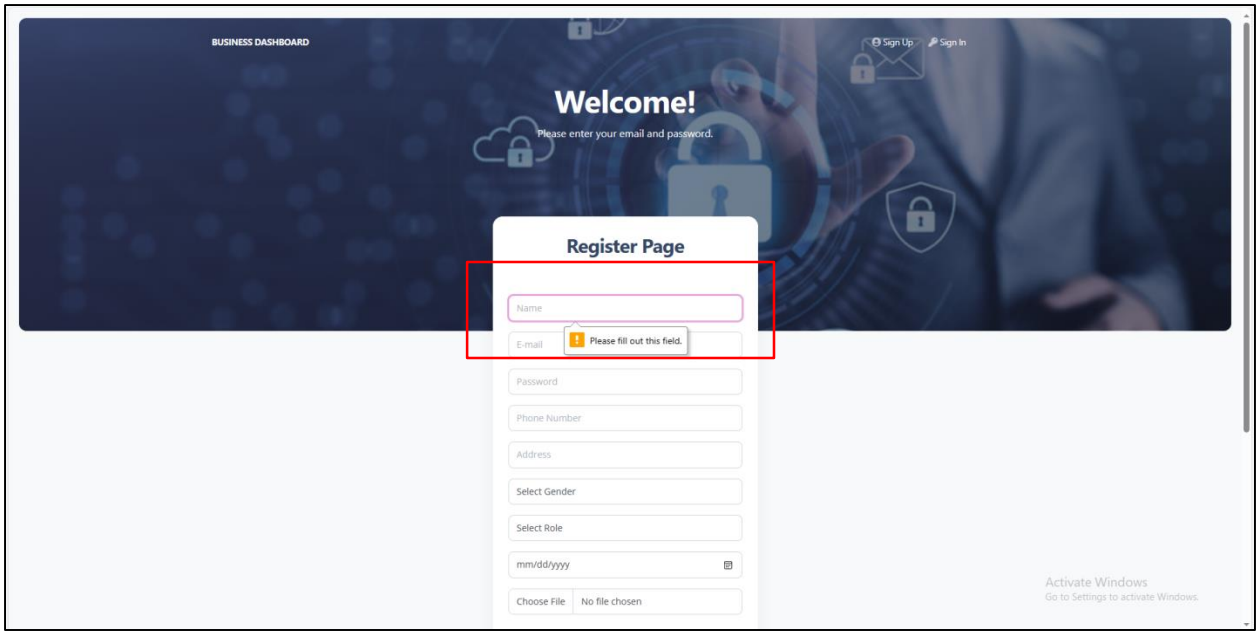
1. **Consistent code style:** Developers should follow a consistent code style throughout the project, which includes naming conventions, indentation, spacing, and other formatting choices. This ensures that the code is readable and easy to understand for all team members.
2. **Proper documentation:** It is important to document the code properly, including fun[illegible] methods, and classes. This will help develop[illegible] understand the purpose and fun[illegible]ionali[illegible] of [illegible]ach [illegible] of co[illegible]d als[illegible]e it e[illegible]o ma[illegible]nd [illegible] the co[illegible] in the [illegible]tur[illegible]

3. **Use of reusable components:** Reusable components help reduce the amount of code that needs to be written and make the project more modular. This also makes it easier to maintain the code and implement changes in the future.
4. **Consistent error handling:** The project should have consistent error handling throughout all the code. This includes handling different types of errors, displaying appropriate error messages, and logging errors for debugging purposes.
5. **Optimal performance:** The code should be optimized for performance, taking into account factors like page load times, memory usage, and CPU usage. This will ensure that the Dashboard and Analytics Platform runs smoothly and efficiently, even when dealing with large amounts of data.

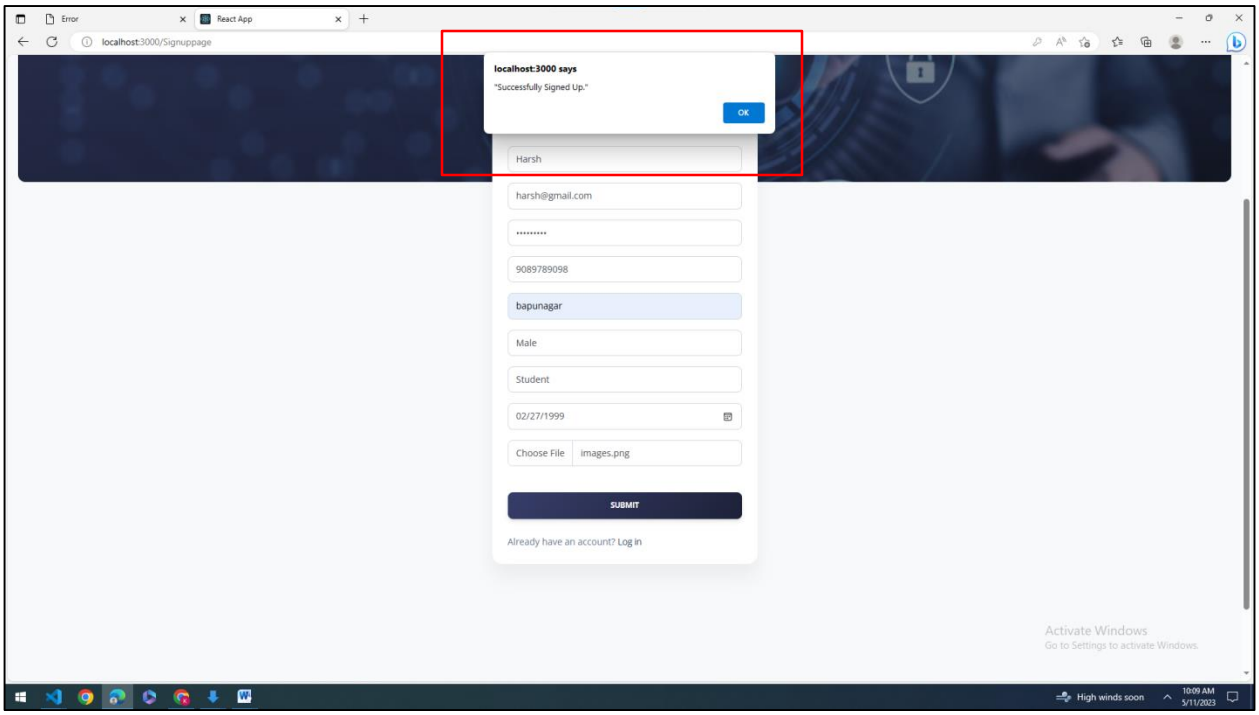
8.4 Implementation



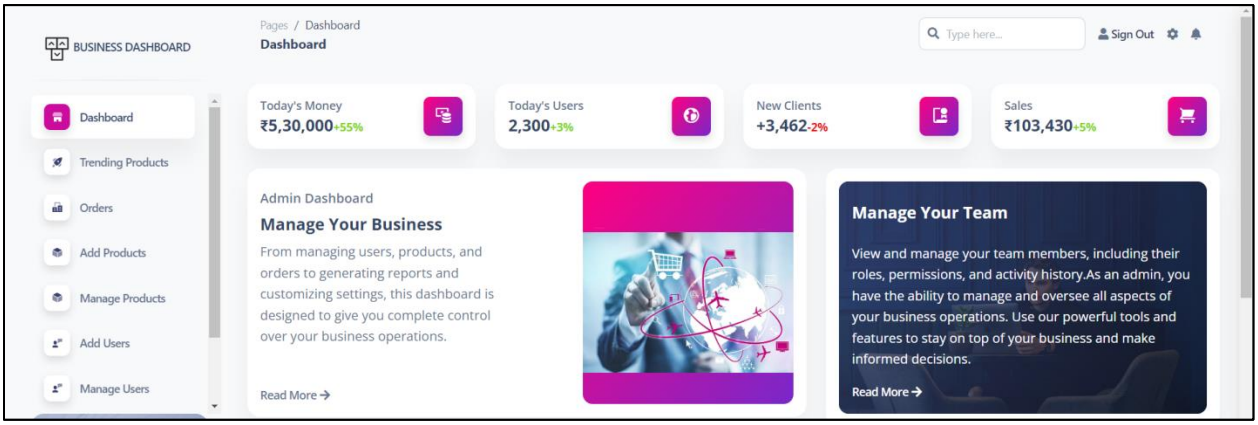
[Fig 8.4.1 Sign Up Page]



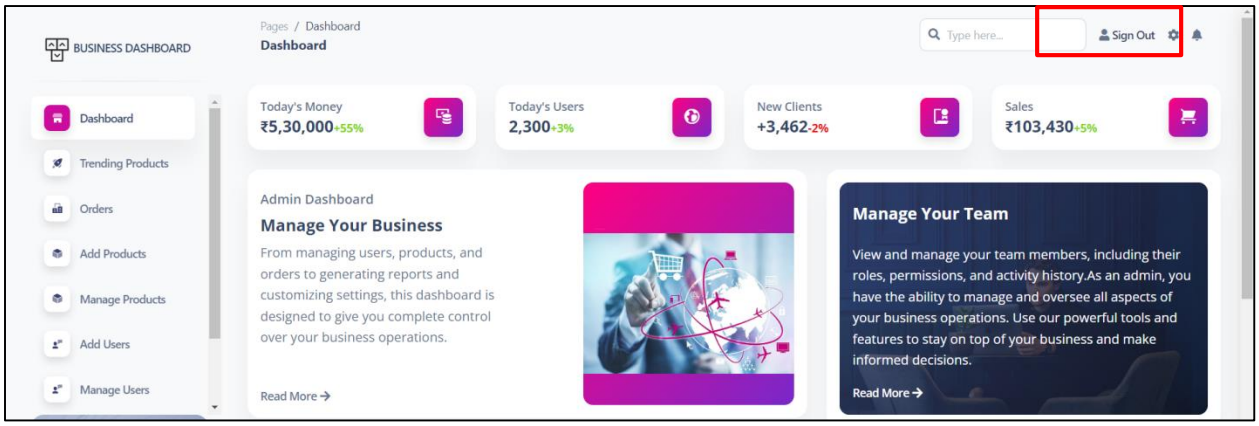
[Fig 8.4.2 Validation in SignUp page]



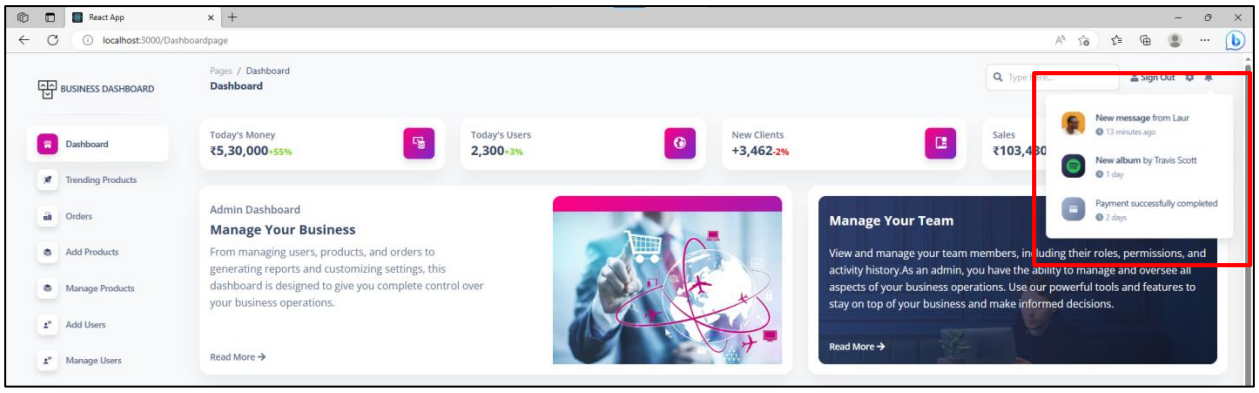
[Fig 8.4.3 User Successfully SignUp]



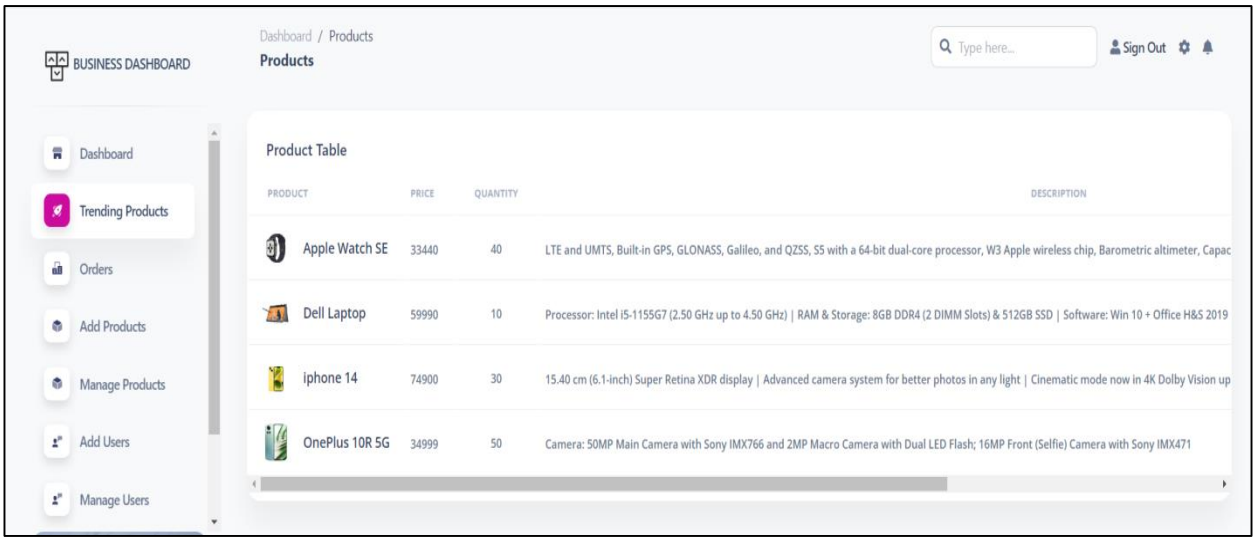
[Fig 8.4.4 Dashboard Page]



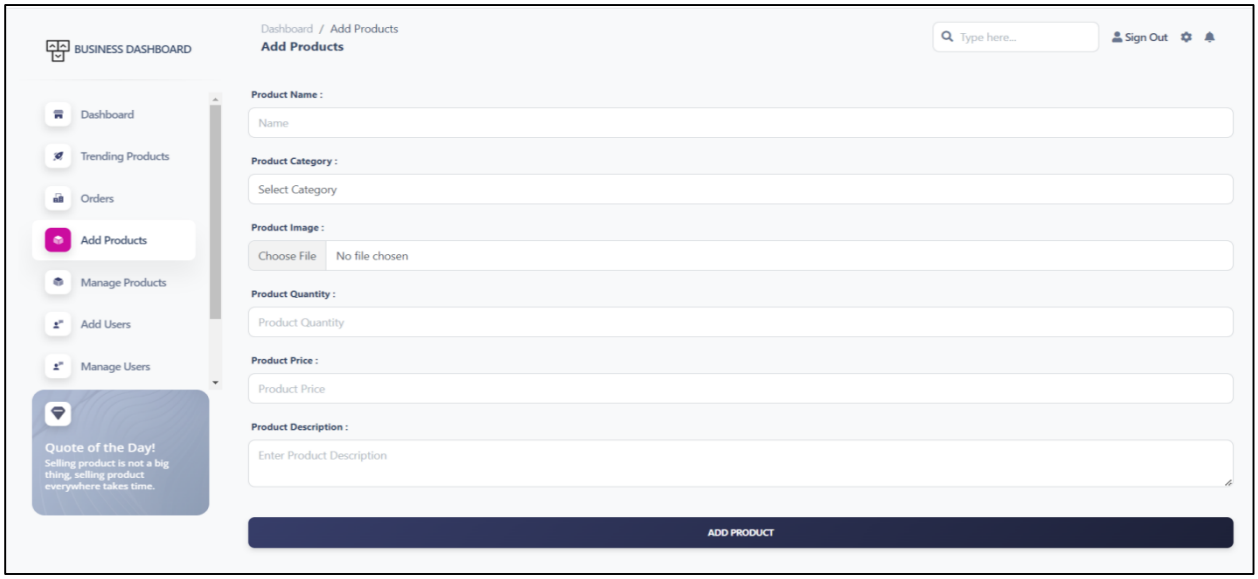
[Fig 8.4.5 Sign Out button on Navbar]



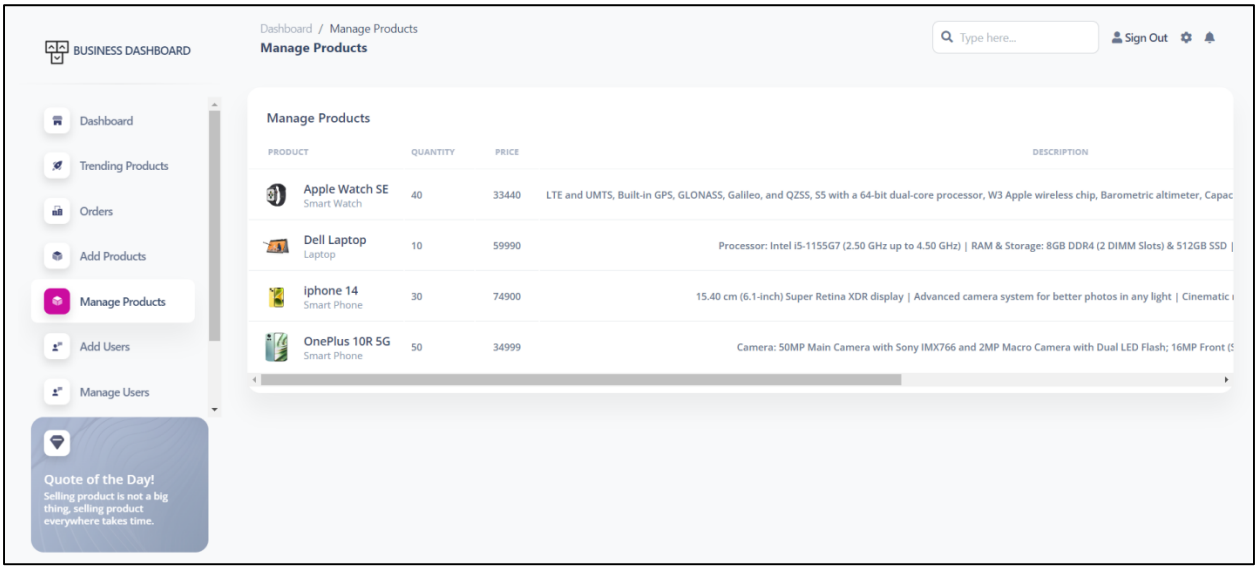
[Fig 8.4.6 Notification button on Navbar]



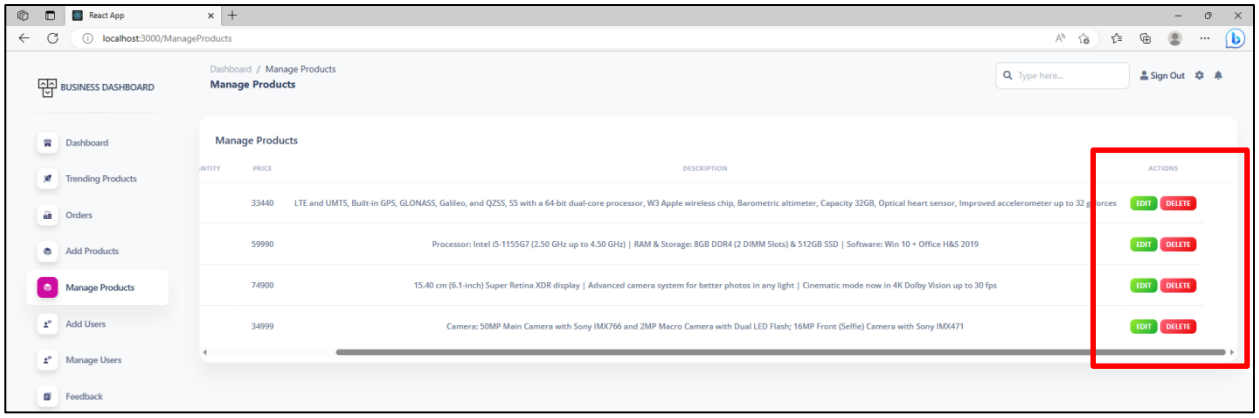
[Fig 8.4.7 Trending Products Page]



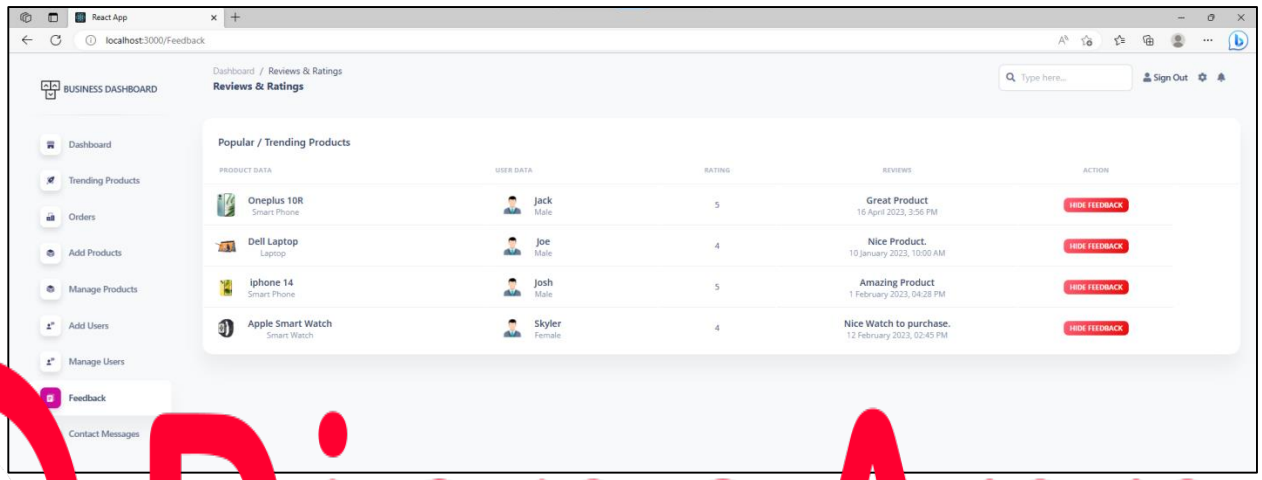
[Fig 8.4.8 Add Products Page]



[Fig 8.4.9 Manage Products Page]



[Fig 8.4.10 Edit or Delete on Manage Products Page]



[Fig 8.4.11 Feedback Page]

9. TESTING

9.1 Testing Strategy

- A well-planned and executed testing strategy can help ensure that the software is of high quality, meets the needs of its users, and is delivered on time and within budget while also minimizing costs and reducing the risk of defects and issues.
- The testing strategy followed by the company is unique in its own way.

9.2 Test Cases:

TC1 ID	Test Condition	Expected Output	Actual Output	Remarks
TC1	Verify that the login Screen contains elements such as Username, Password, Sign in button, Create Account link.	Require Fields Available	Require Fields Available	Positive
TC2	Verify the alert message when clicking on Log-In button without filling in the details Of username and Password.	Alert Message Of Fields Required	Got alert of Field required.	Positive
TC3	Verify that user will be able to login with	Welcome Message and	Redirected to Home Page	Positive

	their account with the correct password.	home Page must be shown	and welcome message Popped.	
TC4	Verify the Message Should display after just entering an username and Leaving Password field blank.	Alert message of Field Required.	Got Alert of Field required.	Positive
TC5	Verify that Password Should be in Encrypted Form.	Password must be in encrypted form	Password Is in encrypted Form.	Positive

Table 9.2.1 Login Form Test Cases

TC1 ID	Test Condition	Expected Output	Actual Output	Remarks
TC1	Verify that all the specified fields are present on the registration page.	Required fields available	Required fields available	Positive
TC2	Verify the page has submit/Sign Up button at the end.	Required fields available	Required fields available	Positive
TC3	Check that not filling out the mandatory fields and clicking the submit button will lead [illegible] error.	Alert message of field Required	Got alert of Field required.	Positive
TC4	Verify [illegible] on the em[illegible] fie[illegible] (va[illegible]	Alert m[illegible]ge of [illegible]nva[illegible] em[illegible] id	Alert mess[illegible] sho[illegible]n of [illegible]vali	Positiv[illegible]

	email Ids should be allowed.)	when incorrect email id is entered	email id when incorrect email id is entered	
TC5	Verify that system generates a validation message when entering existing username.	Alert message of User exists when username exists	Alert message shown of User exists when username exists	Positive

Table 9.2.2 Registration Form Test Cases

9.3 API Testing

- Postman is a software tool used for testing and interacting with Application Programming Interfaces (APIs).
- API testing with Postman is a popular approach to ensuring the quality and reliability of APIs. Postman can be used to test the APIs that fetch data related to weather, temperature, gases, and other environmental factors.
- The APIs can be tested by creating requests in Postman and sending them to the server to retrieve data.
- Overall, Postman is a valuable tool for developers who work with APIs and need a simple and efficient way to test and debug their code. It's widely used in the industry and can help streamline the development process.

10. CONCLUSION & FUTURE WORK

10.1 Conclusion

The Business Dashboard and Analytics Platform is a powerful solution that can help businesses gain real-time insights and make data-driven decisions. By integrating with various data sources and providing a centralized view of business metrics and KPIs, the platform enables businesses to quickly identify areas for improvement and track progress towards their goals.

Additionally, the platform's advanced analytics capabilities, such as data visualization and predictive analytics, allow businesses to gain deeper insights into their data and make informed decisions. Overall, the Dashboard and Analytics Platform is an essential tool for businesses looking to stay ahead of the curve and make data-driven decisions in today's rapidly changing business landscape.

10.2 Future Enhancement

It Is Not Possible to Develop A System That Makes All The Requirements Of The User.
User Requirements Keep Changing As The System Is Being Used. Some Of The Future Enhancements That Can Be Done To This System Are:

1. **Integration with more data sources:** Currently, the platform integrates with various data sources to provide a centralized view of business metrics and KPIs. However, there may be more data sources that businesses would like to connect to the platform, such as social media platforms, email marketing tools, and more. Adding support for these data sources would provide a more comprehensive view of a business's performance.
2. **Machine learning and AI-powered analytics:** While the platform currently offers advanced analytics capabilities, incorporating machine learning and AI-powered analytics would allow businesses to gain even deeper insights into their data. This could include predictive analytics to forecast future trends and identify potential issues before they arise.

3. **Mobile application:** Currently, the platform is accessible through a web browser. Developing a mobile application for the platform would make it more convenient for businesses to access their data on-the-go.
4. **User access control:** Adding user access control features would allow businesses to control who can access their data and what data they can see. This would be particularly useful for larger organizations with multiple departments and teams that require different levels of access to data.

10.3 Summary of Internship

Overall, internship is a really good program, it helps me to enhance and develop my skills, abilities, and knowledge. It was a good experience. **Infolabz IT Services PVT. LTD.** is best IT Company to do the internship since it provides numerous benefits and advantages to the interns. The treatment by the company was just equitable and professional. I have learned from different department's developers that how they work, how they manageclient work etc. I am grateful and thankful to my mentor **Mr. Chintan Nagrecha** and other mentors of our company for the experience and tutoring. They also help me to handle some of my weakness and provided guidance to me whenever I am in need.

11. REFERENCES

- https://in.000webhost.com/
- https://www.postman.com/
- https://github.com/
- https://react-bootstrap.github.io/
- https://www.w3schools.com/
- https://www.npmjs.com/package/react
- https://www.javatpoint.com/reactjs-tutorial
- https://dashthis.com/blog/business-dashboard-examples/
- https://blog.hubspot.com/sales/sales-dashboard