Homeds: Online Pharmacy And Healthcare Web application Using React And Firebase

Summer Internship Report

submitted in partial fulfillment of the requirement for the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Rahul

Enrollment No- 01896302719

Maharaja Surajmal Insitute of Technology

(Affiliated to Guru Gobind Singh Indraprastha University)

Janakpuri, New Delhi-58

October 2021

DECLARATION BY THE CANDIDATE

I the undersigned solemnly declare that the project report "**Homeds: Online Pharmacy And Healthcare Web application Using React And Firebase"** is based on my own work carried out during the course of my internship at JEEVITIV INDIA. I assert the statements made and conclusions drawn are an outcome of my own work. I further certify that

I. The work contained in the report is original and has been done by me under the general supervision of my supervisor.

II. The work has not been submitted to any other Institution for any other degree/diploma/certificate in this university or any other University of India or abroad.

III. The guidelines provided by the university have been kept in mind while writing the report.

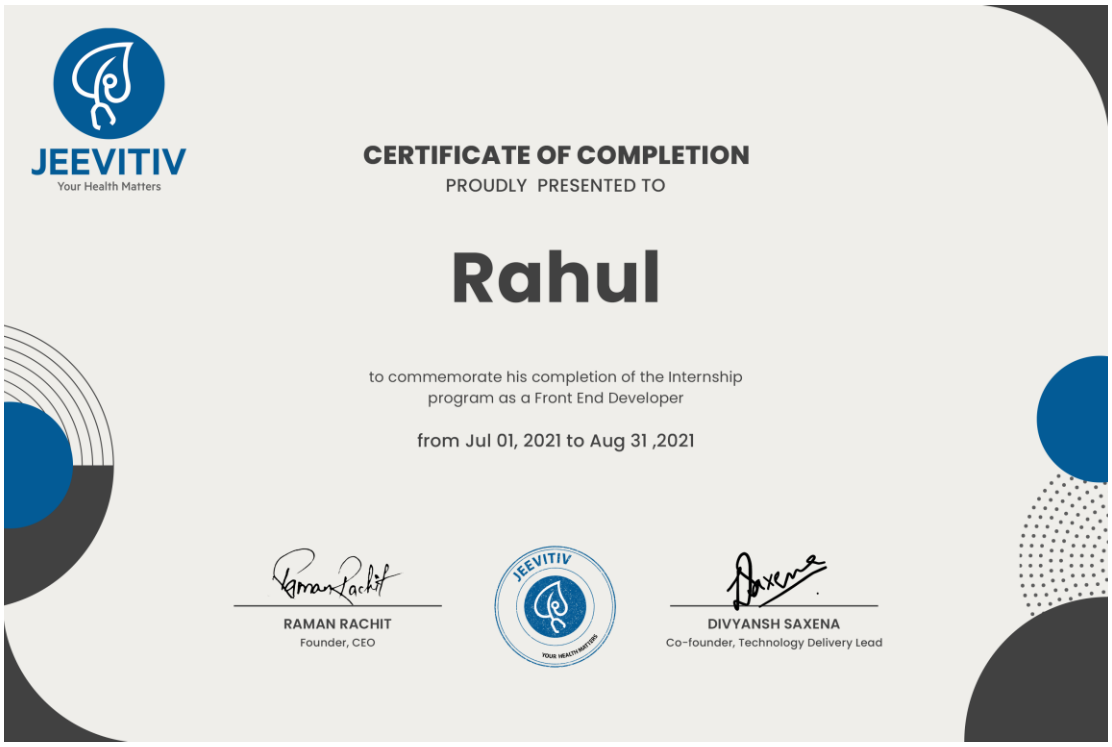
IV. Whenever we have used materials (data, theoretical analysis, and text) from other sources, we have given due credit to them in the text of the report and giving their details in the references.

[illegible]

Rahul

(Enrollment No. 01896302719)

CERTIFICATE OF INTERNSHIP



JEEVITIV
Your Health Matters

CERTIFICATE OF COMPLETION
PROUDLY PRESENTED TO

Rahul

to commemorate his completion of the Internship program as a Front End Developer

from Jul 01, 2021 to Aug 31 ,2021

Raman Rachit

RAMAN RACHIT
Founder, CEO

JEEVITIV
YOUR HEALTH MATTERS

[illegible]

DIVYANSH SAXENA
Co-founder, Technology Delivery Lead

ACKNOWLEDGMENT

A project work owes its success from commencement to completion, to the people in love with researchers at various stages. Let me in this page express my gratitude to all those who helped in various stage of this Project.

I would like to express my sincere gratitude indebtedness to our respected **Director Col. Dr. Ranjit Singh** for giving us this opportunity to discover our skills and implement them. **HOD Dr. Adeel Hashmi** and **Ms. Gunjan Beniwal** of the Computer Science Department for their most valuable and significant guidance in understanding of the project and elaborating our view of studying the project's details.

My note of thanks goes to my mentors, especially our CEO **Mr. Raman Rachit** at **JEEVITIV Pvt. Ltd.** for giving me the opportunity of be a part of it and apply my skills at the best with a lot to learn from.

Last but not least, I pay my sincere thanks and gratitude to my parents who always inspire me and keep me motivated.

Table Of Contents

Figure Index

ABSTRACT

The project comprises of an online pharmacy and healthcare web application to devise a useful platform called **"Homeds"** which enables users to order pharmaceutical and healthcare products online at the comfort of their home by making their purchase easy, simple, and affordable.

The main aim of developing this application is to provide users with the medicines and healthcare products all by just a single click and to reduce the time consumption. The application provides user sign in to the customer. Once the customer is signed in , they can get information about different products and can purchase the suitable product.The free online shopping platform conveniently allows user to shop across a wide selection of products from trusted brands across a range of categories at great prices including Diabetic care , Skin care, Multi-vitamins, Covid Essentials and Healthcare devices. Homeds also offers attractive deals of the day for shopping .With this section, our focus is on each and every category. Only the best deals for each category are shown on this section making users shopping a fun and budget friendly experience. In addition to that the users can view all the items that they have added to their cart and the total payable amount in the checkout page. In case user thinks the selected item is not useful for him, then he can remove that item form the cart.This application allows the user to browse online various products, compare prices and order merchandise sitting at home . Secure registration and profile management facilities for customers along with other features combined together in this application makes it a ready to use facility.

CHAPTER 1
COMPANY PROFILE
(JEEVITIV Pvt. Ltd.)

JEEVITIV
Your Health Matters

Fig.1: Company Logo

Jeevitiv is an EHR (Electronic health record) organization. Absolute destination for your health care. It is a scalable, and cost-effective solution for securely storing patient health records, Systematically operates with a standard range of reliable EHR features along with the facility to book their appointments with associated Doctors, easing the gap between a Doctor and Patient. With an equal facility for the doctors to easily manage their patients. It encompasses all aspects of care, including registration, patient scheduling and billing-making it a well-integrated system. It aims at providing the best service to it's associates with an easy to understand interface. Started in 2020 it brought its name under the top 50 start-ups of 2021.

Category: Hospital & Health Care
Establishment year: 2020
Headquarters: Dehradun, Uttarakhand, India
CEO: Mr. Raman Rachit
Project Lead: Mr. Divyansh Saxena

CHAPTER 2
TECHNOLOGY STACK USED

2.1 INTRODUCTION TO WEB DEVELOPMENT

Web development refers to the building, creating, and maintaining of websites. It includes aspects such as web design, web publishing, web programming, and database management. It is the creation of an application that works over the internet i.e. websites. The word Web Development is made up of two words, that is

Web : It refers to websites, web pages or anything that works over the internet.

Development : Building the application from scratch.

Examples of web applications are social networking sites like Facebook or e-commerce sites like Amazon.

Web Development can be classified into two ways:

- FRONT-END Development
- BACK-END Development

2.2 FRONT-END WEB DEVELOPMENT :

What is Front-end development?

Frontend development refers to that area of web development that focuses on what the users see on their end. It is mostly focused on what some may coin the "client side" of development. Front end developers will be engaged in analyzing code, design, and debugging applications along with ensuring a seamless user experience. You manage what people first see in their browser. They must also see to it that the website or web application is usable across different devices. As a front end developer one is responsible for the look, feel and ultimately design of the site. It involves transforming the code built by back end developers into a graphical interface, making sure that the data is presented in an easy-to-read and understand format. Without front-end development, all one would see on a website or web application are undecipherable codes. Everything you see when you visit Google Apps, Canva, Facebook, and other web applications are products of back-end and front-end developers work together. Frontend developers use several web technologies to transform coded data into user-friendly interfaces **(Shown in Fig 2)** . Among these are HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript

Fig 2 : Technologies used in Front-End development

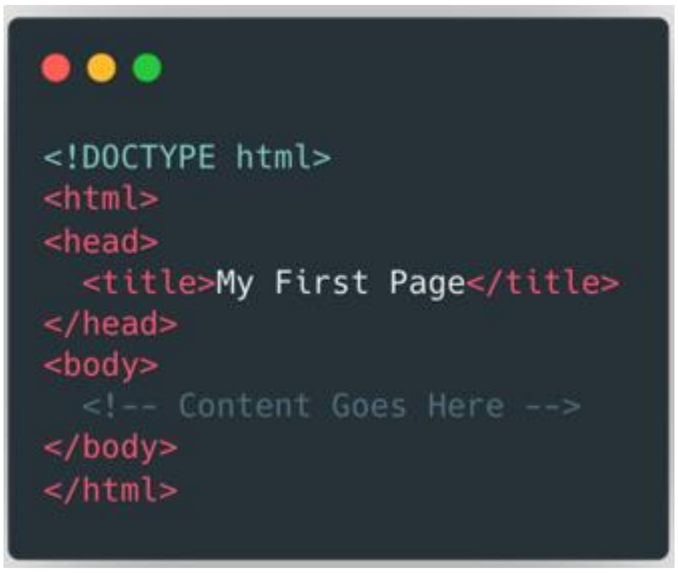
2.2.1) HTML

HTML (HyperText Markup Language) is the most basic building block of the Web. It defines the meaning and structure of web content. "Hypertext" refers to links that connect web pages to one another, either within a single website or between websites. Links are a fundamental aspect of the Web. By uploading content to the Internet and linking it to pages created by other people, you become an active participant in the World Wide Web.

HTML uses "markup" to annotate text, images, and other content for display in a Web browser. HTML markup includes special "elements" such as <head>, <title>, <body> etc .An HTML element is set off from other text in a document by "tags", which consist of the element name surrounded by "<" and ">". The name of an element inside a tag is case insensitive. That is, it can be written in uppercase, lowercase, or a mixture. For example, the <title> tag can be written as <Title>, <TITLE>, or in any other way. Example of a HTML tag <p>I am a paragraph</p>.

HTML Skeleton :

We write our HTML in a standard "skeleton **(Shown in Fig 3)**



```
<!DOCTYPE html>
<html>
<head>
  <title>My First Page</title>
</head>
<body>
  <!-- Content Goes Here -->
</body>
</html>
```

Fig 3 : HTML Skeleton

2.2.2) CSS

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML . CSS describes how elements should be rendered on screen .CSS (Cascading Style Sheets) is used to style and layout web pages — for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features.CSS is what you use to selectively style HTML elements. For example, this CSS selects paragraph text, setting the font color to red:

Example p{ color : red ; }

We still need to apply this CSS to your HTML document. Otherwise, the styling won't change the appearance of the HTML. Open your index.html file. Write the following line in the head (between the <head> and </head> tags) to link stylesheet to html file using <link href = “styles/style.css” rel = “stylesheet” >

Anatomy of a CSS ruleset

The CSS structure basics, from rulesets to values and units consist of : selectors,declarations,properties,values and units **(Shown in Fig 4)** .

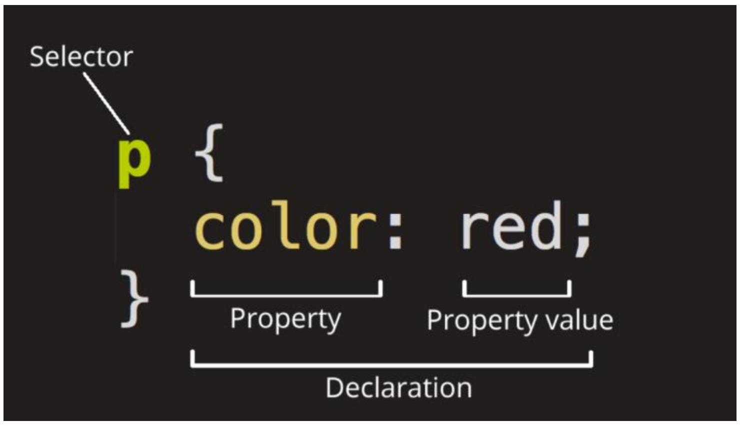


Fig 4 : CSS Ruleset

2.2.3) JavaScript

Front-end developers can already create websites using HTML and CSS. In fact, it wasn't until 1995 that JavaScript emerged. However, it is now difficult to imagine websites without JavaScript as it enables developers to make sites interactive. The programming language can change website content based on a user's action. Techslang's Weekly Poll, for example, was created using JavaScript. Selecting an answer and clicking "Vote" would display the total number of votes for each option.

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.

2.3 INTRODUCTION TO REACT JS

What is React JS?

ReactJS is a declarative, efficient, and flexible JavaScript library for building reusable UI components. It is an open-source, component-based front end library which is responsible only for the view layer of the application. It was initially developed and maintained by Facebook and later used in its products like WhatsApp & Instagram. ReactJS is one of the most popular JavaScript front-end libraries which has a strong foundation and a large community.

Why we use ReactJS?

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

2.3.1) React JS Features

1 . Virtual DOM

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 . JSX

JSX stands for JavaScript XML. It is a JavaScript syntax extension. Its an XML or HTML like syntax used by ReactJS. This syntax is processed into JavaScript calls of React Framework.
JSX provides you to write HTML/XML-like structures (e.g., DOM-like tree structures) in the same file where you write JavaScript code, then preprocessor will transform these expressions into actual JavaScript code. Just like XML/HTML, JSX tags have a tag name, attributes, and children.

3 . Components

A ReactJS application is made up of multiple components, each component responsible for outputting a small, reusable piece of HTML code. The components are the heart of all React applications. These Components can be nested with other components to allow complex applications to be built of simple building blocks.
In ReactJS, we have mainly two types of components. They are

1. Functional Components
2. Class Components

4 . Unidirectional data flow

ReactJS is designed in such a manner that follows unidirectional data flow or one-way data binding. The benefits of one-way data binding give you better control throughout the application. If the data flow is in another direction, then it requires additional features. It is because components are supposed to be immutable and the data within them cannot be changed.
React has a special object called a prop (stands for property) which we use to transport data from one component to another. Props only transport data in a one-way

flow (only from parent to child components). It is not possible with props to pass data from child to parent, or to components at the same level.

5 . Simplicity

ReactJS uses JSX file which makes the application simple and to code as well as understand. We know that ReactJS is a component-based approach which makes the code reusable as your need. This makes it simple to use and learn.

6 . Performance

ReactJS is known to be a great performer. This feature makes it much better than other frameworks out there today. The reason behind this is that it manages a virtual DOM.Due to this, when we create a component, we did not write directly to the DOM. Instead, we are writing virtual components that will turn into the DOM leading to smoother and faster performance.

2.3.2) Advantages of React JS

Intuitive

ReactJS is extremely intuitive to work with and provides interactivity to the layout of any UI. Plus, it enables fast and quality assured application development that in turn saves tome for both - clients and developers.

Declarative

ReactJS enables significant data changes that result in automatic alteration in the selected parts of user interfaces. Owing to this progressive functionality, there is no additional function that you need to perform to update your user interface.

Provides Reusable Components

ReactJS provides reusable components that developers have the authority to reuse and create a new application . Reusability is exactly like a remedy for developers. This platform gives the developers the authority to reuse the components build for some other application having the same functionality. Thereby, reducing the development effort and ensuring a flawless performance.

JavaScript library

A strong blend of JavaScript and HTML syntax is always used, which automatically simplifies the entire process of writing code for the planned project. The JS library consists several functions including one that converts the HTML components into required functions and transforms the entire project so that it is easy to understand.

Components Support

ReactJS is a perfect combination of JavaScript and HTML tags. The usage of the HTML tags and JS codes, make it easy to deal with a vast set of data containing the document object model. During this time, ReactJS works as a mediator which represents the DOM and assists to decide which component needs changes to get the exact results.

Proficient Data Binding

ReactJS trails one-way data binding. This means that absolutely anyone can track all the changes made to any particular segment of the data. This is a symbol of its simplicity.

2.3.3 REACT COMPONENTS

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. When creating a React component, the component's name MUST start with an upper case letter.

- **Class Component**

A class component must include the extends React.Component statement. This statement creates an inheritance to React.Component, and gives your component access to React.Component's functions. The component also requires a render() method, this method returns HTML

For Example :

```
class Car extends React.Component {
  render() {
    return <h2>I’m a Ferrari</h2>;
  }}

ReactDOM.render(myList, document.getElementById('root'));
```

- **Functional Component**

A **React functional component** is a straight JavaScript function that takes props and returns a React element.A React component’s primary function is to classify the displayed view and connect it to the code that governs its actions. React’s functional components reduce this to the most basic profile feasible: a function that accepts properties and returns a JSX definition. The function body contains all of the definitions needed for actions, and the class-related sections of object-oriented components are omitted.

For Example :

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <p>This is a functional</p>
    </div>
  );
}

export default App;
```

2.3.4 PROPS

Components can be passed as props, which stands for properties. Props are like function arguments, and you send them into the component as attributes.Props allows us to pass data between React components. We can pass callback functions and other pieces of data. We can attach multiple props to each component. Passing props and accessing them is very simple.

Passing Props

Props can be passed in the following ways

```
function Home(){
          return (<div>
               //passing name prop to greeting component
          <Greeting name = "Tripp" />
        </div>);
  }
```

Accessing the Props

To access the prop in our functional component we use props.name. (If this was a Class component we would use this.props.name. props.(name of prop being passed) will give us access to our prop just like an argument of a function. Example

```
function Home(){
          return (<div>
          //using prop passed down
          <p>Hi there {props.name}</p>
        </div>);
  }
```

2.3.5 REACT EVENTS

An event is an action that could be triggered as a result of the user action or system generated event. For example, a mouse click, loading of a web page, pressing a key, window resizes, and other interactions are called events.
React has its own event handling system which is very similar to handling events on DOM elements. The react event handling system is known as Synthetic Events. The synthetic event is a cross-browser wrapper of the browser's native event.

In react, we cannot return false to prevent the default behavior. We must call preventDefault event explicitly to prevent the default behavior.

Event Handling: onClick and onChange

Adding Events

- React events are written in camelCase syntax:

  onClick instead of **onclick.**

- React event handlers are written inside curly braces:

  onClick={shoot} instead of **onClick="shoot()".**

Traditional HTML method:

For Example :

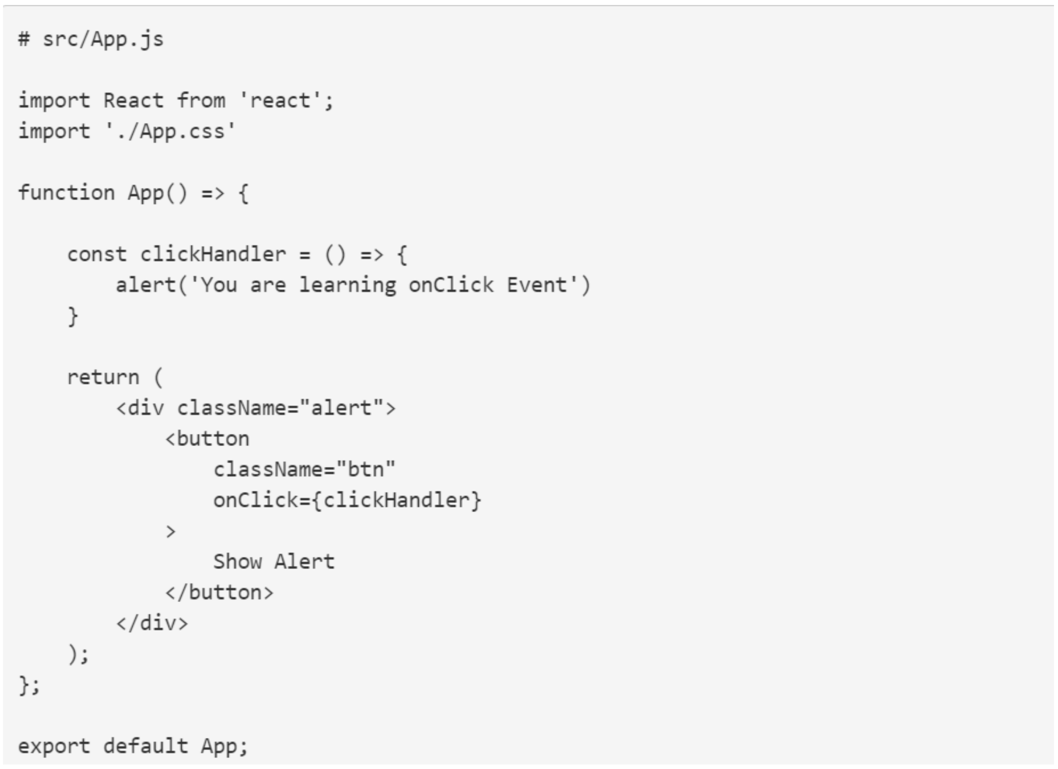
```
<button onclick="shoot()">Take the Shot!</button>
```

React:

For Example :

```
<button onClick={shoot}>Take the Shot!</button>
```

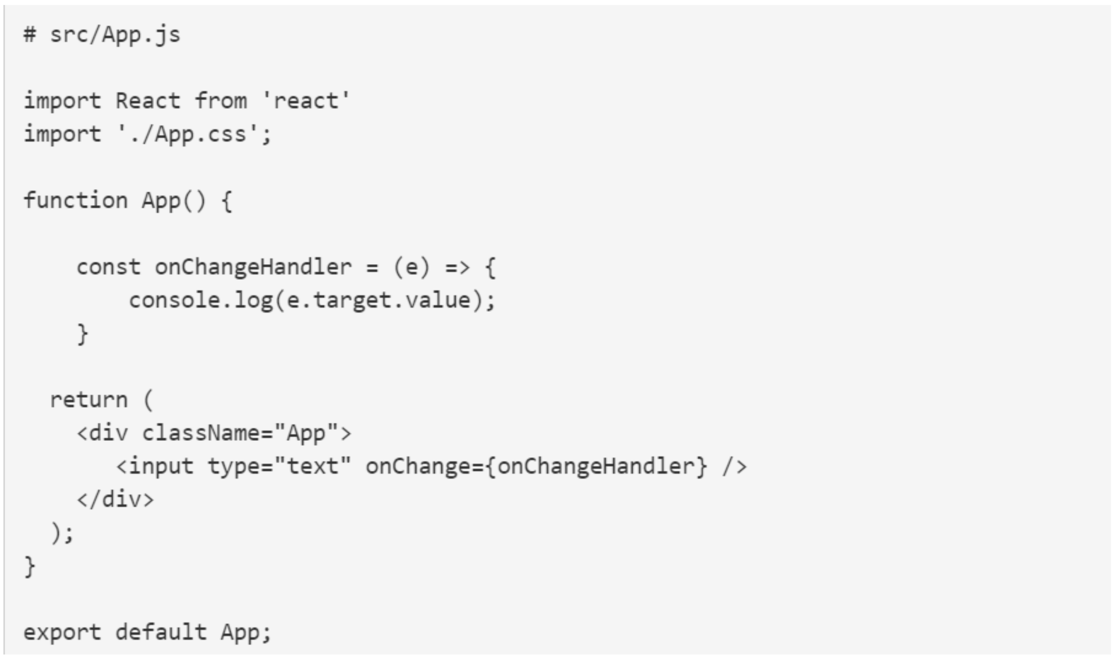
- Using a button and its **onClick event**, as well as how to identify and use various types of event handlers. **(Shown In Fig 5)**



```
# src/App.js

import React from 'react';
import './App.css'

function App() => {

    const clickHandler = () => {
        alert('You are learning onClick Event')
    }

    return (
        <div className="alert">
            <button
                className="btn"
                onClick={clickHandler}
            >
                Show Alert
            </button>
        </div>
    );
};

export default App;
```

Fig 5 : onClick Event

- The onChange event handler is a prop that can be passed through the input elements of JSX. OnChange is used in React to manage user input in real-time. If you want to generate a React type, you must use this event to monitor the value of input elements **(Shown In Fig 6) .**



```
# src/App.js

import React from 'react'
import './App.css';

function App() {

    const onChangeHandler = (e) => {
        console.log(e.target.value);
    }

  return (
    <div className="App">
      <input type="text" onChange={onChangeHandler} />
    </div>
  );
}

export default App;
```

Fig 6 : onChange Event

2.3.6 REACT HOOKS

React Hooks are functions that enable us to access React's state and lifecycle properties from function components. Hooks enable us to better manipulate the state of our **functional components** without having to turn them into class components.

Hooks are the new feature introduced in the React 16.8 version. It allows you to use state and other React features without writing a class. Hooks are the functions which "hook into" React state and lifecycle features from function components. It does not work inside classes.

Hooks are backward-compatible, which means it does not contain any breaking changes. Also, it does not replace your knowledge of React concepts.

When to use a Hooks?

If you write a function component, and then you want to add some state to it, previously you do this by converting it to a class. But, now you can do it by using a Hook inside the existing function component.

Rules of Hooks

Hooks are similar to JavaScript functions, but you need to follow these two rules when using them. Hooks rule ensures that all the stateful logic in a component is visible in its source code. These rules are:

1. Only call Hooks at the top level

Do not call Hooks inside loops, conditions, or nested functions. Hooks should always be used at the top level of the React functions. This rule ensures that Hooks are called in the same order each time a components renders.

2. Only call Hooks from React functions

You cannot call Hooks from regular JavaScript functions. Instead, you can call Hooks from React function components. Hooks can also be called from custom Hooks.

2.3.7) React useState Hook

Hook state is the new way of declaring a state in React app. **useState** is a Hook that lets you add React state to function components. . Hook uses useState() functional component for setting and retrieving state. State generally refers to data or properites that need to be tracking in an application.

For Example :

```
import React, {useState} from "react";
function Example(){
    const [count,setCount] =useState(0);
    return (
        <div>
          <p>You clicked {count}times</p>
          <button onClick={() => setCount(count+1)}>
           Click Me
          </button>
         <div>
));
```

- **Import useState:** To use the useState Hook, we first need to import it into our component. At the top of your component, import the useState Hook as shown

    -- import { useState } from "react";

- **Initialize useState:** We initialize our state by calling useState in our function component. useState accepts an initial state and returns two values:

    I. The current state.

    II. A function that updates the state.

```
--import { useState } from "react";
              function Example() {
      const [count, setCount] = useState("");
              }
```

What Do Square Brackets Mean?

Example :

```
const [fruit, setFruit] = useState("apple");
```

This JavaScript syntax is called “array destructuring”. It means that we’re making two new variables fruit and setFruit, where fruit is set to the first value returned by useState, and setFruit is the second.

For Example:

```
 var fruitStateVariable = useState("apple");
 var fruit = fruitStateVariable[0];
 var setfruit = fruitStateVariable[1];
```

- **Read State:** We can now include our state anywhere in our component.

  For Example :

```
 import { useState } from "react";
 function Example() {
 const [count, setCount] = useState("0");
 return <h1>You clicked {count} times!</h1>
```

- **Update State:** To update our state, we use our state updater function.

  For Example :

```
   import { useState } from "react";
   function Example() {
   const [count, setCount] = useState("0");
  return (
  <>
   <h1>You clicked {count} times!</h1>
   <button
     type="button"
     onClick={() => setCount(2)}
   >Click Button</button>
  </>  )}
```

2.3.8) React useEffect Hooks

What does useEffect do?

By using this Hook, you tell React that your component needs to do something after render. React will remember the function you passed (we'll refer to it as our "effect"), and call it later after performing the DOM updates. In this effect, we set the document title, but we could also perform data fetching or call some other imperative API. useEffect accepts two arguments. The second argument is optional.By default, it runs both after the first render and after every update. Instead of thinking in terms of "mounting" and "updating", you might find it easier to think that effects happen "after render". React guarantees the DOM has been updated by the time it runs the effects

useEffect (<function> , <dependency>)

For Example :

```
import React, {useState,useEffect} from "react";
function Example(){
   const [count,setCount] =useState(0);
   useEffect(()=> {
     document.title ='You clicked ${count} times';
  });
   return (
      <div>
        <p>You clicked {count}times</p>
        <button onClick={() => setCount(count+1)}>
         Click Me
        </button>
       <div>
 )};
```

We declare the count state variable, and then we tell React we need to use an effect. We pass a function to the useEffect Hook. This function we pass *is* our effect. Inside our effect, we set the document title using the document.title browser API. We can read the latest count inside the effect because it's in the scope of our function. When React renders our component, it will remember the effect we used, and then run our effect after updating the DOM. This happens for every render, including the first one.

2.3.9 REACT CONTEXT API

In a typical React application, data is passed top-down (parent to child) via props, but such usage can be cumbersome for certain types of props (e.g. locale preference, UI theme) that are required by many components within an application. Context provides a way to share values like these between components without having to explicitly pass a prop through every level of the tree.
Context is designed to share data that can be considered “global” for a tree of React components, such as the current authenticated user, theme, or preferred language. For Example

2.3.10) React.createContext

--const MyContext = React.createContext(defaultValue);

Creates a Context object. When React renders a component that subscribes to this Context object it will read the current context value from the closest matching Provider above it in the tree.

The defaultValue argument is **only** used when a component does not have a matching Provider above it in the tree. This default value can be helpful for testing components in isolation without wrapping them.

2.3.11) Context.Provider

Every Context object comes with a Provider React component that allows consuming components to subscribe to context changes.The Provider component accepts
a value prop to be passed to consuming components that are descendants of this Provider. All consumers that are descendants of a Provider will re-render whenever the Provider’s value prop changes.

<MyContext.Provider value = {/*some value*/}>

2.3.12) Example

Our goal is to control the playback of the video by clicking on the Click button. For that, we need data about the video status (playing or paused) and a way to update this status by clicking on the button. And also we'd like to escape the "prop drilling".
In a typical React app, we would have a state object in the parent component (App.js) with a status property and a function for updating the status. This state would be passed-down to direct child components (VideoClip component and Controls component) via props, and then from Controls component further to PlayPauseButton component. Classical "prop-drilling".

Let's use the help of the Context API.

```
import React from 'react';
import VideoClip from './components/video-clip.component';
import Controls from './components/controls.component';
import './App.css';
function App() {
  return (
    <div className="App">
        <VideoClip />
        <Controls />
    </div>
  );
}
export default App;
```
src/App.js

- Create VideoContext with default status value as 'paused' and a default (empty) function for updating the status.

```
import React, { createContext } from 'react';
const VideoContext = createContext({
  status: 'paused',
  togglePlayPause: () => {},
});
```

```
export default VideoContext;
```

src/context/video.context.js

Both VideoClip component and PlayPauseButton component must have access to the Video Context. We'll add state to the App.js component by implementing **useState** Hook. The default value of the status must be the same as it's default value in the Video Context.

```
import React, { useState} from 'react';
…
function App() {
  const [status, setStatus] = useState('paused');
  const togglePlayPause = () => setStatus(status === 'playing' ? 'paused' : 'playing');
...
}
```

src/context/video.context.js

In order for any child, grandchild, great-grandchild, and so on to have access to Video Context, we must wrap the parent element into VideoContext.Provider component, which will be used to pass the status and **togglePlayPause()** function via a **value** prop.

```
 ...
import VideoContext from './context/video.context';
return (
    <div className="App">
      <VideoContext.Provider
        value={{
          status,
          togglePlayPause,
        }}
      >
        <VideoClip />
        <Controls />
      </VideoContext.Provider>
        </div> );…
```

src/App.js

- To consume VideoContext we are going to use **useContext** Hook.

```
import React, { useContext } from 'react';
import VideoContext from '../context/video.context';
...

const PlayPauseButton = () => {
  const { status, togglePlayPause } = useContext(VideoContext);
  return (
    <button style={styles} onClick={togglePlayPause}>
      {status === 'playing' ? 'PAUSE' : 'PLAY'}
    </button>
  );};
```

src/components/play-pause-button.component.js

2.4 FIREBASE

Firebase is a platform developed by Google for creating mobile and web applications. It was originally an independent company founded in 2011. In 2014, Google acquired the platform and it is now their flagship offering for app development

2.4.1) Firebase Environment Setup

Step 1: Create a Firebase project and register your app

Create a Firebase project

a) In the Firebase console, click Add project.

b) To add Firebase resources to an existing Google Cloud project, enter its project name or select it from the dropdown menu.

c) To create a new project, enter the desired project name. You can also optionally edit the project ID displayed below the project name.

d) Firebase generates a unique ID for your Firebase project based upon the name you give it. If you want to edit this project ID, you must do it now as it cannot be altered after Firebase provisions resources for your project. Visit Understand Firebase Projects to learn about how Firebase uses the project ID.

e) If prompted, review and accept the Firebase terms.

f) Click Continue.

g) Click Create project (or Add Firebase, if you're using an existing Google Cloud project).

h) Firebase automatically provisions resources for your Firebase project. When the process completes, you'll be taken to the overview page for your Firebase project in the Firebase console.

i) Register your app

j) After you have a Firebase project, you can register your web app with that project.

k) In the center of the Firebase console's project overview page, click the Web icon (plat_web) to launch the setup workflow.

- If you've already added an app to your Firebase project, click Add app to display the platform options.

- Enter your app's nickname.
- This nickname is an internal, convenience identifier and is only visible to you in the Firebase console.

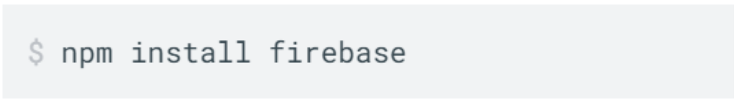
- Click Register app.

- Follow the on-screen instructions to add and initialize the Firebase SDK in your app.

- You can also find more detailed instructions for adding, initializing, and using the Firebase SDK in the next steps of this getting started page.

Step 2: Install the SDK and initialize Firebase

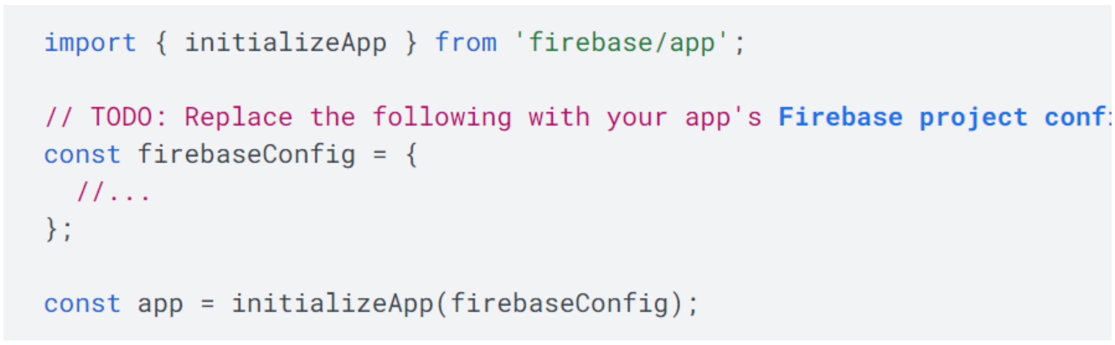
Follow the steps to Install Firebase :-



```
$ npm install firebase
```

Fig 7 : Install Firebase using npm:

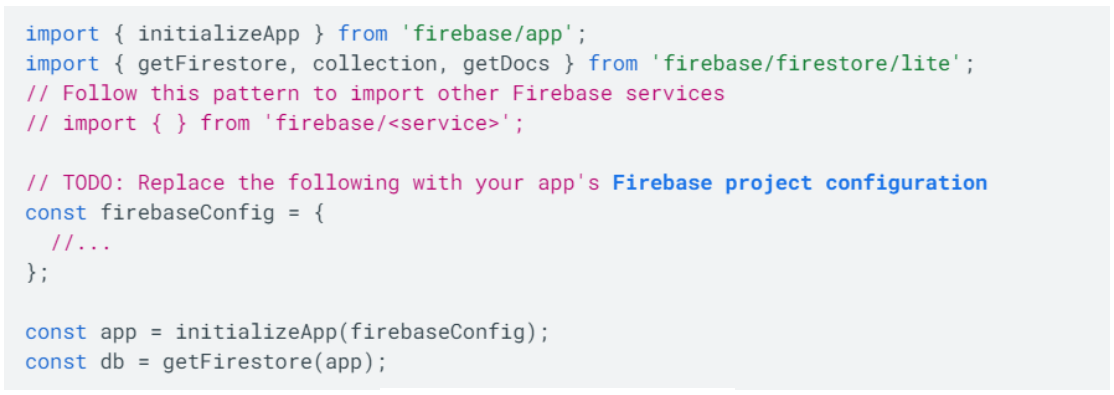
- Initialize Firebase in your app and create a Firebase App object as following:



```
import { initializeApp } from 'firebase/app';

// TODO: Replace the following with your app's Firebase project conf
const firebaseConfig = {
  //...
};

const app = initializeApp(firebaseConfig);
```

Fig 8 : Initializing Firebase

Step 3: Access Firebase in your app as following



```
import { initializeApp } from 'firebase/app';
import { getFirestore, collection, getDocs } from 'firebase/firestore/lite';
// Follow this pattern to import other Firebase services
// import { } from 'firebase/<service>';

// TODO: Replace the following with your app's Firebase project configuration
const firebaseConfig = {
  //...
};

const app = initializeApp(firebaseConfig);
const db = getFirestore(app);
```

Fig 9 : Accessing Firebase

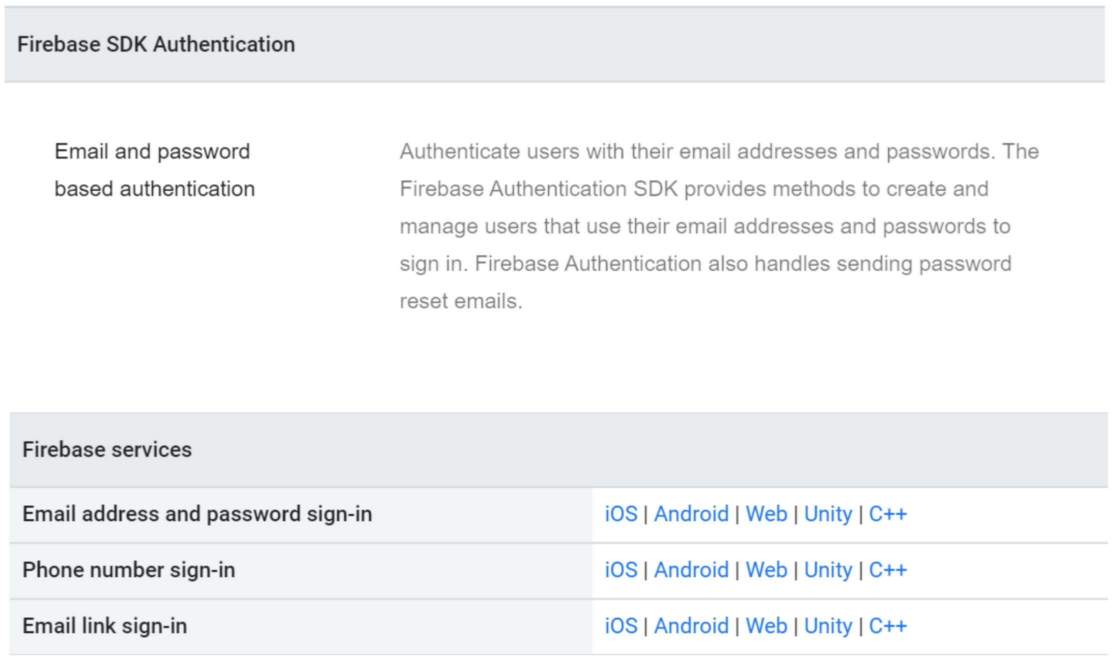
2.4.2) Firebase Authentication

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

How does it work?

To sign a user into your app, you first get authentication credentials from the user. These credentials can be the user's email address and password, or an OAuth token from a federated identity provider. Then, you pass these credentials to the Firebase Authentication SDK. Our backend services will then verify those credentials and return a response to the client.

After a successful sign in, you can access the user's basic profile information, and you can control the user's access to data stored in other Firebase products. You can also use the provided authentication token to verify the identity of users in your own backend services.**(Shown In Fig 10)**



Firebase SDK Authentication	
Email and password based authentication	Authenticate users with their email addresses and passwords. The Firebase Authentication SDK provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

Firebase services	
Email address and password sign-in	iOS | Android | Web | Unity | C++
Phone number sign-in	iOS | Android | Web | Unity | C++
Email link sign-in	iOS | Android | Web | Unity | C++

Fig 10 : Firebase Authentication

2.4.3) Firebase Hosting

Firebase Hosting provides fast and secure hosting for your web app, static and dynamic content, and microservices. Firebase Hosting is production-grade web content hosting for developers. With a single command, you can quickly deploy web apps.

For serving your content,Firebase offers several domain and subdomain options:

By default, every Firebase project has free subdomains on the web.app and firebase app.com domains. These two sites serve the same deployed content and configuration.You can create multiple sites if you have related sites and apps that serve different content but still share the same Firebase project resources (for example if you have a blog, admin panel, and public app).

You can connect your own domain name to a Firebase-hosted site.

Chapter 3
PROJECT DEMONSTRATION

3.1 AIM - This chapter will explore the different aspects concerned with the implementation of the developed system.The main aim of developing this application is to provide users with the medicines and healthcare products all by just a single click and increase access, lower the transaction rate and product costs, convenience and greater anonymity for consumers. Offer accessibility to people with limited mobility and people in remote areas. These provide medialerts (personalized medicine reminder service), discounts, doorstep delivery within a short time and save time consumption.

3.2 SETTING UP PROJECT USING REACT

React Js has been used for creating front-end of various components with Java Script as it helps in the creation of reusable components.

Setting up a React Environment`

If you have npx and Node.js installed, you can create a React application by using create-react-app

Step 1 - Install create-react-app

Browse through the desktop and install the Create React App using command prompt as shown below .Run this command to create a React application named my-react-app:-

----npx create-react-app homeds

This creates a my-react-app directory, and does several things inside it:

- Installs some npm packages essential to the functionality of the app.
- Writes scripts for starting and serving the application.
- Creates a structure of files and directories that define the basic app architecture.
- Initializes the directory as a git repository, if you have git installed on your computer.

Step 2 - Run the React Application

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

Run this command to move to the my-react-app directory:

------cd homeds

Run this command to run the React application my-react-app:

------npm start

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

The result **(Shown In Fig 11 And 12)**

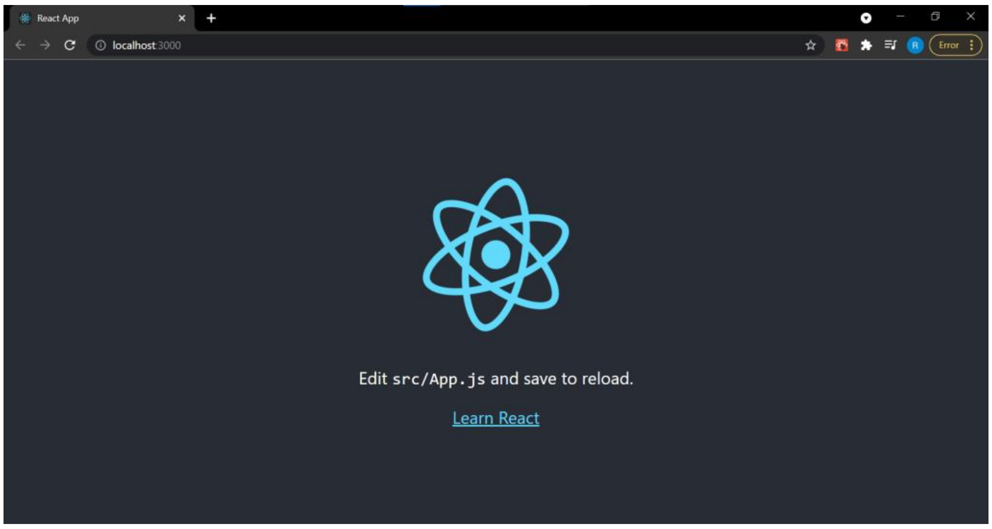
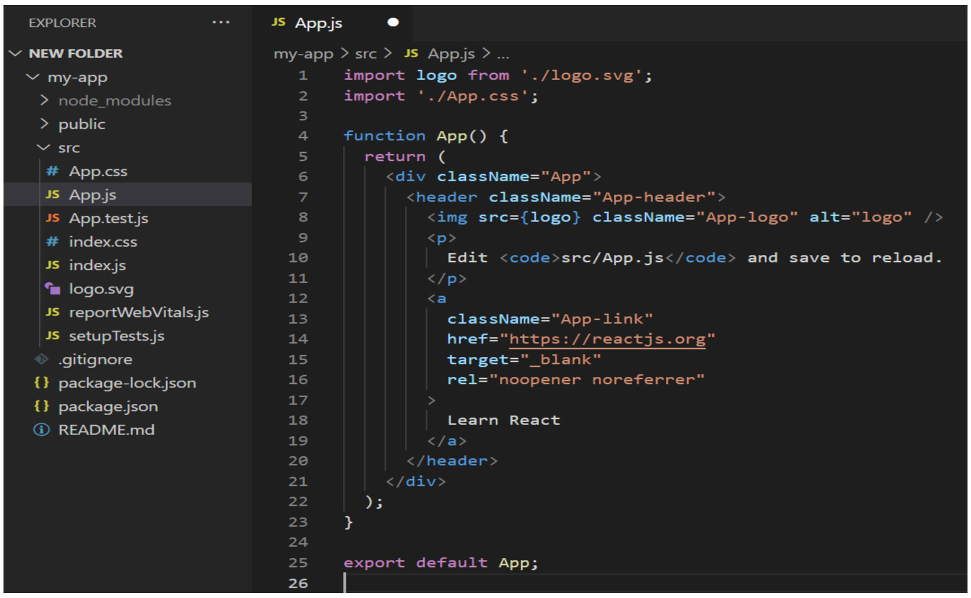


Fig 11 : Initial React App Screen



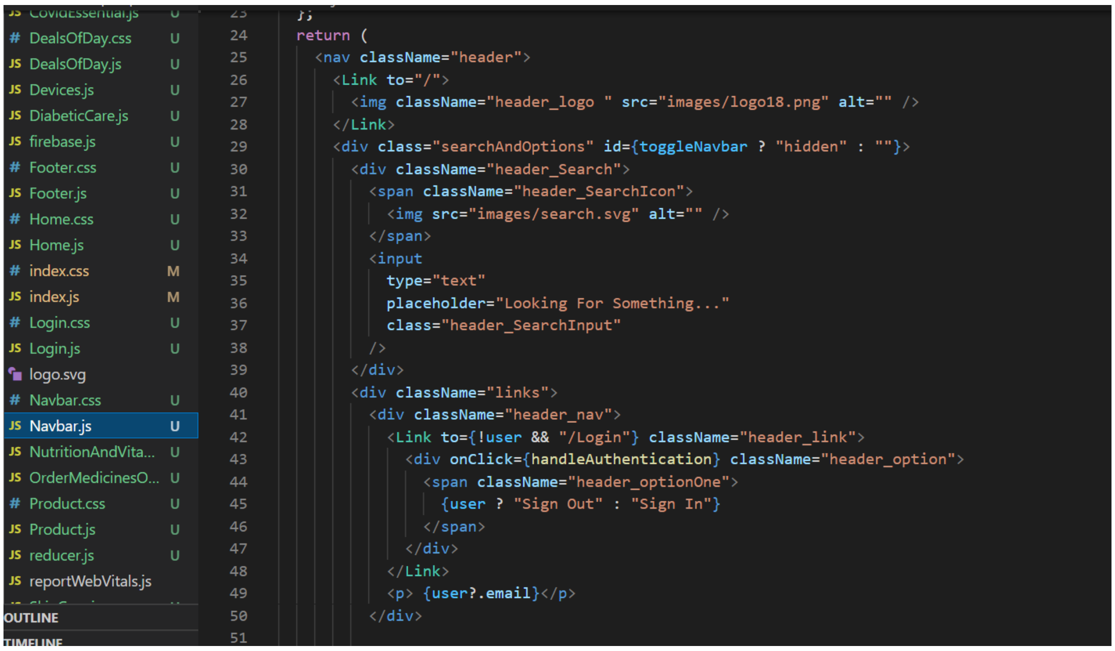
```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;

```

Fig 12 : React Application structure

3.3 CREATING COMPONENTS

3.2.1) Creating Navigation Bar Component

First is the Navigation bar where the email id through which the user has signed is shown along with a Sign Out Option. Upon clicking on “Categories” option will redirect the customer to the Shop By category section. Total number of items will be shown alongside the cart icon **(Shown In Fig 13).**

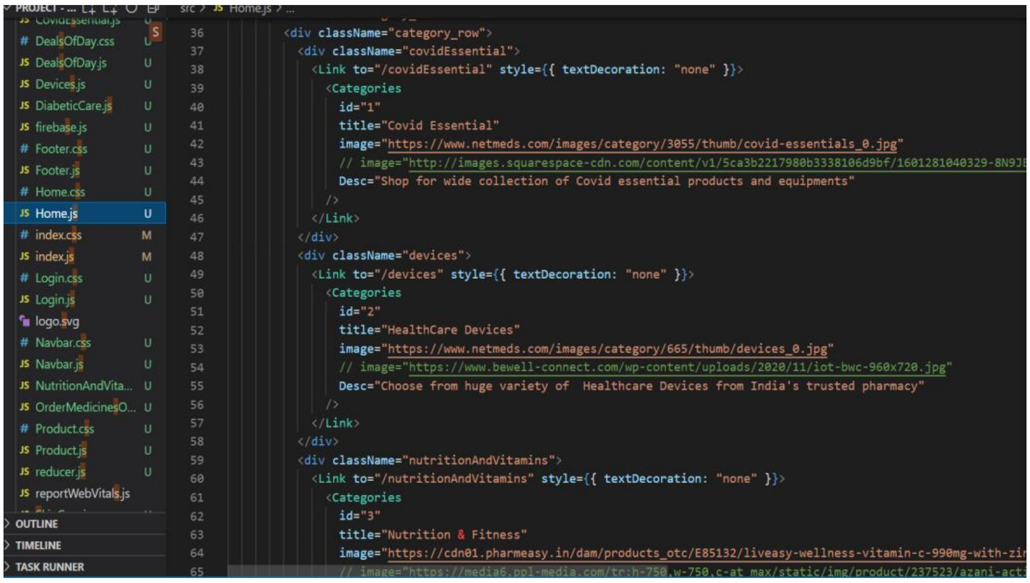


```
    return (
      <nav className="header">
        <Link to="/">
          <img className="header_logo " src="images/logo18.png" alt="" />
        </Link>
        <div class="searchAndOptions" id={toggleNavbar ? "hidden" : ""}>
          <div className="header_Search">
            <span className="header_SearchIcon">
              <img src="images/search.svg" alt="" />
            </span>
            <input
              type="text"
              placeholder="Looking For Something..."
              class="header_SearchInput"
            />
          </div>
          <div className="links">
            <div className="header_nav">
              <Link to={!user && "/Login"} className="header_link">
                <div onClick={handleAuthentication} className="header_option">
                  <span className="header_optionOne">
                    {user ? "Sign Out" : "Sign In"}
                  </span>
                </div>
              </Link>
              <p> {user?.email}</p>
            </div>
```

Fig 13: Navigation Bar Component

3.2.2) Creating Home Page Component

Inside Home page we will call various components like Navigation bar, Shop By Category , Deals of the Day and Footer **(Shown In Fig 14).**.

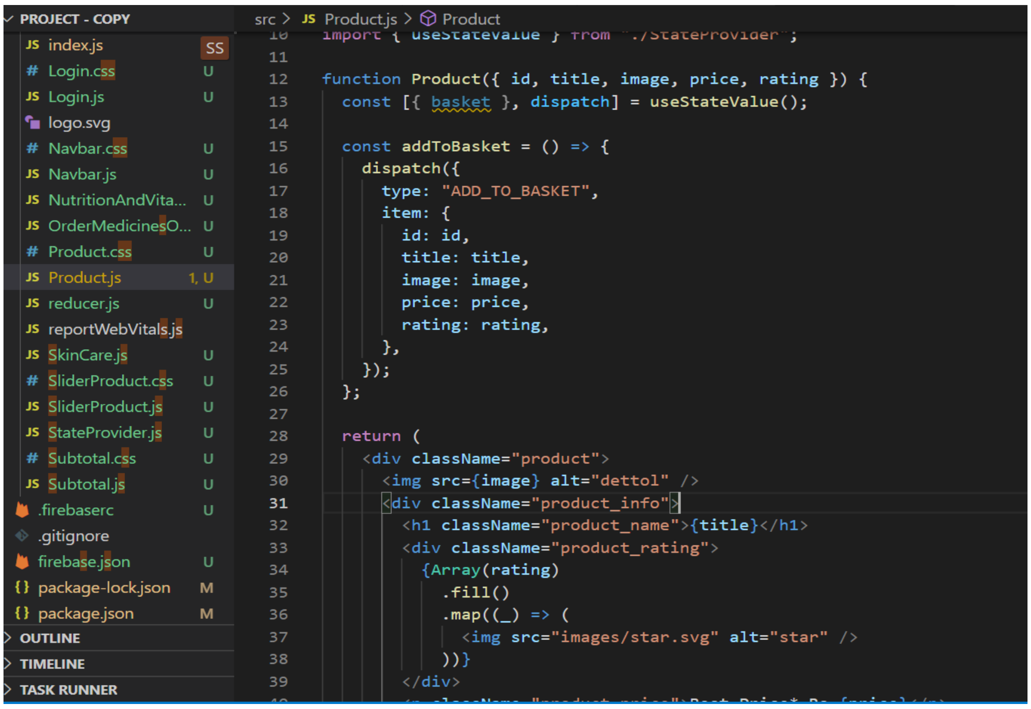


```
      <div className="category_row">
        <div className="covidEssential">
          <Link to="/covidEssential" style={{ textDecoration: "none" }}>
            <Categories
              id="1"
              title="Covid Essential"
              image="https://www.netmeds.com/images/category/3055/thumb/covid-essentials_0.jpg"
              // image="http://images.squarespace-cdn.com/content/v1/5ca3b2217980b3338106d9bf/1601281040329-8N9J
              Desc="Shop for wide collection of Covid essential products and equipments"
            />
          </Link>
        </div>
        <div className="devices">
          <Link to="/devices" style={{ textDecoration: "none" }}>
            <Categories
              id="2"
              title="HealthCare Devices"
              image="https://www.netmeds.com/images/category/665/thumb/devices_0.jpg"
              // image="https://www.bewell-connect.com/wp-content/uploads/2020/11/iot-bwc-960x720.jpg"
              Desc="Choose from huge variety of  Healthcare Devices from India's trusted pharmacy"
            />
          </Link>
        </div>
        <div className="nutritionAndVitamins">
          <Link to="/nutritionAndVitamins" style={{ textDecoration: "none" }}>
            <Categories
              id="3"
              title="Nutrition & Fitness"
              image="https://cdn01.pharmeasy.in/dam/products_otc/E85132/liveasy-wellness-vitamin-c-990mg-with-zi
              // image="https://media6.ppl-media.com/tr:h-750,w-750,c-at_max/static/img/product/237523/azani-act
```

Fig 14 : Home Page Component

3.2.3) Creating Product Component

Product component refers to single product/item that we incorporate in categories section and deals of the day section **(Shown In Fig 15).**

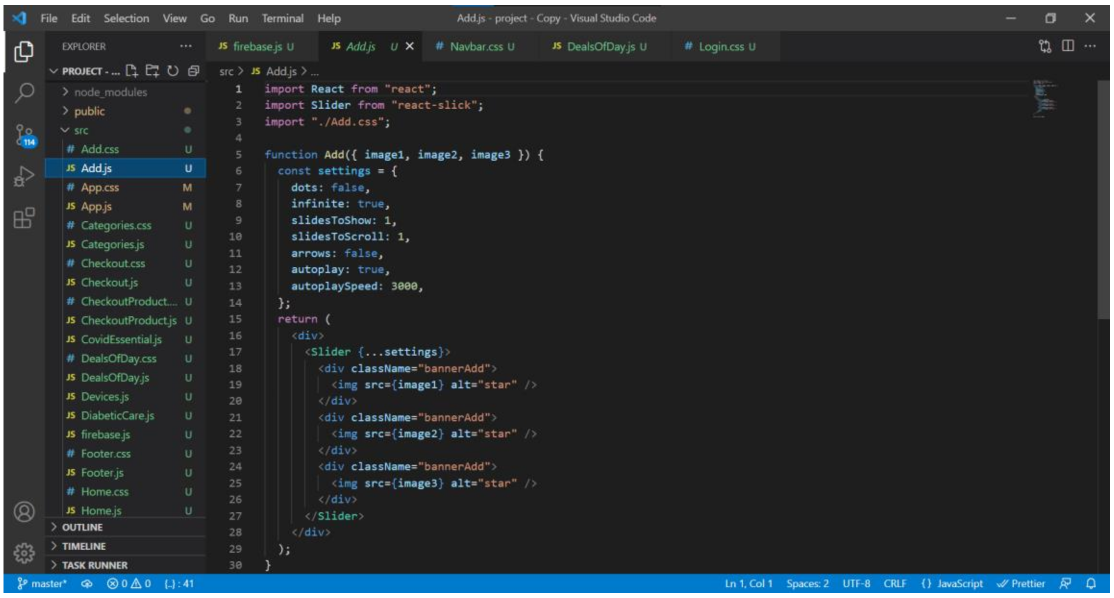


```
import { useStateValue } from "./StateProvider";

function Product({ id, title, image, price, rating }) {
  const [{ basket }, dispatch] = useStateValue();

  const addToBasket = () => {
    dispatch({
      type: "ADD_TO_BASKET",
      item: {
        id: id,
        title: title,
        image: image,
        price: price,
        rating: rating,
      },
    });
  };

  return (
    <div className="product">
      <img src={image} alt="dettol" />
      <div className="product_info">
        <h1 className="product_name">{title}</h1>
        <div className="product_rating">
          {Array(rating)
            .fill()
            .map((_) => (
              <img src="images/star.svg" alt="star" />
            ))}
        </div>
```

Fig 15 : Product Component

3.2.4) Creating Add Component

Add component consist of react slick component .React slick is a carousel component built with React. It is a react port of slick carousel **(Shown In Fig 16).**

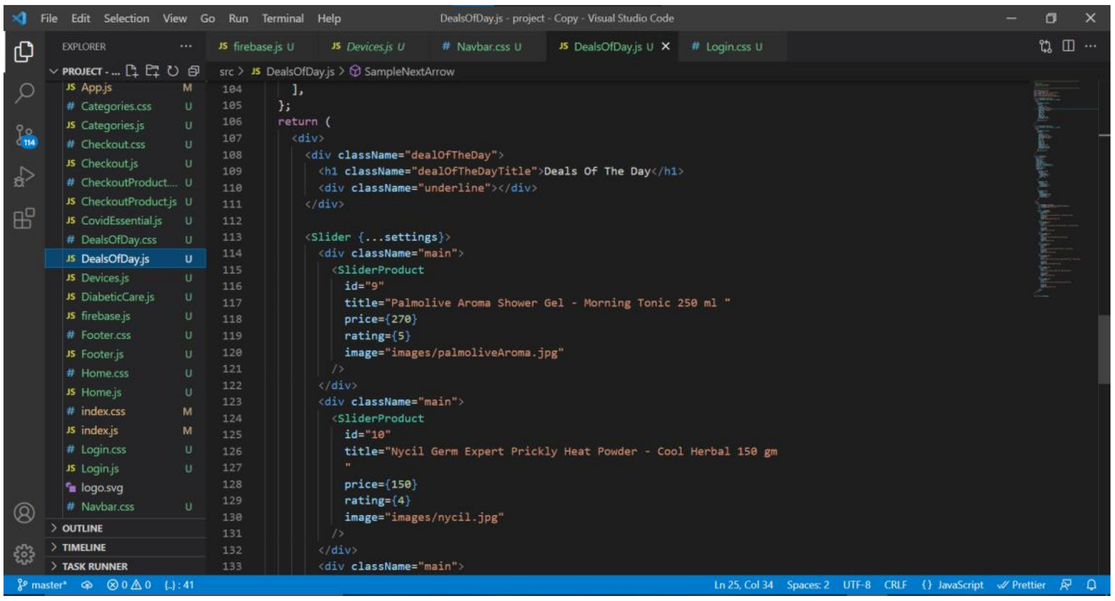


```
import React from "react";
import Slider from "react-slick";
import "./Add.css";

function Add({ image1, image2, image3 }) {
  const settings = {
    dots: false,
    infinite: true,
    slidesToShow: 1,
    slidesToScroll: 1,
    arrows: false,
    autoplay: true,
    autoplaySpeed: 3000,
  };
  return (
    <div>
      <Slider {...settings}>
        <div className="bannerAdd">
          <img src={image1} alt="star" />
        </div>
        <div className="bannerAdd">
          <img src={image2} alt="star" />
        </div>
        <div className="bannerAdd">
          <img src={image3} alt="star" />
        </div>
      </Slider>
    </div>
  );
}
```

Fig 16 : Add Component

3.2.4) Creating Deal Of The Day Component

We use React Slick component to create carousel in this section .Only the best deals for each category are updated on this section **(Shown In Fig 17).**

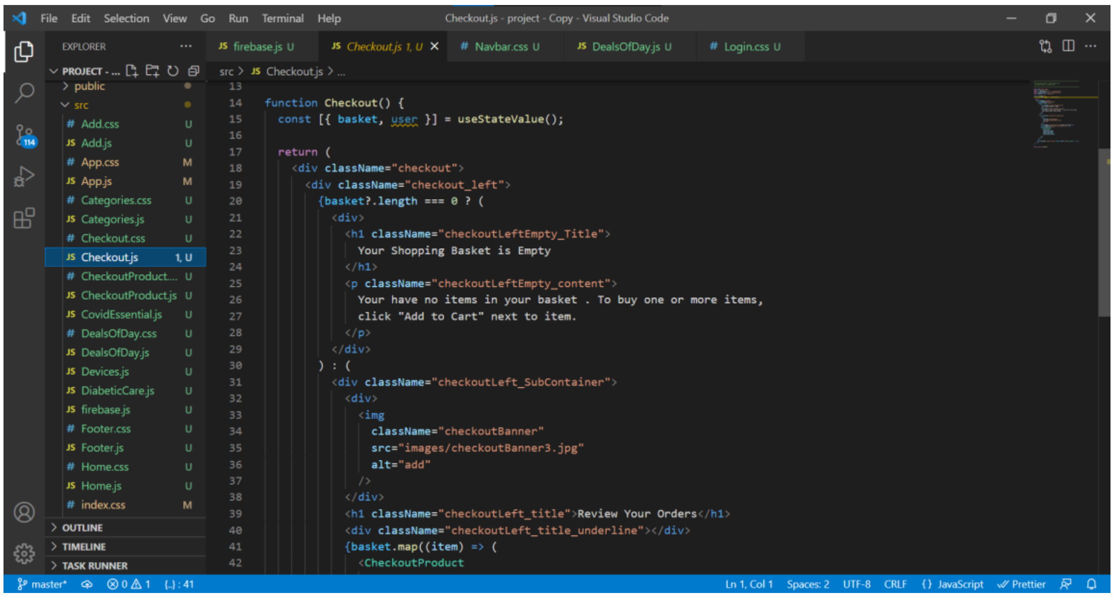


```
      ],
  };
  return (
    <div>
      <div className="dealOfTheDay">
        <h1 className="dealOfTheDayTitle">Deals Of The Day</h1>
        <div className="underline"></div>
      </div>

      <Slider {...settings}>
        <div className="main">
          <SliderProduct
            id="9"
            title="Palmolive Aroma Shower Gel - Morning Tonic 250 ml "
            price={270}
            rating={5}
            image="images/palmoliveAroma.jpg"
          />
        </div>
        <div className="main">
          <SliderProduct
            id="10"
            title="Nycil Germ Expert Prickly Heat Powder - Cool Herbal 150 gm
"
            price={150}
            rating={4}
            image="images/nycil.jpg"
          />
        </div>
        <div className="main">
```

Fig 17 : Deal Of Day Component

3.2.5) Creating Checkout Page Component

The users can view all the items that they have added to their cart and the total payable amount and total no of items in the checkout page **(Shown In Fig 18).** .

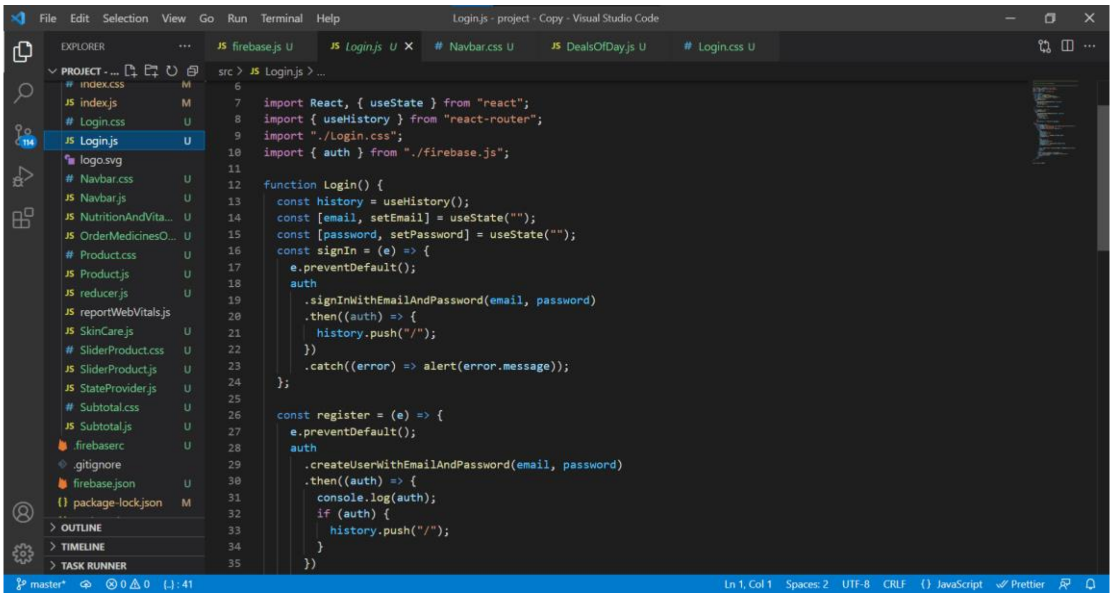


```
function Checkout() {
  const [{ basket, user }] = useStateValue();

  return (
    <div className="checkout">
      <div className="checkout_left">
        {basket?.length === 0 ? (
          <div>
            <h1 className="checkoutLeftEmpty_Title">
              Your Shopping Basket is Empty
            </h1>
            <p className="checkoutLeftEmpty_content">
              Your have no items in your basket . To buy one or more items,
              click "Add to Cart" next to item.
            </p>
          </div>
        ) : (
          <div className="checkoutLeft_SubContainer">
            <div>
              <img
                className="checkoutBanner"
                src="images/checkoutBanner3.jpg"
                alt="add"
              />
            </div>
            <h1 className="checkoutLeft_title">Review Your Orders</h1>
            <div className="checkoutLeft_title_underline"></div>
            {basket.map((item) => (
              <CheckoutProduct
```

Fig 18 : Checkout Page Component

3.2.5) Creating SignUp / SignIn Component

Firebase is used for managing user authentication and their credentials .It include taking email id and password as input from the user **(Shown In Fig 19)**.



```

import React, { useState } from "react";
import { useHistory } from "react-router";
import "./Login.css";
import { auth } from "./firebase.js";

function Login() {
  const history = useHistory();
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const signIn = (e) => {
    e.preventDefault();
    auth
      .signInWithEmailAndPassword(email, password)
      .then((auth) => {
        history.push("/");
      })
      .catch((error) => alert(error.message));
  };

  const register = (e) => {
    e.preventDefault();
    auth
      .createUserWithEmailAndPassword(email, password)
      .then((auth) => {
        console.log(auth);
        if (auth) {
          history.push("/");
        }
      })
```

Fig 19 : SignUp / SignIn Component

Chapter 4
PROJECT OVERVIEW

4.1 PROJECT SCREENSHOT

1) Sign Up / Sign In

a) When the user enters the website, User will be asked to create a Homeds account by filling in the credentials ie Email Id and Password. Once user has entered all the credentials ,they can either click on "Create Account" button if they are a new user or they can click on "Sign In" if they already have a Homeds account.**(Shown In Fig 20)**.

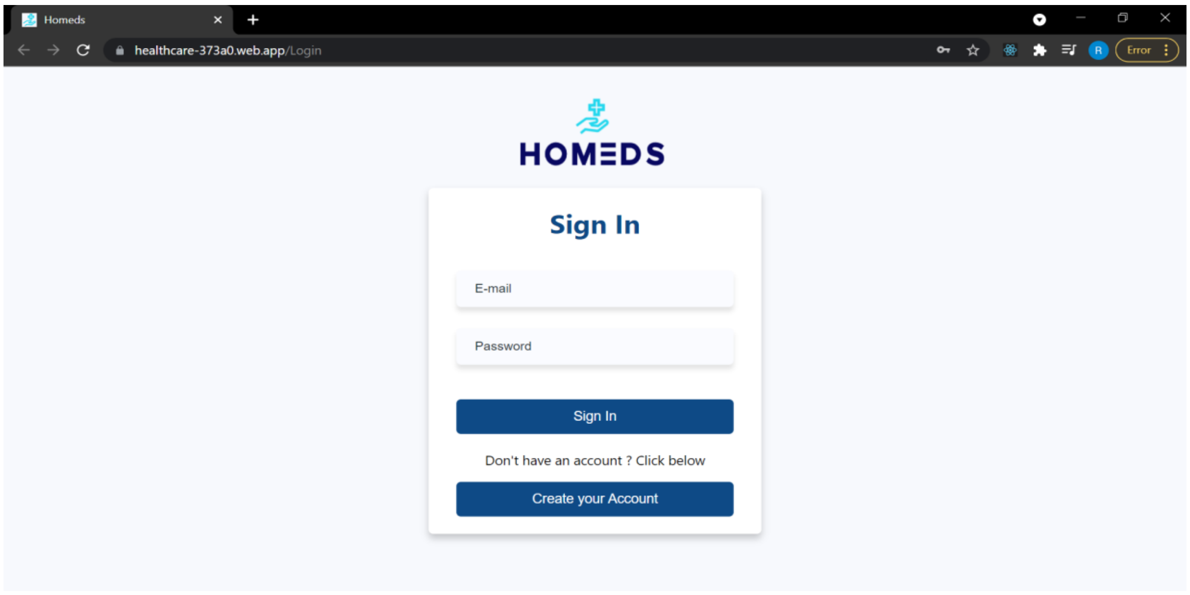
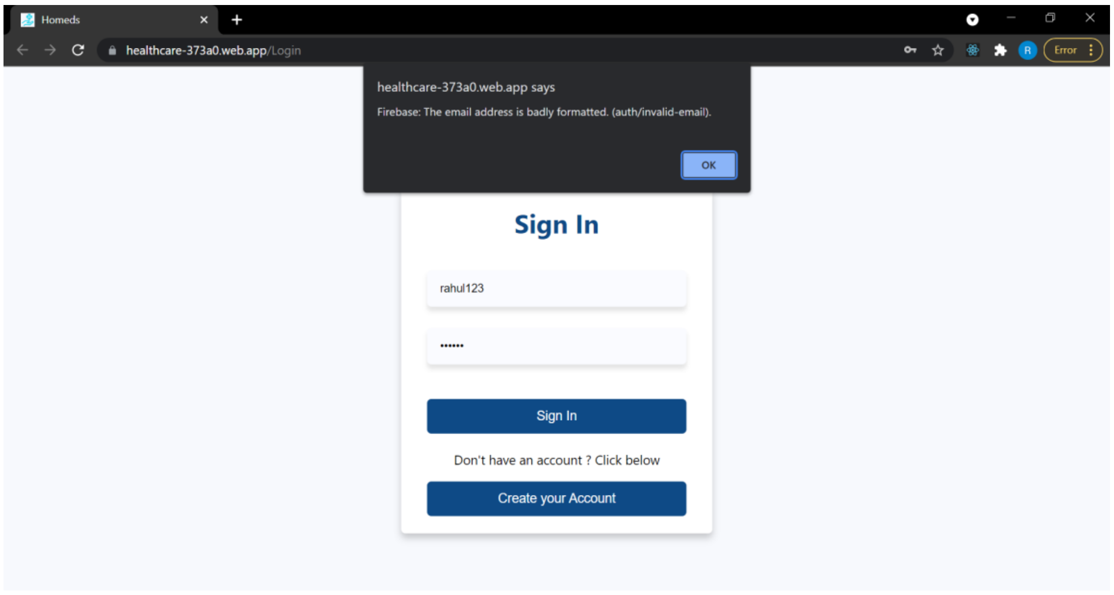

Homeds
healthcare-373a0.web.app/Login
Error
HOMEDS
Sign In
E-mail
Password
Sign In
Don't have an account ? Click below
Create your Account

Fig 20 : Sign In/ Sign Up page

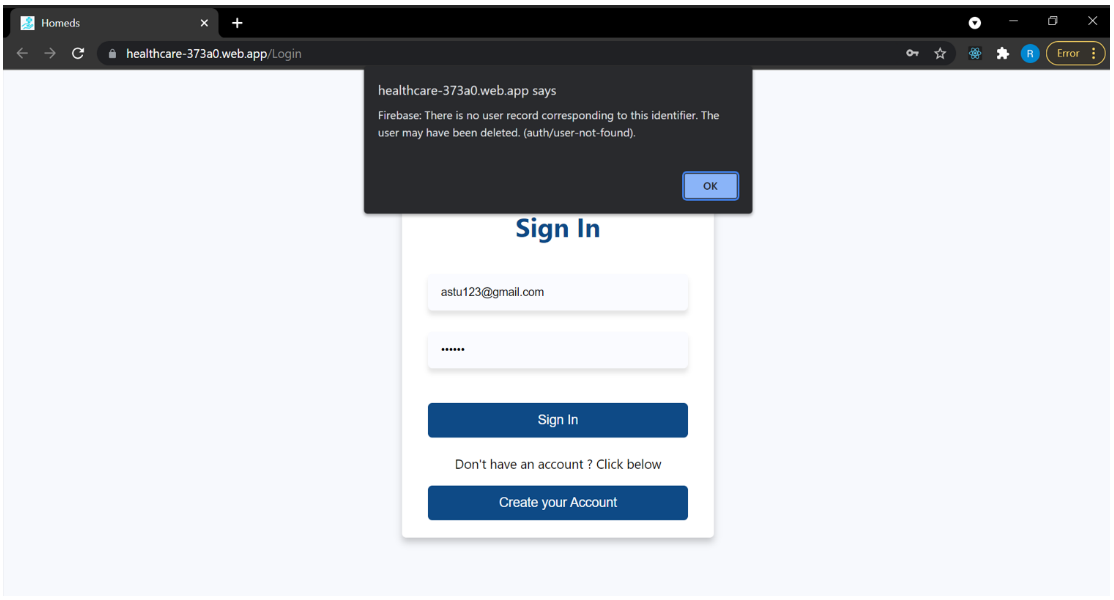
b) In case the user enter a wrong format of credentials or if user attempt to make a new account from a already used Email Id .Following error will be displayed by firebase**(Shown In Fig 21 & 22)**.



healthcare-373a0.web.app says

Firebase: The email address is badly formatted. (auth/invalid-email).

Fig 21 : If user enter Email in wrong format



healthcare-373a0.web.app says

Firebase: There is no user record corresponding to this identifier. The user may have been deleted. (auth/user-not-found).

Fig 22 : If User attempt to sign up using a previously used account

2) Home Page

Once the user is signed in , they will be directed to Home Page**(Shown In Fig 23)**.

a) Navigation bar

On the top is the Navigation bar where the email id through which the user has signed is shown along with a Sign Out Option. Upon clicking on "Categories" option will redirect the customer to the Shop By category section. Total number of items will be shown alongside the cart icon.

b) Shop By Category

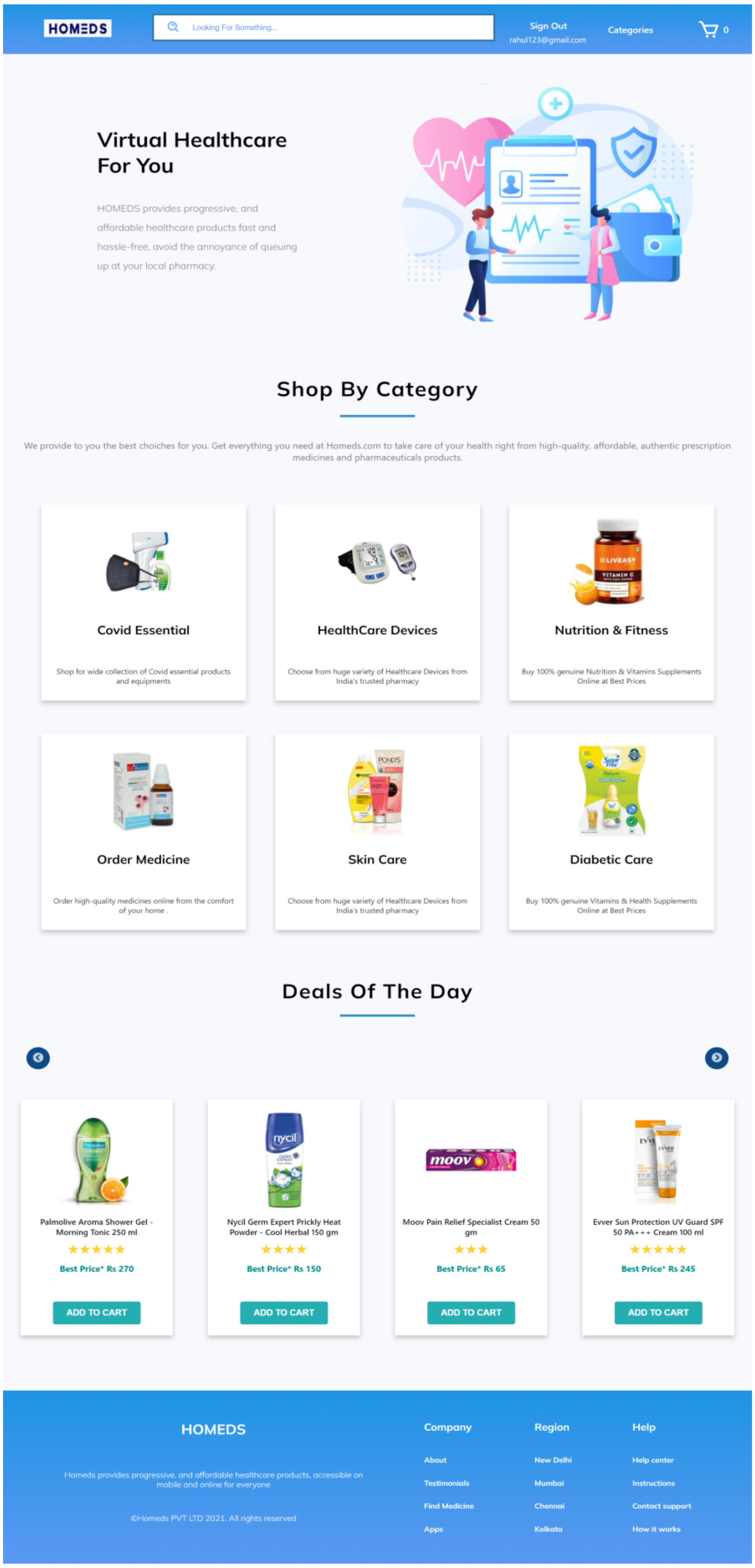
Next in the Home page is the Shop By Category section. This section allows user to shop across a wide selection of products from trusted brands across a range of categories at great prices . Customer can shop from six different categories including Diabetic care , Skin care, Nutrition and vitamins, Covid Essential,Healthcare Devices and can even Order Medicine .

c)Deals of the Day

With this section, our focus is on each and every category. Only the best deals for each category are updated on this section making your shopping a fun and budget friendly experience.

d)Footer

Footer consist of information about the Homeds .All the links related to names of different regions of India where we offer our services. In addition to that all the necessary information regarding the Help center and Instructions on how our site work and the contact to solve the queries of customer are mentioned in the footer.



Virtual Healthcare For You

HOMEDS provides progressive, and affordable healthcare products fast and hassle-free, avoid the annoyance of queuing up at your local pharmacy.

Shop By Category

We provide to you the best choices for you. Get everything you need at Homeds.com to take care of your health right from high-quality, affordable, authentic prescription medicines and pharmaceuticals products.

Covid Essential

Shop for wide collection of Covid essential products and equipments

HealthCare Devices

Choose from huge variety of Healthcare Devices from India's trusted pharmacy

Nutrition & Fitness

Buy 100% genuine Nutrition & Vitamins Supplements Online at Best Prices

Order Medicine

Order high-quality medicines online from the comfort of your home .

Skin Care

Choose from huge variety of Healthcare Devices from India's trusted pharmacy

Diabetic Care

Buy 100% genuine Vitamins & Health Supplements Online at Best Prices

Deals Of The Day

Palmolive Aroma Shower Gel - Morning Tonic 250 ml
★★★★★
Best Price* Rs 270
ADD TO CART

Nycil Germ Expert Prickly Heat Powder - Cool Herbal 150 gm
★★★★
Best Price* Rs 150
ADD TO CART

Moov Pain Relief Specialist Cream 50 gm
★★★
Best Price* Rs 65
ADD TO CART

Evver Sun Protection UV Guard SPF 50 PA+++ Cream 100 ml
★★★★★
Best Price* Rs 245
ADD TO CART

HOMEDS

Homeds provides progressive, and affordable healthcare products, accessible on mobile and online for everyone

©Homeds PVT LTD 2021. All rights reserved

Company	Region	Help
About	New Delhi	Help center
Testimonials	Mumbai	Instructions
Find Medicine	Chennai	Contact support
Apps	Kolkata	How it works

Fig 23 : Home Page

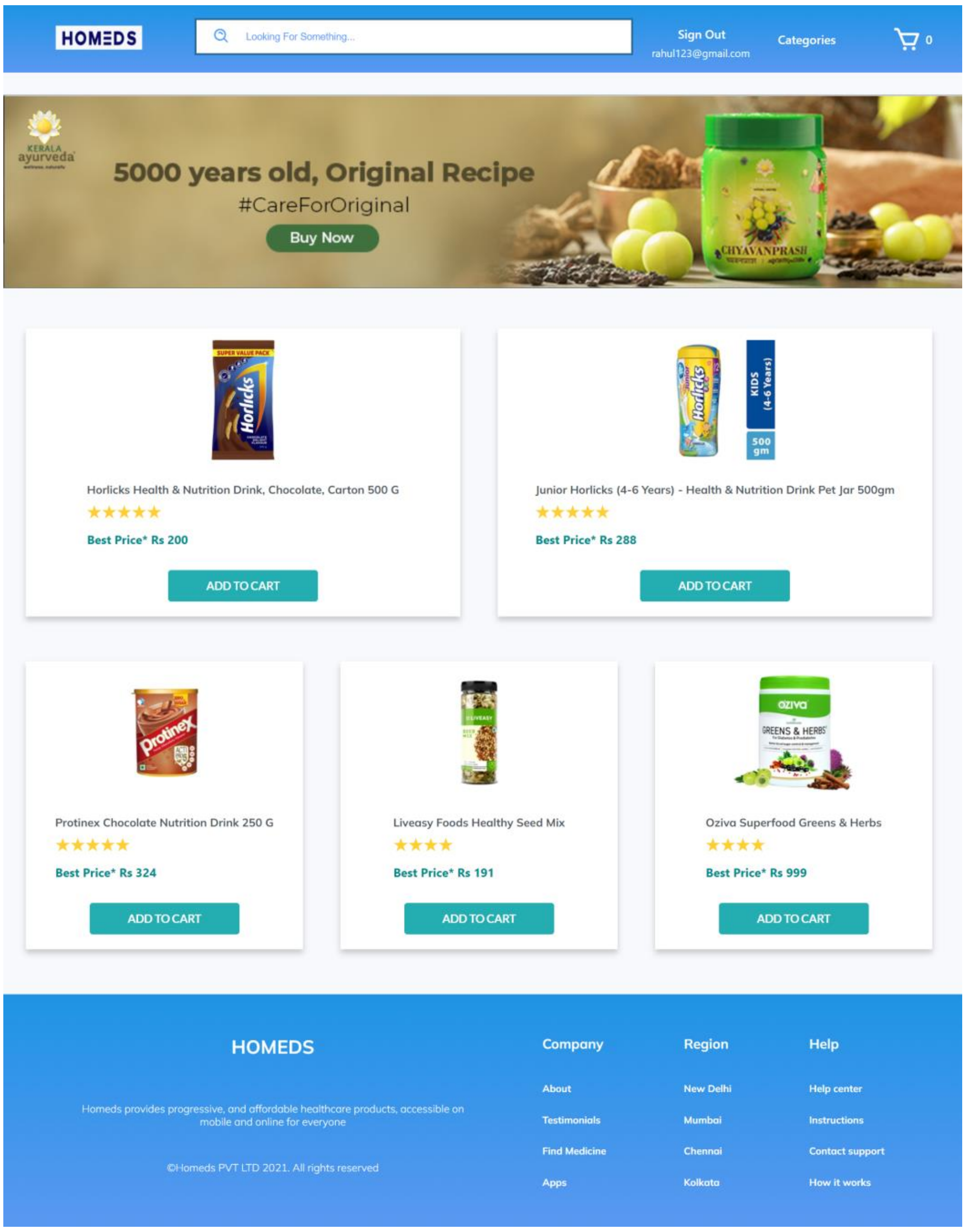
3) Category Pages

Once the user click on any one of the six categories , they will be directed to the following pages from where they can add item to their cart by clicking on "**Add to Cart** "button **(Shown In Fig 24)**.

1) Covid Essential 2) HealthCare Devices

3) Nutrition And Fitness 4) Order Medicine

5)Diabetic Care 6) Skin Care



Horlicks Health & Nutrition Drink, Chocolate, Carton 500 G
★★★★★
Best Price* Rs 200

Junior Horlicks (4-6 Years) - Health & Nutrition Drink Pet Jar 500gm
★★★★★
Best Price* Rs 288

Protinex Chocolate Nutrition Drink 250 G
★★★★★
Best Price* Rs 324

Liveasy Foods Healthy Seed Mix
★★★★
Best Price* Rs 191

Oziva Superfood Greens & Herbs
★★★★
Best Price* Rs 999

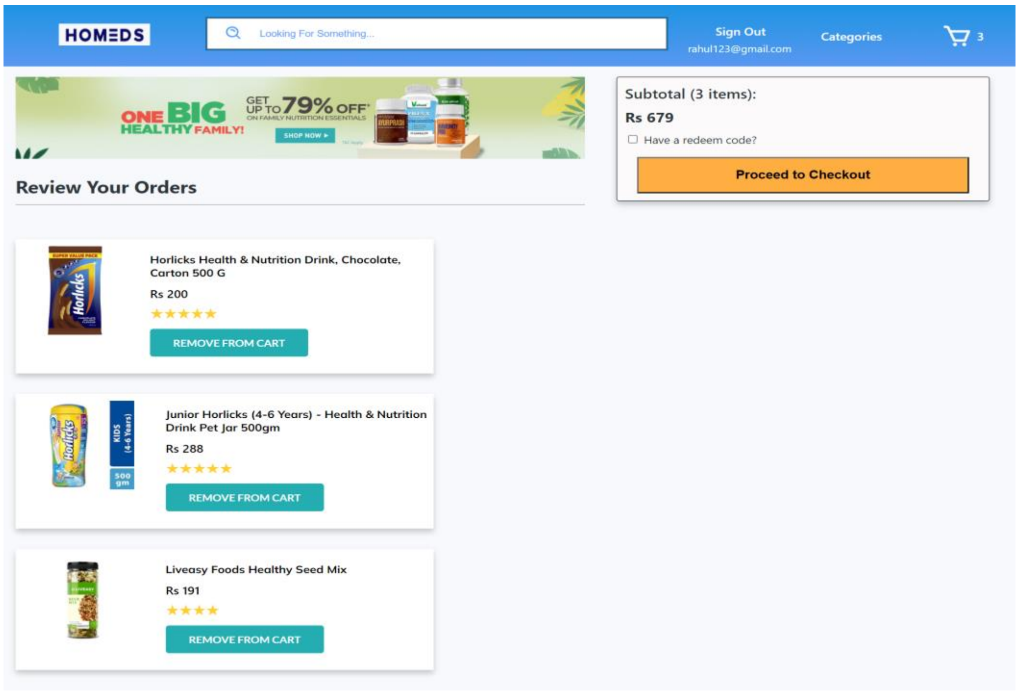
Homeds provides progressive, and affordable healthcare products, accessible on mobile and online for everyone

©Homeds PVT LTD 2021. All rights reserved

Fig 24 : Nutrition And Fitness Pages

4) Checkout Page

When user click on the cart icon in the navbar The users can view all the items that they have added to their cart and the total payable amount and total no of items in the checkout page. In case user thinks the selected item is not useful for him, then he can remove that item form the cart **(Shown In Fig 25 & 26)**..



Subtotal (3 items):
Rs 679
Have a redeem code?
Proceed to Checkout

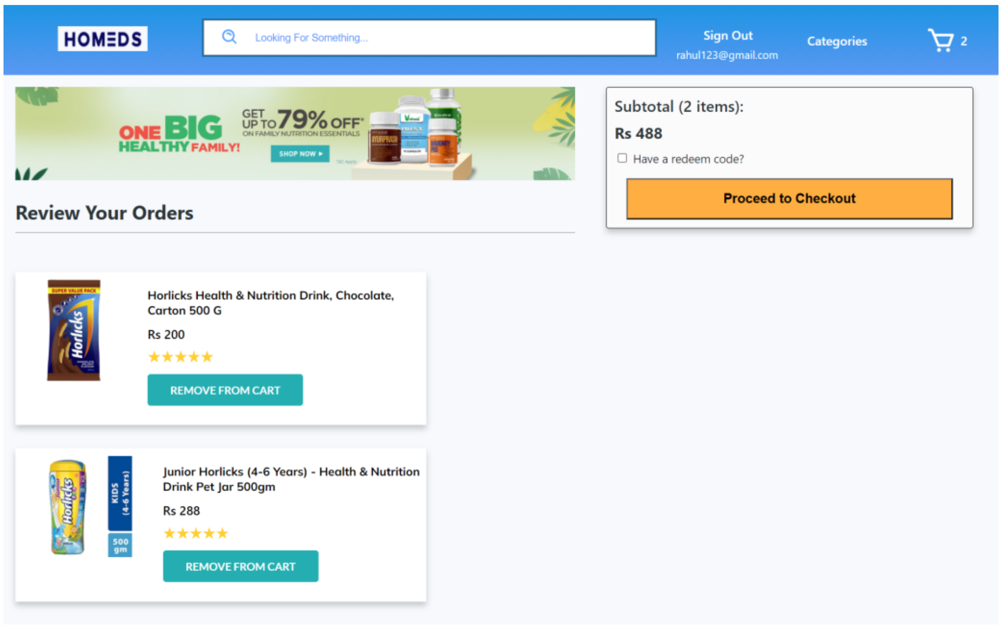
Review Your Orders

Horlicks Health & Nutrition Drink, Chocolate, Carton 500 G
Rs 200
REMOVE FROM CART

Junior Horlicks (4-6 Years) - Health & Nutrition Drink Pet Jar 500gm
Rs 288
REMOVE FROM CART

Liveasy Foods Healthy Seed Mix
Rs 191
REMOVE FROM CART

Fig 25 : Items added in cart



Subtotal (2 items):
Rs 488
Have a redeem code?
Proceed to Checkout

Review Your Orders

Horlicks Health & Nutrition Drink, Chocolate, Carton 500 G
Rs 200
REMOVE FROM CART

Junior Horlicks (4-6 Years) - Health & Nutrition Drink Pet Jar 500gm
Rs 288
REMOVE FROM CART

Fig 26 : Items removed from cart

5) Sign Out

In order to sign out the user have to click the Sign Out button in the navigation bar . Once user have sign out of his account he will be directed to home page and they have to click on the sign in Button in order to Log in their account again**(Shown In Fig 26 & 27)**.

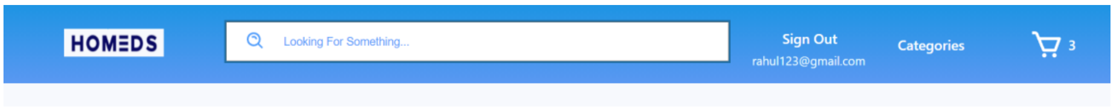
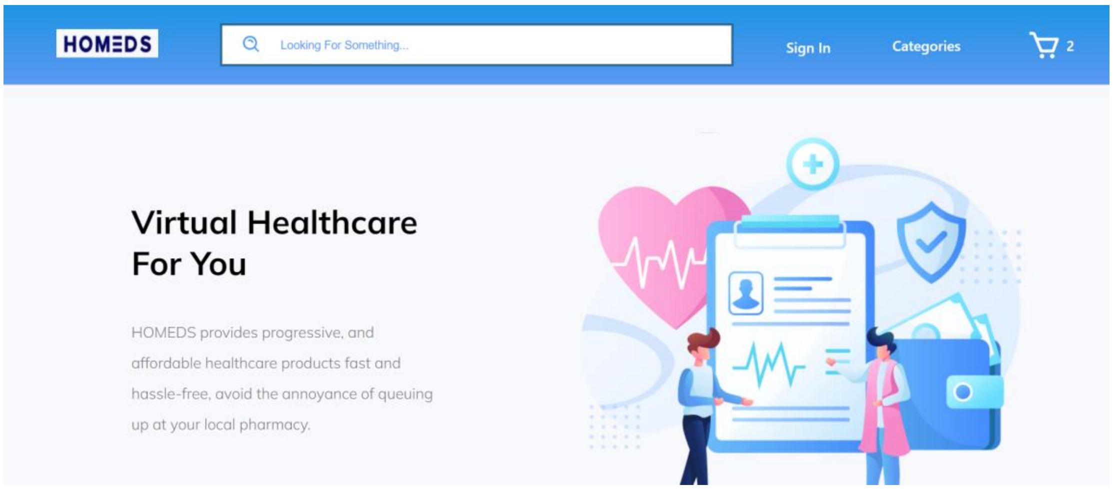


Fig 27 : When user is signed in



Virtual Healthcare For You

HOMEDS provides progressive, and affordable healthcare products fast and hassle-free, avoid the annoyance of queuing up at your local pharmacy.

Fig 28 : :When user is signed out

CHAPTER-5:
CONCLUSION AND FUTURE SCOPE

5.1 Future Scope And Benefits

1) Convenience: It is easy to order online medicines and healthcare. Individuals who live great distances from a terrestrial pharmacy, the elderly, disabled persons, and those whose daily schedule includes additional hassles, all can benefit from the easy and fast purchasing of medicines online. Additionally, shipping charges are often less than the
expense incurred from travel costs associated with a visit to a traditional pharmacy.

2) Time Saving: In just a matter of minutes, you can log into the internet, head over to an E-pharmacy website and buy any prescription medicine and other healthcare products that you want. It takes less time and effort than travelling to your local pharmacy and having to wait in line for your medicine.
3)**Range of Choices:** It offers a great variety of options. They usually have larger stocks of drugs than a physical pharmacy
Online pharmacies and healthcare products web app offer better pricing than offline stores, with increased access, lower transaction and product costs, convenience and greater anonymity for consumers. They offer accessibility to people with limited mobility and people in remote areas. Therefore considering all the above mentioned benefits.we can say that the concept used in this project would be an asset to people as well as market.Also each and every module of the application has been written in a modular way, which makes it easy to upgrade.\

5.2 Conclusion

All aforementioned tasks in the objectives have been completed in the project and corresponding results and outputs have been attached.All the objectives were completed in the given frame of time. All the functionality are tested and seem to be working fine.

Bibliography

- **React :** https://reactjs.org/docs/
- **React Tutorial W3Schools :**

  https://www.w3schools.com/REACT/DEFAULT.ASP
- **ReactJS Tutorial - javatpoint :** https://www.javatpoint.com/react-introduction
- **React Context API :**

  https://reactjs.org/docs/context.html
- **React Slick:**

  https://react-slick.neostack.com/
- **Firebase Documentation :**

  https://firebase.google.com/docs/guides