FULL-STACK-2 PROJECT

(2020-21)

E-commerce Website

Institute of Engineering & Technology

Team Members

Aman Gulwani

` **(181500804)**

Vishal Verma

(181500811)

Supervised By

Mr. Pankaj Kapoor

Department of Computer Engineering & Applications

TABLE OF CONTENTS

Declaration **4**

Certificate **5**

Acknowledgement **5**

Introduction **7**

Overview and Motivation 7

Introduction to Problem Statements 9

Objectives- 9

System Requirement Analysis **10**

System Analysis 10

Role of System Analyst 10

Users 11

Methodology 11

Requirement Analysis 12

Website Handover: 12

Dependencies/External Systems 13

User Interface: 13

Hardware Interface: 13

Hardware and Software Requirements **13**

Hardware Requirements: 13

Software Requirements: 14

Implementation Details **14**

Contribution Summary 15

Project Work **15**

OUTPUT SCREENS AND SOURCE CODE: **16**

Output Screens : 16

Source code : 18

Benefits and scope of the project **33**

Benefits of the Project 33
Future Scope 33
Limitations of System Proposed 34
Conclusion **34**

Declaration

I hereby declare that the work which is being presented in the Full stack MERN technologies Project "**E-Commerce Website**", in partial fulfillment of the requirements for Full stack MERN technologies is an authentic record of my own work carried under the supervision of **Mr. Pankaj Kapoor.**

Vishal Verma

Aman Gulwani

Certificate

This is to certify that the project entitled “E-Commerce Website” carried out in Full Stack using MERN technologies is a bonafide work done by Vishal Verma (181500811) and Aman Gulwani (181500069) and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Mr. Pankaj kapoor

Date:10/04/2021

Vishal Verma

Aman Gulwani

Acknowledgement

Presentation, inspiration, and motivation have always played a key role in the success of any venture.

Making this project was a great learning experience. During this period of building the project, we learned a lot about web development as well as team work.

We would like to express our special thanks of gratitude to Mr. Pankaj Kapoor Sir for guiding and teaching us throughout this project. We pay our deep sense of gratitude to the GLA University, Mathura for conducting this project.

We are grateful to our parents who have been an important inspiration. We want to thank them for their encouragement which helps us in completion of this project.

Abstract

This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet by using digital devices. Thus the customer

will get the service of online shopping and home delivery from his favorite shop. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops. Since the application is available in the Smartphone it is easily accessible and always available.

This project aims to synthesize goal-orientated processes such as problem solving, decision-making, communication and to solve time consuming tasks in less time.

This convenient system of working will take all the major concerns like getting out of the home, high prices etc. which helps in smooth functioning of the system which in turn leads to better consequences .This system is constructed using Fullstack mern technology to provide a user-friendly interface.

This system's back-end is constructed using PHP and MySQL to ensure reliable database and its front-end is designed using HTML5, CSS3, Bootstrap and JavaScript to provide a user-friendly interface.

Introduction

Overview and Motivation

This report file gives the complete description of the project titled as “E-Commerce”. It will illustrate various functionalities of an e-commerce website. This project is a web based shopping system for an existing shop. The project objective is to deliver the online shopping application into android platform. Online shopping is the process whereby consumers directly buy goods or services from a seller in real-time, without an intermediary service, over the Internet. It is a form of electronic commerce. This project is an attempt to provide the advantages of online shopping to customers of a real shop. It helps buying the products in the shop anywhere through internet. Thus the customer will get the service of online shopping and home delivery from his favorite shop .The project aims to synthesize goal-orientated processes such as problem solving, decision-making, communication and to solve time consuming tasks in less time.

The central concept of the application is to allow the customer to shop Virtually using the Internet and allow customers to buy the items and articles of their desire from the store.

The information pertaining to the products are stores on an RDBMS at the server side (store). The Server process the customers and the items are shipped to the address submitted by them. The application was designed into two modules first is for the customers who wish to buy the articles. Second is for the storekeepers who maintains and updates the information pertaining to the articles and those of the customers. The end user of this product is a departmental store where the application is hosted on the web and the administrator maintains the database.

The application which is deployed at the customer database, the details of The items are brought forward from the database for the customer view based on the selection through the menu and the database of all the products are updated at the end of each transaction. Data entry into the application can be done through various screens designed for various levels of users. Once the authorized personnel feed the relevant data into the system, Several reports could be generated as per the security.

Introduction to Problem Statements

As online shopping became a trend nowadays the regular shops are losing their customers to online brands. Customers have effortless shopping experience and saving time through shopping online. For competing with those online brands , If shops are providing an online portal where their customers can shop through internet and get the products at their doors it will increase the number of customers.

Objectives-

The objective of the project is to make an application in android platform to purchase items in an existing shop. In order to build such an application complete web support need to be provided. A complete and efficient web application which can provide the online shopping experience is the basic objective of the project. The web application can be implemented in the form of an android application with web view..

System Requirement Analysis

System Analysis

System analysis is the process of gathering and interpreting facts, diagnosing problems and using the information to recommend improvements on the system. System analysis is a problem solving activity that requires intensive communication between the system users and system developers.

System analysis or study is an important phase of any system development process. The system is viewed as a whole, the inputs are identified and the system is subjected to close study to identify the problem areas. The solutions are given as a proposal. The proposal is reviewed on user request and suitable changes are made. This loop ends as soon as the user is satisfied with the proposal.

Role of System Analyst

The system analyst is a person who is thoroughly aware of the system and guides the system development project by giving proper directions. She/he is an expert having technical and interpersonal skills to carry out development tasks required at each phase. Some of the main roles of system analyst are:

- Defining and understanding the requirements of users through various fact finding techniques.
- Prioritizing the requirements by obtaining user consensus.
- Maintains analysis and evaluation to arrive at appropriate system which is more user friendly.
- Draw certain specifications which are easily understood by users and programmers in precise and detailed form.
- Implement the logical design of a system which must be modular.

Users

A user is a person who uses or operates the application. The user of the web portal will be students , employees and Admin of this portal only.

Methodology

Our methodology is designed in such a manner to help you take maximum advantage of the internet technologies. It incorporates all aspects related to our website and allows us to ensure that the final product is of the highest standards. Below are the steps we will take to ensure that all your deliverables are completed in time, also we will try to solve each and every problem efficiently.

Requirement Analysis

The first step for us is to analyze our target market‟s requirements. Who will be visiting our website, what will be the purpose of their visit, what is the primary goal of our website, how can our team best cater to their needs etc. Many such questions are analyzed for the Needs
Analysis stage. If we are given access to the current website statistics, we would also like to analyze our current page views, average user time spent on the site, top landing pages, existing search engine rankings, existing bounce rates and many such factors. We analyze our online target audience and assess our differentiation strategy to best attract and retain our online visitors.

Website Handover:

Once the site is set up on the destination server, upon the approval we make the website live. One final set of testing is done on the live website for the quality assurance purposes, we then hand over the control of the site to admin.

In this project the work is divided into 5 modules. In each module we will be performing the different tasks to reach the target. Modules are as follows. :

- Project software and hardware setup and designing the homepage of Website.
- Creating the layouts of the pages.
- Backend task will be the fourth module.
- Last module is the backup module to look back at the project working and make it better.

Dependencies/External Systems

User Interface:

The webpages of E-commerce website will permit complete navigation; including all the functionalities, starting from a simple login or registration, to access the necessary information.

Hardware Interface:

There are external machines/devices used by the portal, each related to a user interface. These are a server machine at the admin end and hosting the portal, a PC keeping log of the registration and login entries made to them & providing them with an interface to do their part of functionality offered, last one is a PC or a smartphone or a tab at the user's end to access the portal. The devices at the user end behave as terminals and not for storing any type of data. Also capable of taking user input. All order and decisions should be stored on the server.

Hardware and Software Requirements

Hardware Requirements:

- Computer with RAM greater than 2 GB.
- Computer with i3 or above versions of Intel processors.

- Active internet connection provider Wi-Fi module.

Software Requirements:

- Windows 7 or above versions
- Visual Studio Code
- Web Browser
- React requirements(npm installed)
- Github
- Adobe XD

Implementation Details

To implement is defined as to put something in effect.
We have designed the website‟s front-end using HTML5, CSS3, Bootstrap, and some basics of JavaScript to provide a user-friendly interface. And its back-end is constructed using PHP and MySQL to ensure a reliable database. To make implementation details more clear here are

some of the highlighting points:

- To implement homepage in an efficient way.
- To enable users to add their details on a website.
- To enable users to maintain and update their details.
- To make the complete login/sign-up process easy.
- To make the complete website process easy.

Contribution Summary

This project is completed by 2 members of the team. So, their contributions are:

- **Aman Gulwani:-** Front-end, back-end,different view panels, synopsis and video file of the project along with project handler.
- **Vishal Verma:-** Front-end ,back-end ,data collector ,presentation file and report file of the project.

Project Work

The online E-commerce website is built both for desktop as well as smartphones. However it is also responsive for other digital devices as well like tabs, etc. Front-end of the website is built using HTML5, CSS3, Bootstrap and some basics of JavaScript to provide a user friendly interface. Its back-end is constructed using PHP and MySQL to ensure a reliable database. Website has many parts to showcase and receive information.Here in this project our website will provide information of the products with images and the cart option to cart the products.

Along with this our website provides easy way of access to the website where one can buy product easily.

The homepage of the website is the most important page as well as the first impression of the website. Here in the project homepage of the website is strikingly impressive. Anyone who comes to the website for fetching details or login/signup will be welcome by the homepage given below.

OUTPUT SCREENS AND SOURCE CODE:

Output Screens :

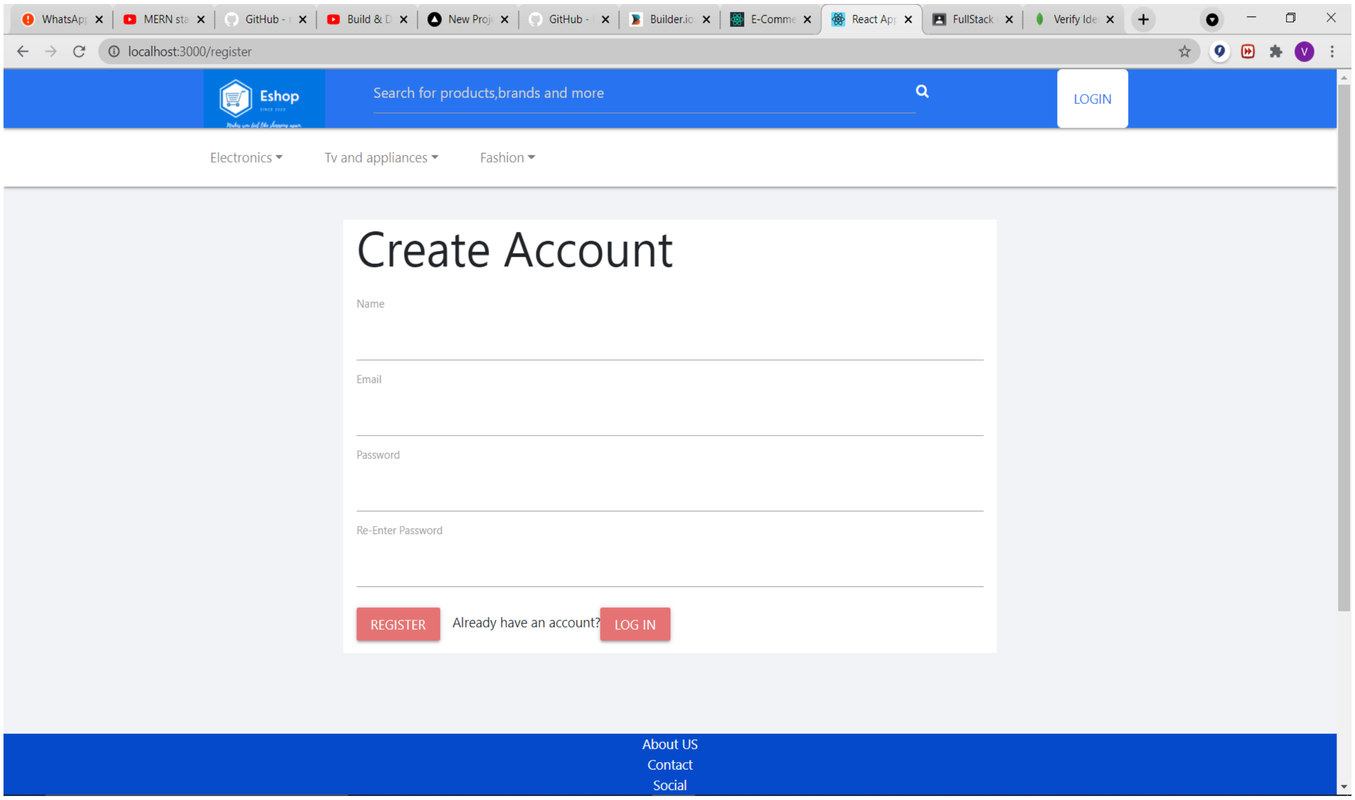

Fig - 1.1Create account page

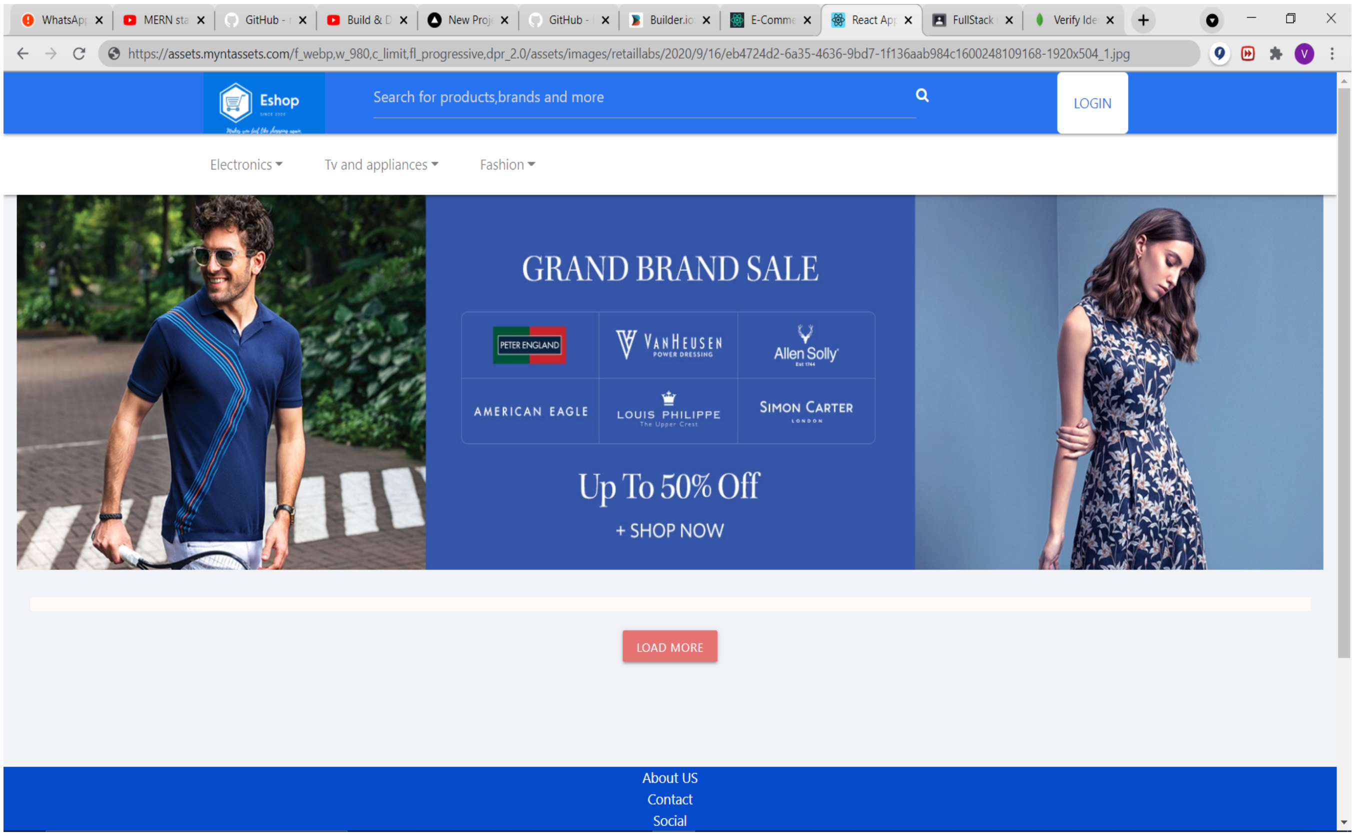

Fig - 1.2 Home page

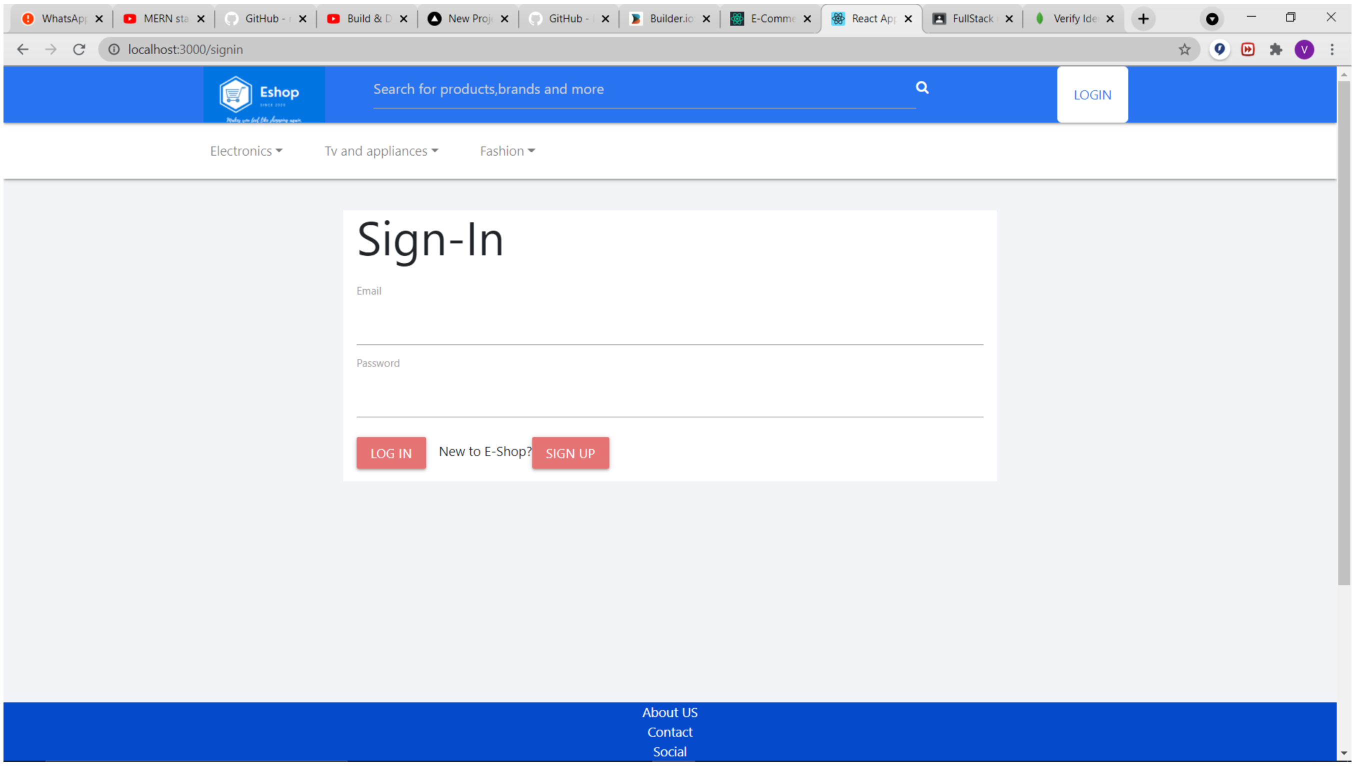

Fig - 1.3 Sign-in page

Source code :

```
<!DOCTYPE
html>
             <html lang="en">
              <head>
                <meta charset="utf-8" />
                <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
                <meta name="viewport" content="width=device-width, initial-scale=1" />
                <meta name="theme-color" content="#000000" />
                <meta
                  name="description"
                  content="Web site created using create-react-app"
```

```
   />

   <link
href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min
.css" rel="stylesheet" />
   <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />

   <!--

     manifest.json provides metadata used when your web app is installed on a

     user's mobile device or desktop. See
https://developers.google.com/web/fundamentals/web-app-manifest/
   -->

   <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />

   <!--

     Notice the use of %PUBLIC_URL% in the tags above.

     It will be replaced with the URL of the `public` folder during the
build.
     Only files inside the `public` folder can be referenced from the HTML.


     Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will

     work correctly both with client-side routing and a non-root public URL.

     Learn how to configure a non-root public URL by running `npm run build`.

   -->

   <title>React App</title>

 </head>

 <body style="background-color: #f1f3f6;">

   <noscript>You need to enable JavaScript to run this app.</noscript>

   <div id="root"></div>

   <!--

     This HTML file is a template.

     If you open it directly in the browser, you will see an empty page.


     You can add webfonts, meta tags, or analytics to this file.
     The build step will place the bundled scripts into the <body> tag.


     To begin the development, run `npm start` or `yarn start`.

     To create a production bundle, use `npm run build` or `yarn build`.

   -->

 </body>

</html>
```

Fig - 1.4 index.html file

```
import React, {
Component } from
'react';
                    import Navbar from 'react-bootstrap/Navbar';

                    import Nav from 'react-bootstrap/Nav';

                    import {Badge} from 'antd';

                    import { Link } from 'react-router-dom';

                    import './style.css';

                    import { connect } from 'react-redux';

                    import * as Icon from 'react-bootstrap-icons';

                    import Axios from 'axios';

                    import { logout } from '../../store/actions/userActions';

                    import { withRouter } from "react-router";

                    import logo from './logo.jpg' ;



                    class Header extends Component {

                        constructor(props) {

                            super(props)

                            this.state = {
                            };
                        }


                       logoutHandler = () => {
                           Axios.get(`http://localhost:8000/api/users/logout`)
                           .then(response => {
                                if (response.status === 200) {
                                    this.props.history.push("/login");
                                } else {
                                    alert('Log Out Failed')
                                }
                            });
                        };
                       handleLogout = () => {
                            this.props.dispatch(logout());
                            this.props.history.push("/signin");
                        };




                       render() {
                           console.log('navbar',this);
                           return (
                               <Navbar expand="lg" className="navbar01 " fixed="top">
                                   <Link to="/home" className="N01" id="Logo" ><img
                    id='Logo-pic' src={logo} alt='Eshop' width='100%' /></Link>
                                   <Navbar.Toggle aria-controls="responsive-navbar-nav"
                    className="navbar-toggler01"/>
                                   <Navbar.Collapse id="responsive-navbar-nav">
```

```
                       <div className="input-group">

                           <input className="form-control22"
type="text" placeholder="Search for products,brands and more"
aria-label="Search"></input>
                           <div className="input-group-prepend">

                               <span><i className="fa fa-search"
aria-hidden="true"></i></span>
                           </div>

                       </div>

                   {this.props.userSignin.userInfo ? (

                           <div className='box_opt'>

                               <Link to='/profile'
className="login-btn">{this.props.userSignin.userInfo.name} </Link>
                               {/* <button type="button"
onClick={this.handleLogout} className="login-btn">Logout</button>
*/}
                               <Link to='/cart' className="login-btn"
>
                                   <Badge

                                   // count={this.props.user.user &&
this.props.user.user.cart.length}
                                   >

                                       <Icon.Cart size={25} />

                                   </Badge>

                               </Link>

                           </div>

                           )

                           : (


                               <Link to='/signin'
className="login-btn">LOGIN</Link>
                           )

                   }
                   {this.props.userSignin.userInfo &&
this.props.userSignin.userInfo.isAdmin && (


                       <Link to='/upload'
className="login-btn"><Icon.Upload size={25} /></Link>
                   )}



               </Navbar.Collapse>

           </Navbar>

       )

   }

}



const mapStateToProps = (state) => {

   console.log("state", state);

   return {

       userSignin: state.userSignin,

       user: state.user

   }

}



const ShowTheLocationWithRouter = withRouter(Header);



export default connect(mapStateToProps,
null)(ShowTheLocationWithRouter);
```

Fig - 1.5 Header.js file

```
import React, {
Component } from
'react';
                       import Navbar from 'react-bootstrap/Navbar';

                       import Nav from 'react-bootstrap/Nav';

                       import NavDropdown from 'react-bootstrap/NavDropdown';

                       import { Link } from 'react-router-dom';

                       import './style.css';



                       class SubHeader extends Component {

                          render() {

                              return (

                                  <Navbar expand="lg" className="navbar02">

                                      <Nav className="mr-auto sub-nav">

                                          <NavDropdown title="Electronics"
                       className="drop">
                                              <NavDropdown.Item > <Link className='text'
                       to='/Mobile'>Mobile</Link></NavDropdown.Item>
                                              <NavDropdown.Item > <Link className='text'
                       to="/Laptop">Laptops</Link> </NavDropdown.Item>
                                              <NavDropdown.Item > <Link className='text'
                       to="/Camera">Camera</Link></NavDropdown.Item>
                                              <NavDropdown.Item > <Link className='text'
                       to="/Tablets">Tablets</Link></NavDropdown.Item>
                                          </NavDropdown>

                                          <NavDropdown title="Tv and appliances"
                       className="drop">
                                              <NavDropdown.Item ><Link className='text'
                       to="/TV">TV</Link></NavDropdown.Item>
                                              <NavDropdown.Item ><Link className='text'
                       to="/Washing-machine">Washing machine</Link></NavDropdown.Item>
                                              <NavDropdown.Item ><Link className='text'
                       to="/Air-Conditioners">Air Conditioners</Link></NavDropdown.Item>
                                              <NavDropdown.Item ><Link className='text'
                       to="/Kitchen-appliances">Kitchen
                       appliances</Link></NavDropdown.Item>
                                          </NavDropdown>

                                          <NavDropdown title="Fashion" className="drop">



                                          </NavDropdown>

                                      </Nav>

                                  </Navbar>

                              )

                          }

                       }



                       export default SubHeader;
```

Fig - 1.6 Subheader.js file

```
import React, { Component }
from 'react';
                               import { connect } from 'react-redux';

                               import { Link} from 'react-router-dom';

                               import './style.css';

                               import 'materialize-css/dist/css/materialize.min.css';

                               import user_actions from
                               "../../store/actions/user_actions";





                               // const Promise = global.Promise;



                               class Login2 extends Component {
```

```
   constructor(props){
       super(props)


           this.state = {
               email: '',
               password: '',
               errors: [],
           };


   }





   dispalyErrors = errors =>
       errors.map((error, i) => <p key={i}> {error} </p>)




   handleChange = event => {
       this.setState({[event.target.name]:
event.target.value})
   }


   submitForm = event => {
       event.preventDefault();


       let dataToSubmit = {
           email: this.state.email,
           password: this.state.password
       };


       if (this.isFormvalid(this.state)) {
           this.setState( { errors: [] } )

this.props.dispatch(user_actions.loginUser(dataToSubmit))
               .then(response => {
                   console.log('res',response)
                   if (response.payload.isAuth){
                           alert('login suucefully')
                           this.props.history.push('/')
                           }


                   else {
                       alert('Failed to login... Wrong
email or password')
                       this.setState({
                           errors:
this.state.errors.concat(
                               "Failed to login... Wrong
email or password"
                           )
                       })
                   }
               }
           )



           // try {
           //     console.log('login',this.props)

           //
this.props.dispatch(user_actions.loginUser(dataToSubmit))
           //         if(this.props.user.loginSuccess =
{loginSuccess:true} ){
           //             this.props.history.push('/')

           //         }

           //         else {

           //             this.setState({

           //                 errors:
this.state.errors.concat(
           //                     "failed to login"

           //                 )

           //             })

           //         }



           // }

           // catch(errors){

           //     console.log('errorsss',errors)

           //      this.setState({

           //                  errors:
this.state.errors.concat(
           //                      "fail login"

           //                  )

           //              })

           // }

       }

       else {

           alert('Fill all the details')

           this.setState({

               errors: this.state.errors.concat('Fill all
the details')
```

```
           })
       }
   }


   isFormvalid = ({email, password}) => email && password;




   render() {
       console.log('loginpage', this.props.user);
       return (
           <div className="container login-box">
               <h2>Log in</h2>
               <div className='row'>
                   <form className='col-12' >
                       <div className='row'>
                           <div className='input-feild col
s12'>
                               <label
htmlFor="email">Email</label>
                               <input
                                   name="email"
                                   value={this.state.email}
                                   onChange={e =>
this.handleChange(e)}
                                   id="email"
                                   type="email"
                                   className='validate'
                                   autoComplete='on'
                               />
                               <span
                               className='helper-text'
                               data-error='type a right
email'
                               data-success='right'
                               />

                           </div>
                       </div>
                       <div className='row'>
                           <div className='input-feild col
s12'>
                               <label
htmlFor="password">Password</label>
                               <input
                                   name="password"

value={this.state.password}
                                   onChange={e =>
this.handleChange(e)}
                                   id="password"
                                   type="password"
                                   className='validate'
                                   autoComplete='on'
                               />

                               <span
                                   className='helper-text'
                                   data-error='Wrong
password'
                                   data-success='right'
                               />
                           </div>
                       </div>


                           { this.state.errors.length > 0
&& (
                                   <div>

{this.dispalyErrors(this.state.errors)}
                                   </div>


                               )}




                       <div className='row'>
                           <div className='col s6'>
                               <button className='btn
waves-effect red lighten-2'
                                       type='submit'
                                       name='action'

onClick={this.submitForm}
                               >
                                   Login
                               </button>
                               &nbsp;
                               &nbsp;
                               <Link to='/register'>
                                   <button className='btn
waves-effect red lighten-2'
                                       type='submit'
                                       name='action'
                                   >
                                       Sign Up
                                   </button>
                               </Link>
                           </div>


                       </div>




                   </form>
               </div>

           </div>
       )
   }
}


const mapStateToProps = (state) => {
   console.log("state", state);
   return {
       user: state.user,
   }
}


// const mapDispatchToProps = (dispatch) => {
//     return {
//         loginUser: (dataToSubmit) => {
//             dispatch(loginUser(dataToSubmit))
//         }

//     }

// }


export default connect(mapStateToProps, null) (Login2);
```

Fig - 1.7 Login.js file

Benefits and scope of the project

Benefits of the Project

Project benefits are the advantages and gains that are delivered by a project. The project “E-commerce‟ is built to bring benefits for the people who are willing to get the best and inrange products easily. So, to improve the process of online marketing and online shopping this project provides the following benefits:-

- Easy to buy any product.
- Fast and effective way to choose your desired product.
- Easy interface.
- Totally free of cost.

Future Scope

Project already provides many features which are valuable and are enough in most of the cases but improvements are always welcome. So to make the project more valuable than now more features can be added in future like editing and modifying the existing resumes or cv .

For example in most cases students have theirs but they are not update or interactive then modifying the existing resumes will aid the student or employees a lot.

Or in future we can add a chat-bot intelligence through which one would be able to easily correct their resumes that will play a major role in it.

Limitations of System Proposed

- This project is implemented for frontend only, backend is an optional part.
- The page layouts are designed of limited logos of companies or universities only.

Conclusion

Our intention of this Project was to establish a contrasting, economical, easy and most importantly, a better way to provide an easy access to the user to buy the product which can solve the shopping problems in pandemic. Our proposed system for E-commerce website which comes under the field of Frontend web development. Our main objective was to design a website of E-commerce is for easily availing the facility for the citizens of the country.

References

- https://www.w3schools.com/
- https://www.beta-labs.in/
- www.Youtube.com

****