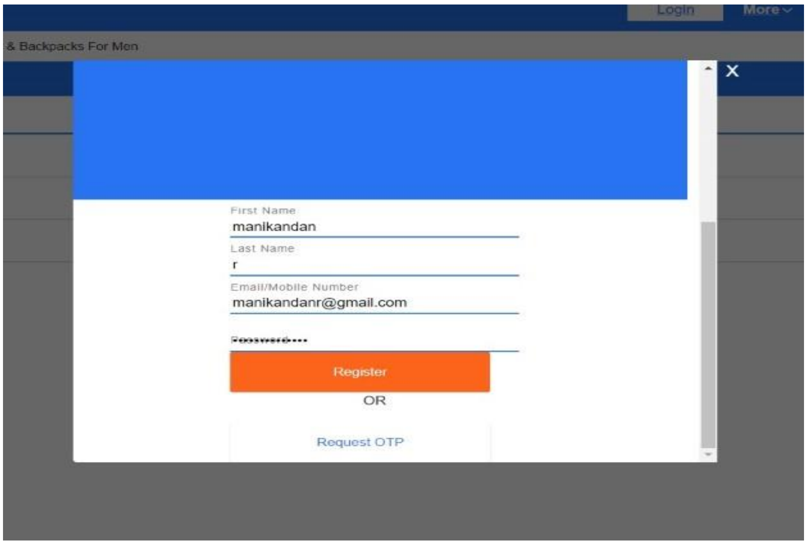
PROJECT DEVELOPMENT PHASE

SPRINT – 1

DATE	17 NOVEMBER 2022
TEAM ID	PNT2022TMID02427
PROJECT NAME	SMART FASHION RECOMMENDER APPLICATION

LOGIN :



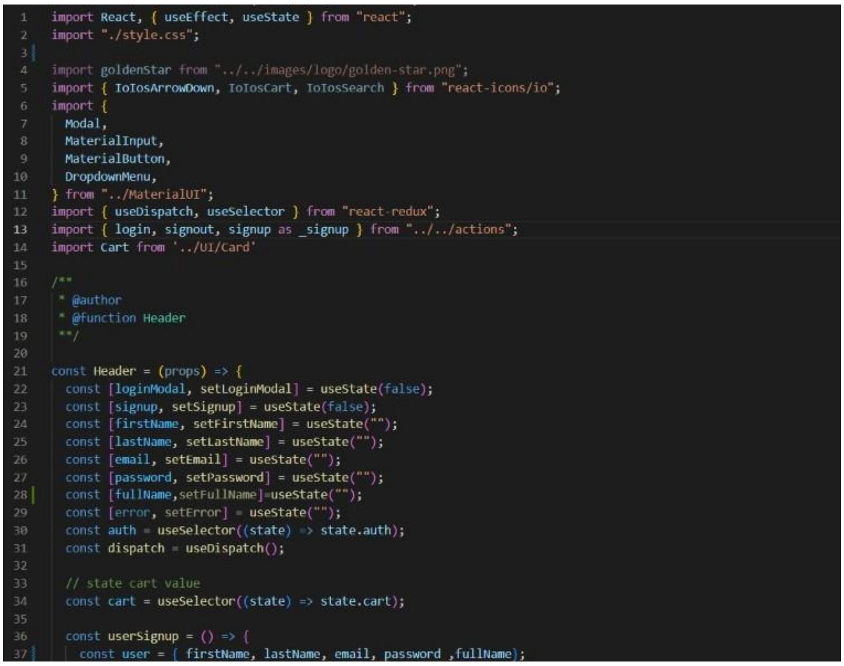
First Name: manikandan

Last Name: r

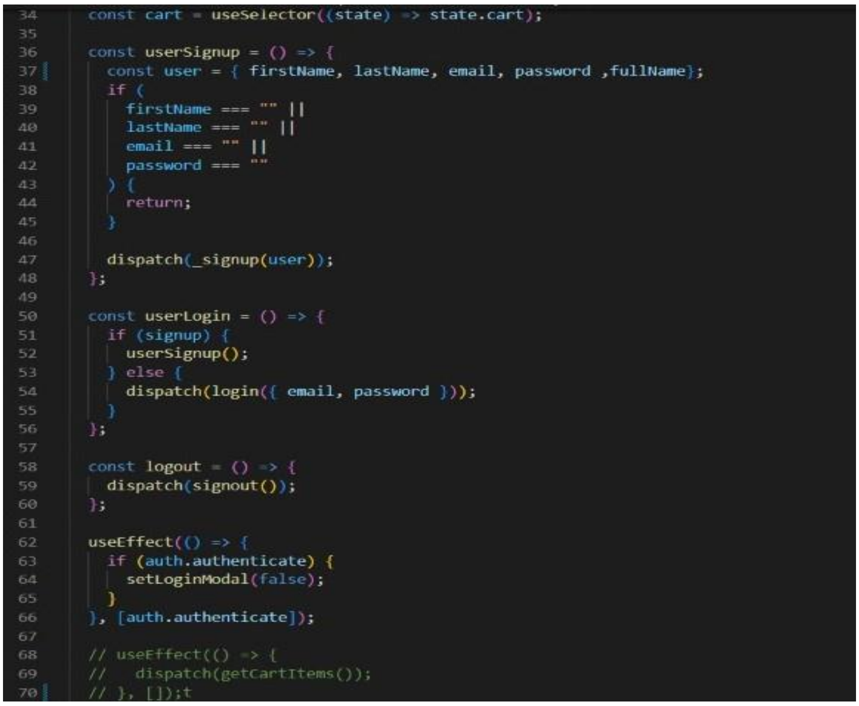
Email/Mobile Number: manikandanr@gmail.com

Password••••

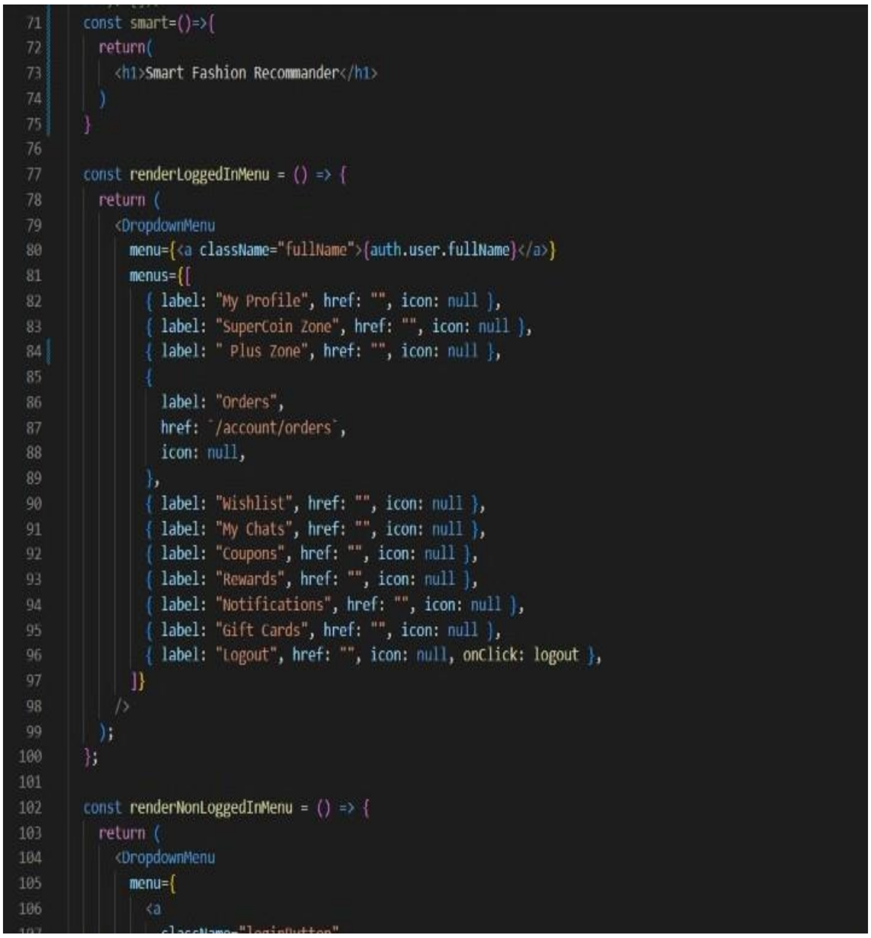
CODE:



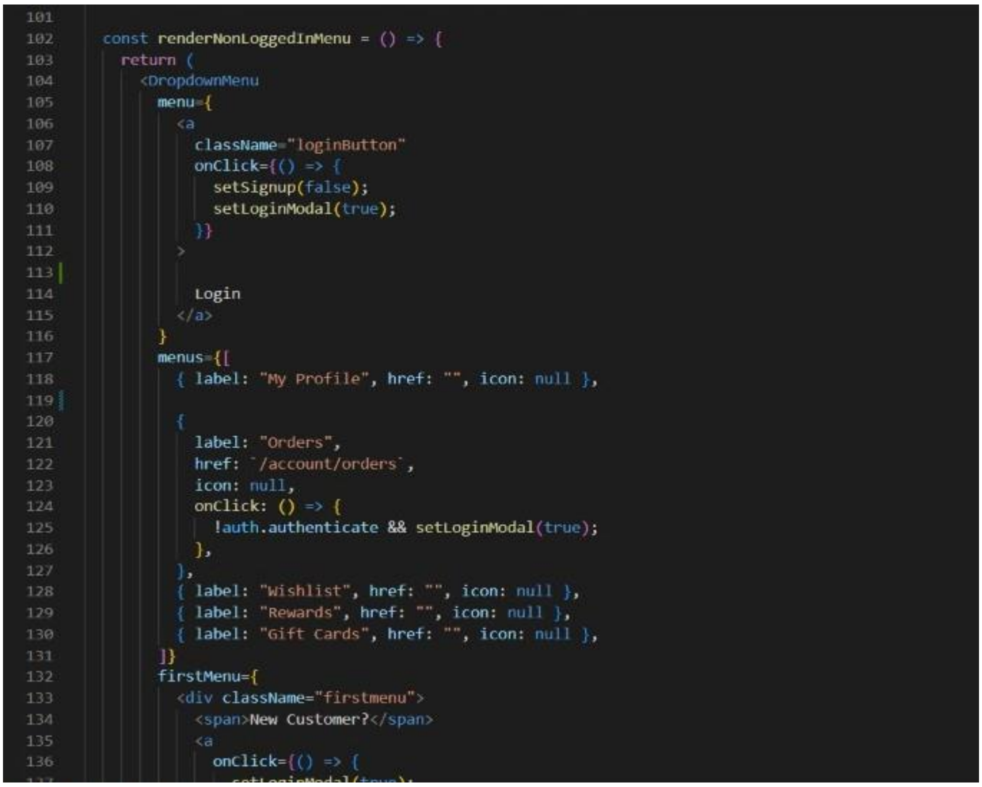
```
import React, { useEffect, useState } from "react";
import "./style.css";

import goldenStar from "../../images/logo/golden-star.png";
import { IoIosArrowDown, IoIosCart, IoIosSearch } from "react-icons/io";
import {
  Modal,
  MaterialInput,
  MaterialButton,
  DropdownMenu,
} from "../MaterialUI";
import { useDispatch, useSelector } from "react-redux";
import { login, signout, signup as _signup } from "../../actions";
import Cart from '../UI/Card'

/**
 * @author
 * @function Header
 **/

const Header = (props) => {
  const [loginModal, setLoginModal] = useState(false);
  const [signup, setSignup] = useState(false);
  const [firstName, setFirstName] = useState("");
  const [lastName, setLastName] = useState("");
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [fullName,setFullName]=useState("");
  const [error, setError] = useState("");
  const auth = useSelector((state) => state.auth);
  const dispatch = useDispatch();

  // state cart value
  const cart = useSelector((state) => state.cart);

  const userSignup = () => {
    const user = { firstName, lastName, email, password ,fullName};
```



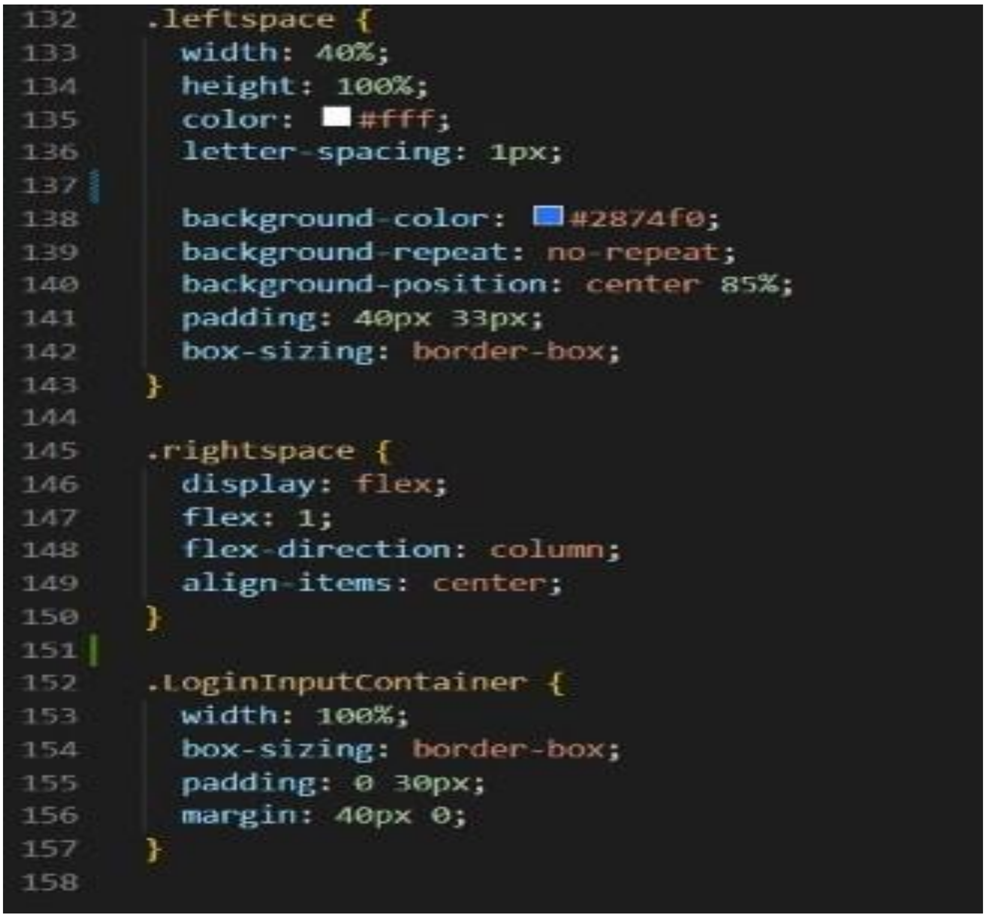
```
  const cart = useSelector((state) => state.cart);

  const userSignup = () => {
    const user = { firstName, lastName, email, password ,fullName};
    if (
      firstName === "" ||
      lastName === "" ||
      email === "" ||
      password === ""
    ) {
      return;
    }

    dispatch(_signup(user));
  };

  const userLogin = () => {
    if (signup) {
      userSignup();
    } else {
      dispatch(login({ email, password }));
    }
  };

  const logout = () => {
    dispatch(signout());
  };

  useEffect(() => {
    if (auth.authenticate) {
      setLoginModal(false);
    }
  }, [auth.authenticate]);

  // useEffect(() => {
  //   dispatch(getCartItems());
  // }, []);t
```



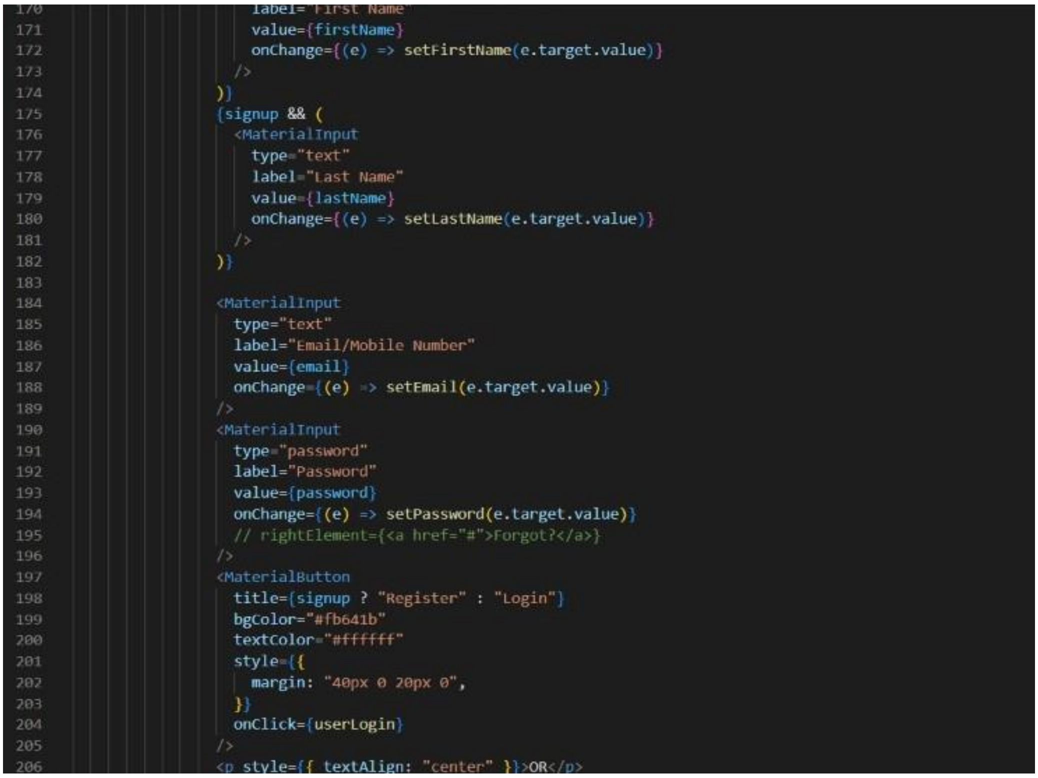
```
  const smart=()=>{
    return(
      <h1>Smart Fashion Recommander</h1>
    )
  }

  const renderLoggedInMenu = () => {
    return (
      <DropdownMenu
        menu={<a className="fullName">{auth.user.fullName}</a>}
        menus={[
          { label: "My Profile", href: "", icon: null },
          { label: "SuperCoin Zone", href: "", icon: null },
          { label: " Plus Zone", href: "", icon: null },
          {
            label: "Orders",
            href: `/account/orders`,
            icon: null,
          },
          { label: "Wishlist", href: "", icon: null },
          { label: "My Chats", href: "", icon: null },
          { label: "Coupons", href: "", icon: null },
          { label: "Rewards", href: "", icon: null },
          { label: "Notifications", href: "", icon: null },
          { label: "Gift Cards", href: "", icon: null },
          { label: "Logout", href: "", icon: null, onClick: logout },
        ]}
      />
    );
  };

  const renderNonLoggedInMenu = () => {
    return (
      <DropdownMenu
        menu={
          <a
            className="loginButton"
```



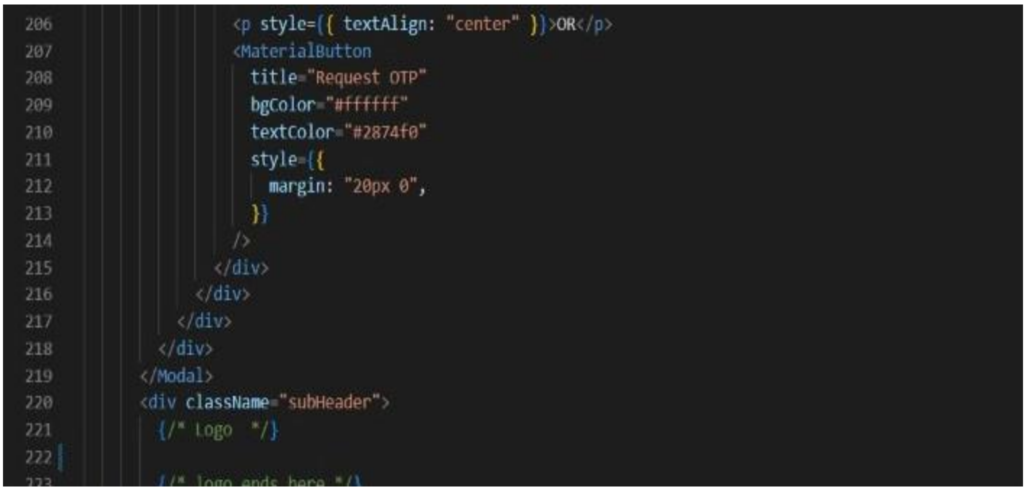
```
const renderNonLoggedInMenu = () => {
  return (
    <DropdownMenu
      menu={
        <a
          className="loginButton"
          onClick={() => {
            setSignup(false);
            setLoginModal(true);
          }}
        >

          Login
        </a>
      }
      menus={[
        { label: "My Profile", href: "", icon: null },

        {
          label: "Orders",
          href: `/account/orders`,
          icon: null,
          onClick: () => {
            !auth.authenticate && setLoginModal(true);
          },
        },
        { label: "Wishlist", href: "", icon: null },
        { label: "Rewards", href: "", icon: null },
        { label: "Gift Cards", href: "", icon: null },
      ]}
      firstMenu={
        <div className="firstmenu">
          <span>New Customer?</span>
          <a
            onClick={() => {
              setLoginModal(true);
```



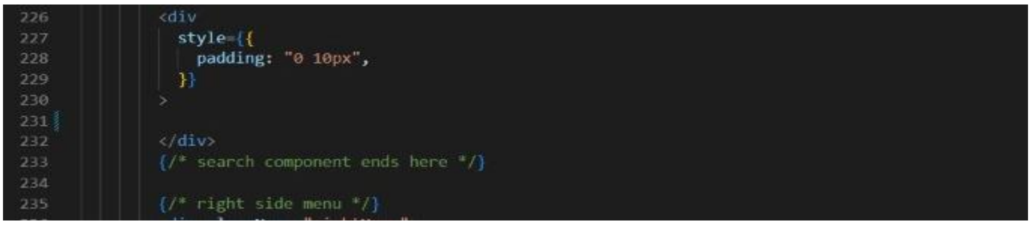
```
.leftspace {
  width: 40%;
  height: 100%;
  color: #fff;
  letter-spacing: 1px;

  background-color: #2874f0;
  background-repeat: no-repeat;
  background-position: center 85%;
  padding: 40px 33px;
  box-sizing: border-box;
}

.rightspace {
  display: flex;
  flex: 1;
  flex-direction: column;
  align-items: center;
}

.LoginInputContainer {
  width: 100%;
  box-sizing: border-box;
  padding: 0 30px;
  margin: 40px 0;
}
```



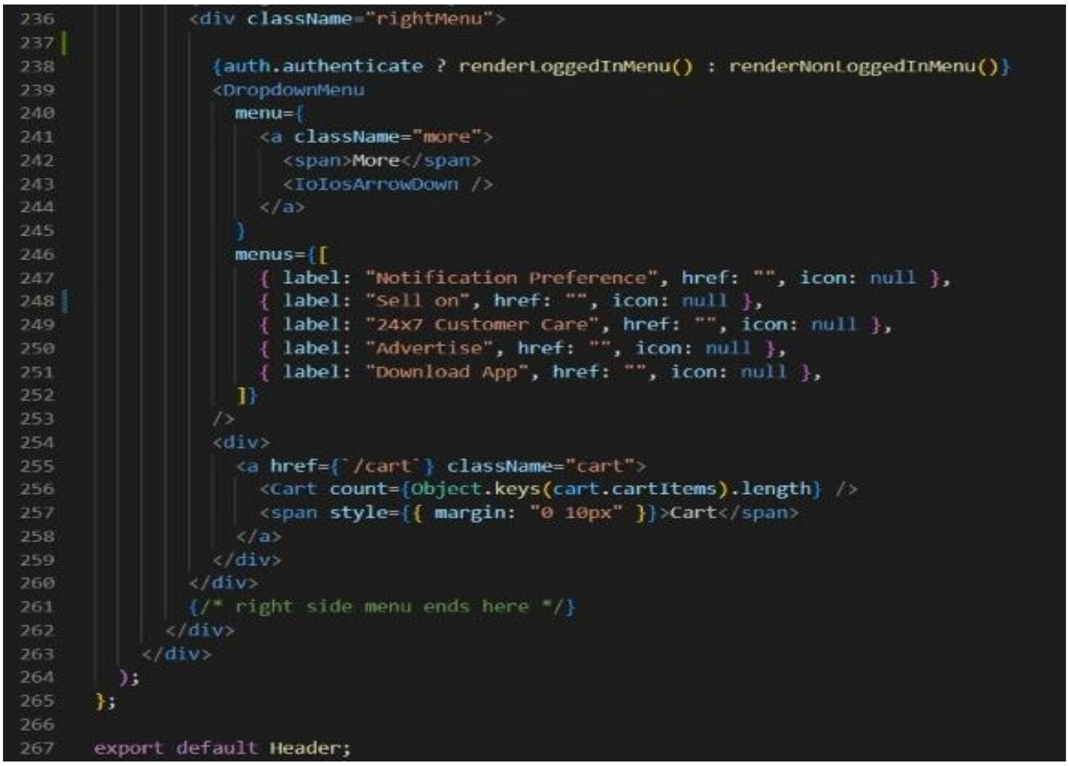
```
    label="First Name"
    value={firstName}
    onChange={(e) => setFirstName(e.target.value)}
  />
)}
{signup && (
  <MaterialInput
    type="text"
    label="Last Name"
    value={lastName}
    onChange={(e) => setLastName(e.target.value)}
  />
)}

<MaterialInput
  type="text"
  label="Email/Mobile Number"
  value={email}
  onChange={(e) => setEmail(e.target.value)}
/>
<MaterialInput
  type="password"
  label="Password"
  value={password}
  onChange={(e) => setPassword(e.target.value)}
  // rightElement={<a href="#">Forgot?</a>}
/>
<MaterialButton
  title={signup ? "Register" : "Login"}
  bgColor="#fb641b"
  textColor="#ffffff"
  style={{
    margin: "40px 0 20px 0",
  }}
  onClick={userLogin}
/>
<p style={{ textAlign: "center" }}>OR</p>
```



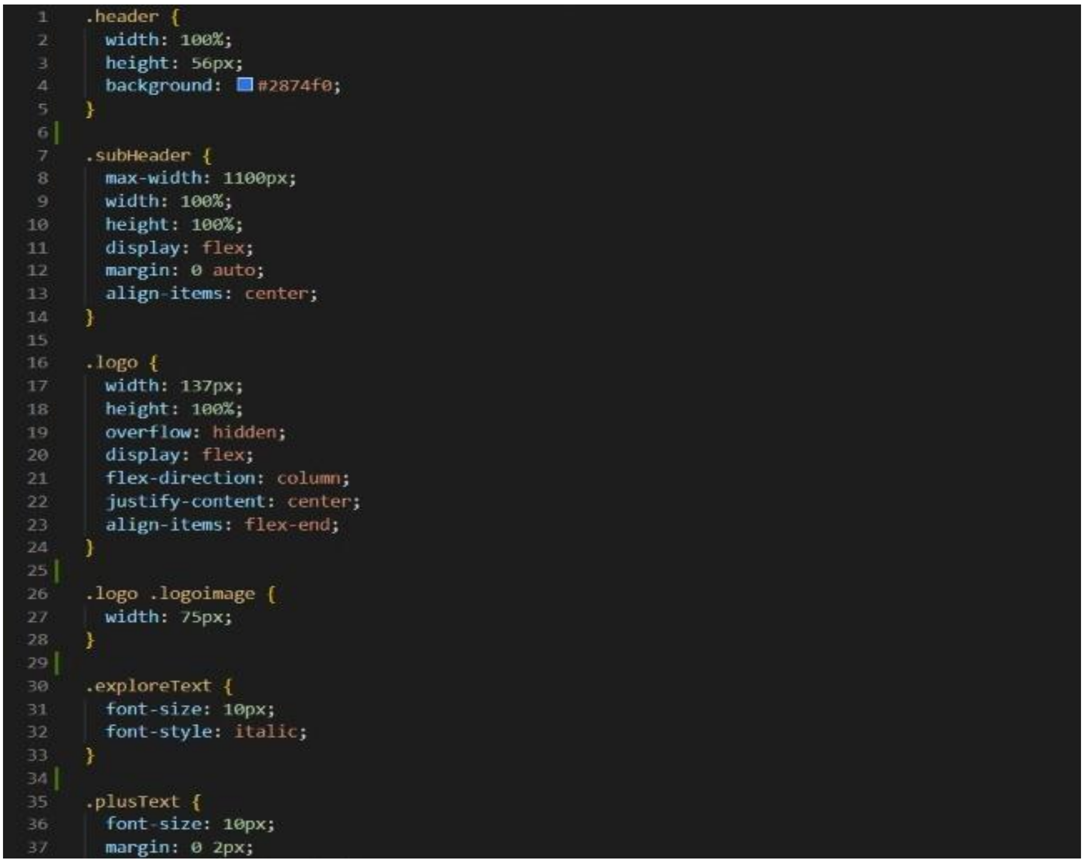
```
          <p style={{ textAlign: "center" }}>OR</p>
          <MaterialButton
            title="Request OTP"
            bgColor="#ffffff"
            textColor="#2874f0"
            style={{
              margin: "20px 0",
            }}
          />
        </div>
      </div>
    </div>
  </div>
</Modal>
<div className="subHeader">
  {/* Logo */}

  {/* logo ends here */}
```



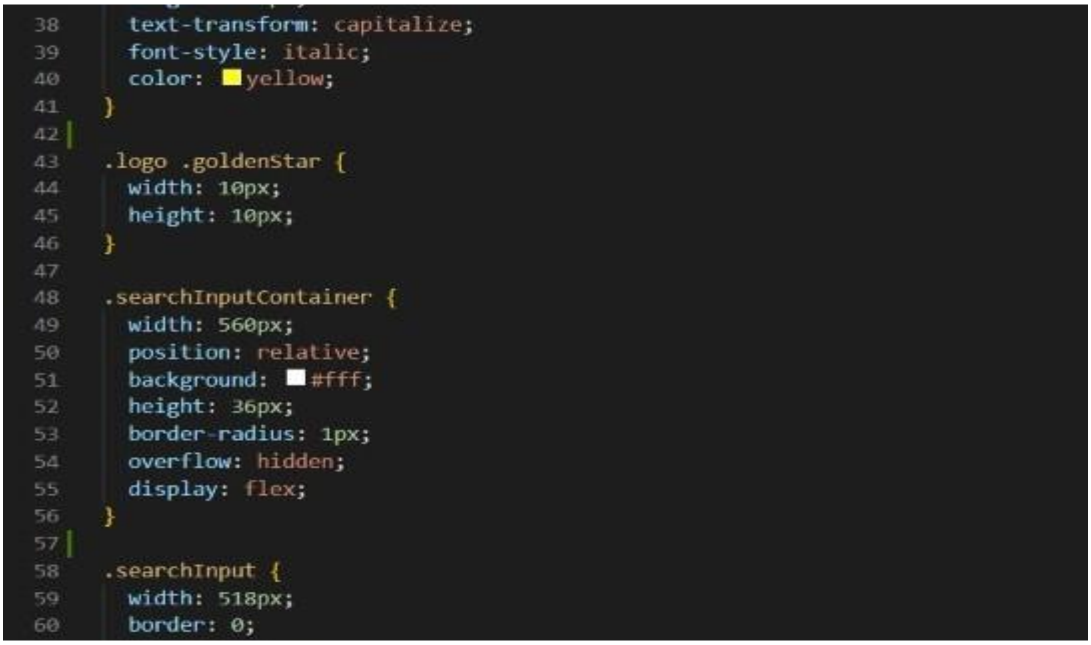
```
<div
  style={{
    padding: "0 10px",
  }}
>

</div>
{/* search component ends here */}

{/* right side menu */}
```



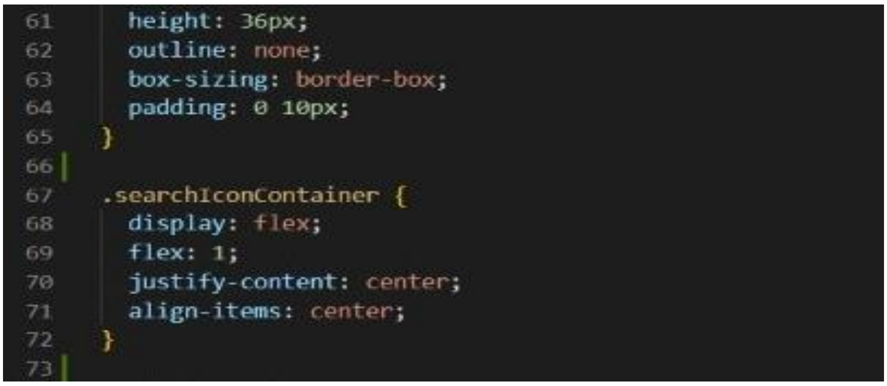
```jsx
        <div className="rightMenu">

          {auth.authenticate ? renderLoggedInMenu() : renderNonLoggedInMenu()}
          <DropdownMenu
            menu={
              <a className="more">
                <span>More</span>
                <IoIosArrowDown />
              </a>
            }
            menus={[
              { label: "Notification Preference", href: "", icon: null },
              { label: "Sell on", href: "", icon: null },
              { label: "24x7 Customer Care", href: "", icon: null },
              { label: "Advertise", href: "", icon: null },
              { label: "Download App", href: "", icon: null },
            ]}
          />
          <div>
            <a href={`/cart`} className="cart">
              <Cart count={Object.keys(cart.cartItems).length} />
              <span style={{ margin: "0 10px" }}>Cart</span>
            </a>
          </div>
        </div>
        {/* right side menu ends here */}
      </div>
    </div>
  );
};

export default Header;
```



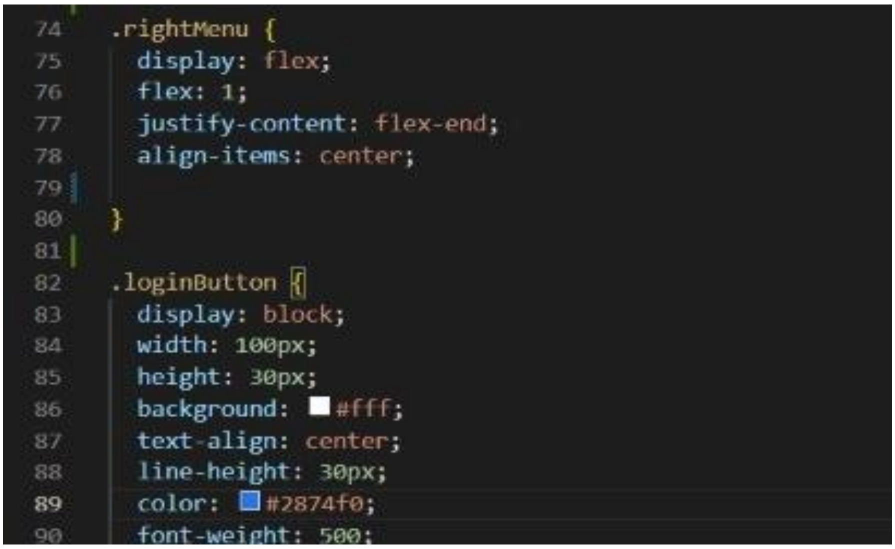
```css
.header {
  width: 100%;
  height: 56px;
  background: #2874f0;
}

.subHeader {
  max-width: 1100px;
  width: 100%;
  height: 100%;
  display: flex;
  margin: 0 auto;
  align-items: center;
}

.logo {
  width: 137px;
  height: 100%;
  overflow: hidden;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: flex-end;
}

.logo .logoimage {
  width: 75px;
}

.exploreText {
  font-size: 10px;
  font-style: italic;
}

.plusText {
  font-size: 10px;
  margin: 0 2px;
```



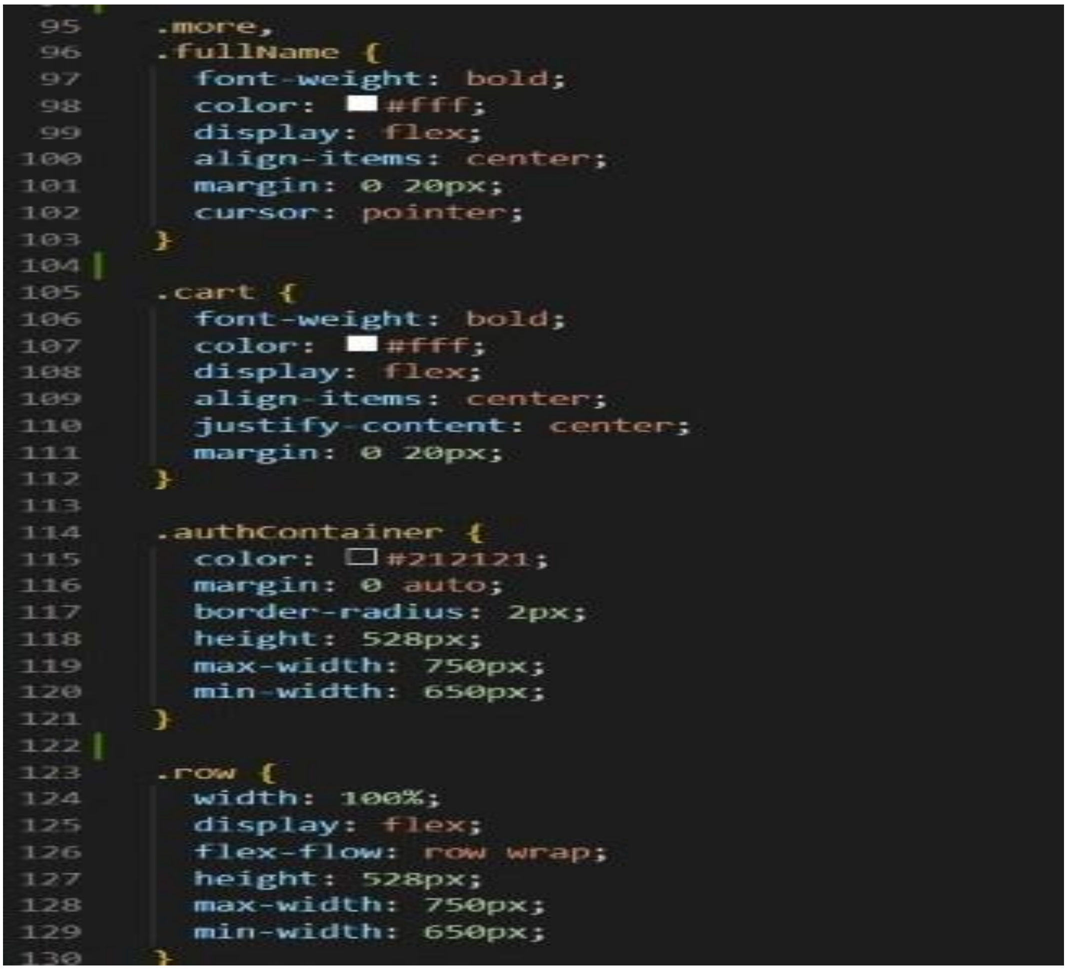
```css
  text-transform: capitalize;
  font-style: italic;
  color: yellow;
}

.logo .goldenStar {
  width: 10px;
  height: 10px;
}

.searchInputContainer {
  width: 560px;
  position: relative;
  background: #fff;
  height: 36px;
  border-radius: 1px;
  overflow: hidden;
  display: flex;
}

.searchInput {
  width: 518px;
  border: 0;
```



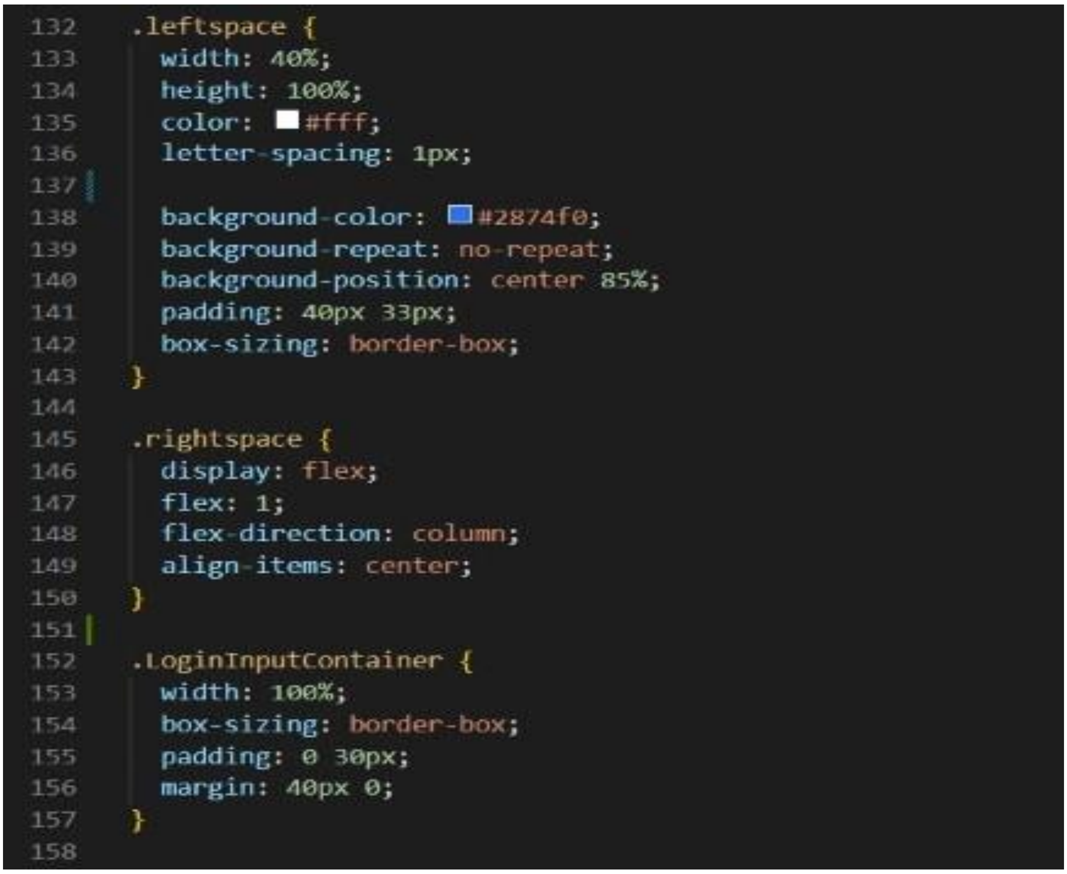
```css
  height: 36px;
  outline: none;
  box-sizing: border-box;
  padding: 0 10px;
}

.searchIconContainer {
  display: flex;
  flex: 1;
  justify-content: center;
  align-items: center;
}

```



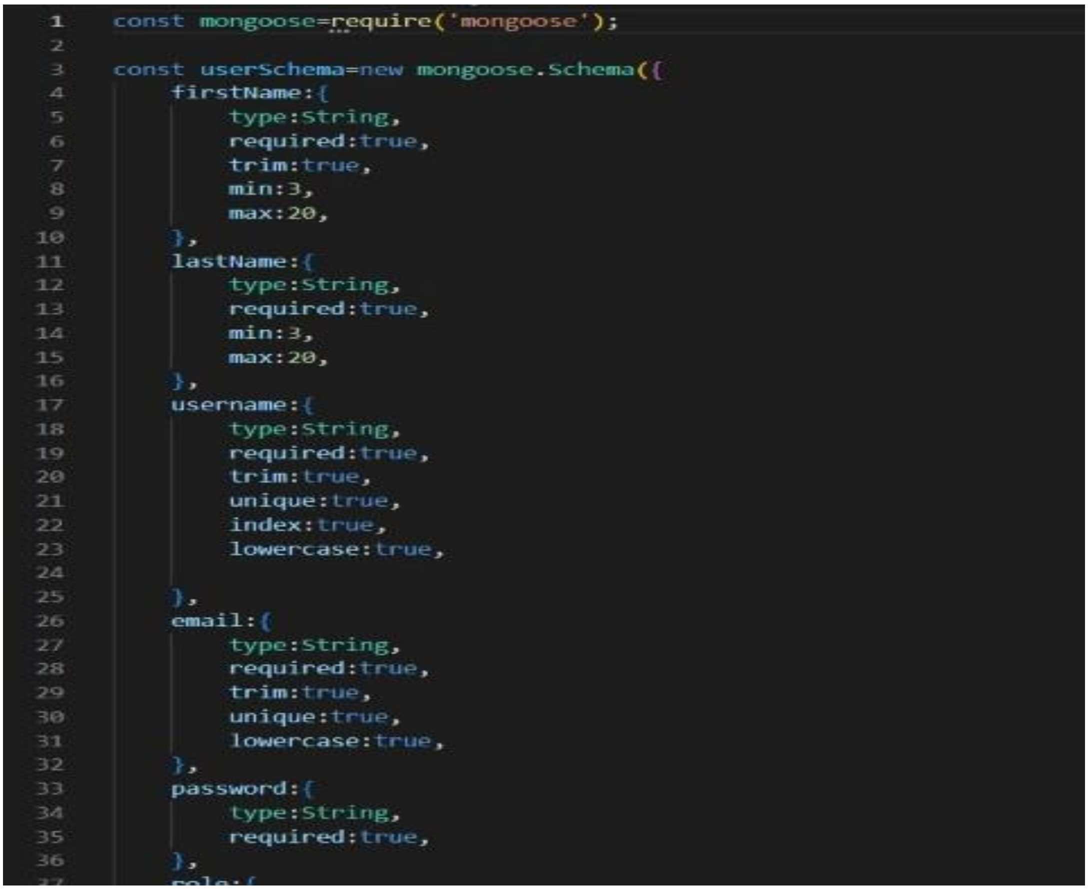
```css
.rightMenu {
  display: flex;
  flex: 1;
  justify-content: flex-end;
  align-items: center;

}

.loginButton {
  display: block;
  width: 100px;
  height: 30px;
  background: #fff;
  text-align: center;
  line-height: 30px;
  color: #2874f0;
  font-weight: 500;
```



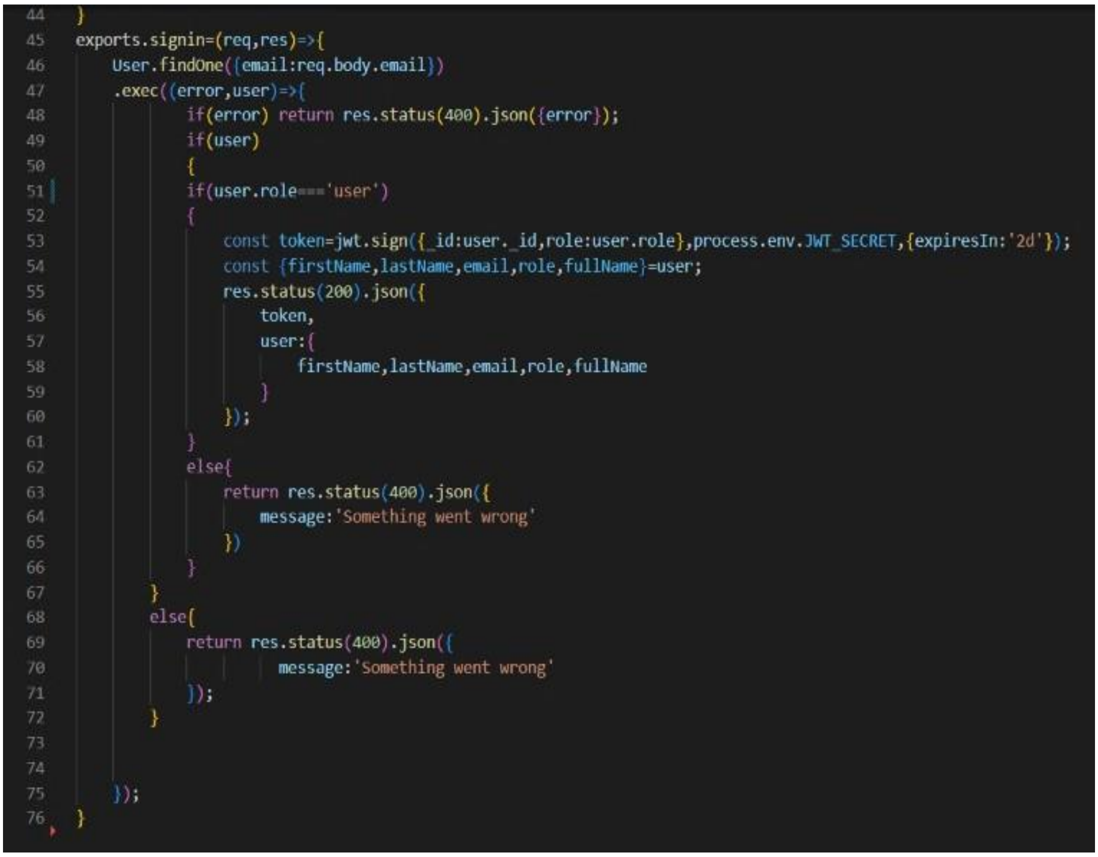
```css
  .more,
  .fullName {
    font-weight: bold;
    color: #fff;
    display: flex;
    align-items: center;
    margin: 0 20px;
    cursor: pointer;
  }

  .cart {
    font-weight: bold;
    color: #fff;
    display: flex;
    align-items: center;
    justify-content: center;
    margin: 0 20px;
  }

  .authContainer {
    color: #212121;
    margin: 0 auto;
    border-radius: 2px;
    height: 528px;
    max-width: 750px;
    min-width: 650px;
  }

  .row {
    width: 100%;
    display: flex;
    flex-flow: row wrap;
    height: 528px;
    max-width: 750px;
    min-width: 650px;
  }
```



```css
.leftspace {
  width: 40%;
  height: 100%;
  color: #fff;
  letter-spacing: 1px;

  background-color: #2874f0;
  background-repeat: no-repeat;
  background-position: center 85%;
  padding: 40px 33px;
  box-sizing: border-box;
}

.rightspace {
  display: flex;
  flex: 1;
  flex-direction: column;
  align-items: center;
}

.LoginInputContainer {
  width: 100%;
  box-sizing: border-box;
  padding: 0 30px;
  margin: 40px 0;
}
```

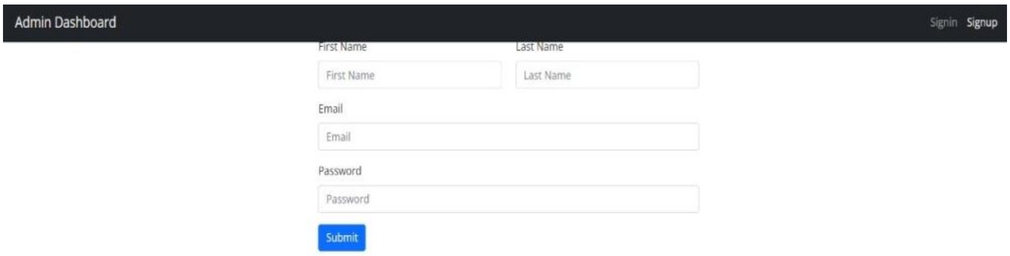


```
const mongoose=require('mongoose');

const userSchema=new mongoose.Schema({
    firstName:{
        type:String,
        required:true,
        trim:true,
        min:3,
        max:20,
    },
    lastName:{
        type:String,
        required:true,
        min:3,
        max:20,
    },
    username:{
        type:String,
        required:true,
        trim:true,
        unique:true,
        index:true,
        lowercase:true,

    },
    email:{
        type:String,
        required:true,
        trim:true,
        unique:true,
        lowercase:true,
    },
    password:{
        type:String,
        required:true,
    },
    role:{
```

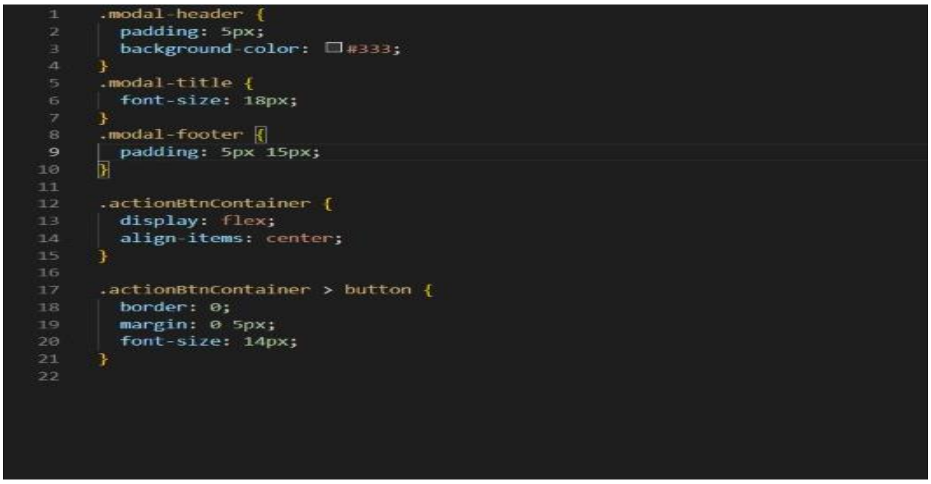


```
}
exports.signin=(req,res)=>{
    User.findOne({email:req.body.email})
    .exec((error,user)=>{
            if(error) return res.status(400).json({error});
            if(user)
            {
            if(user.role==='user')
            {
                const token=jwt.sign({_id:user._id,role:user.role},process.env.JWT_SECRET,{expiresIn:'2d'});
                const {firstName,lastName,email,role,fullName}=user;
                res.status(200).json({
                    token,
                    user:{
                        firstName,lastName,email,role,fullName
                    }
                });
            }
            else{
                return res.status(400).json({
                    message:'Something went wrong'
                })
            }
        }
        else{
            return res.status(400).json({
                      message:'Something went wrong'
            });
        }


    });
}
```

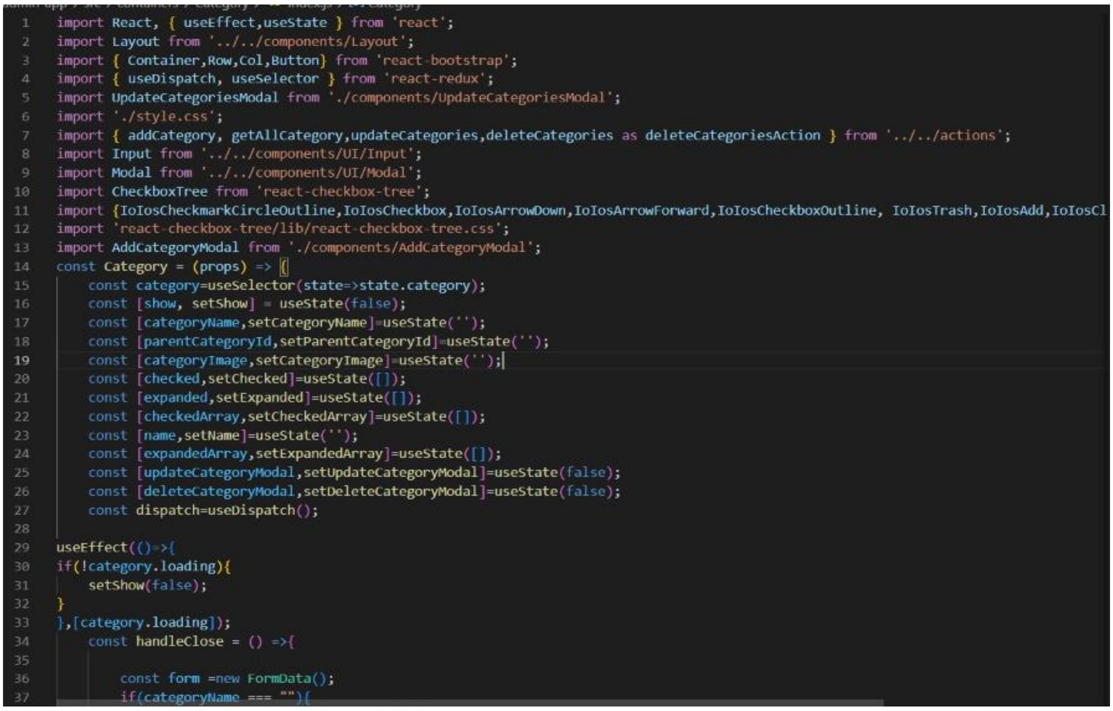
SIGNUP:



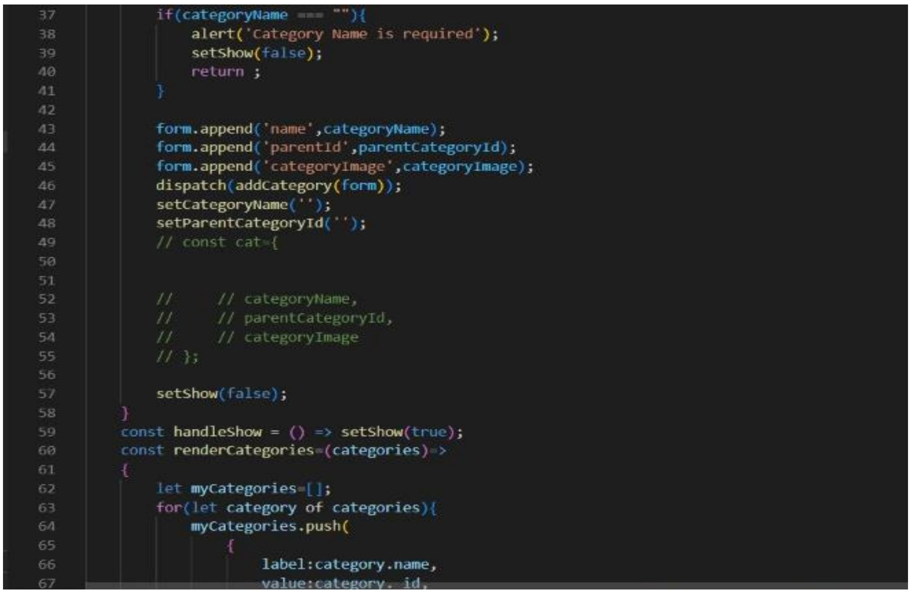
CODE:



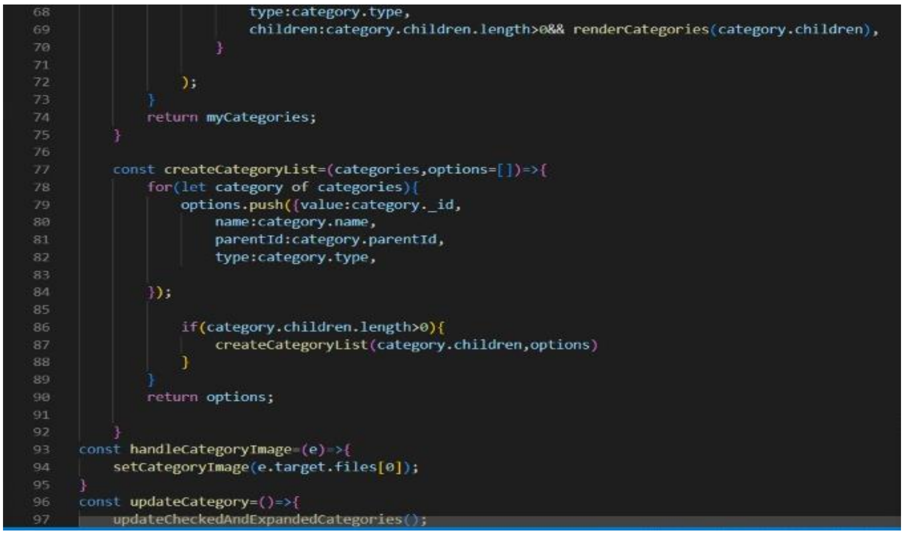
```css
.modal-header {
  padding: 5px;
  background-color: #333;
}
.modal-title {
  font-size: 18px;
}
.modal-footer {
  padding: 5px 15px;
}

.actionBtnContainer {
  display: flex;
  align-items: center;
}

.actionBtnContainer > button {
  border: 0;
  margin: 0 5px;
  font-size: 14px;
}
```



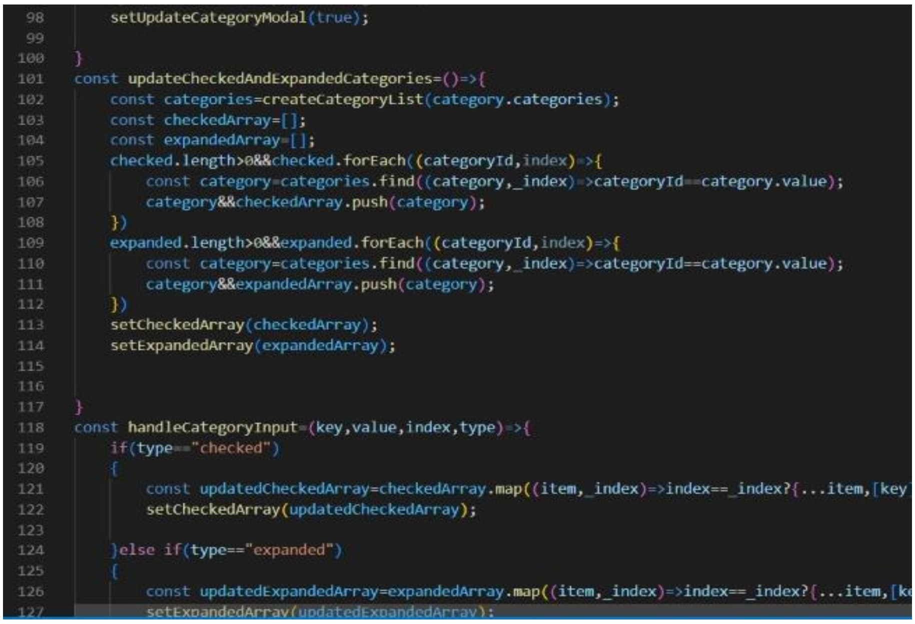
```javascript
import React, { useEffect,useState } from 'react';
import Layout from '../../components/Layout';
import { Container,Row,Col,Button} from 'react-bootstrap';
import { useDispatch, useSelector } from 'react-redux';
import UpdateCategoriesModal from './components/UpdateCategoriesModal';
import './style.css';
import { addCategory, getAllCategory,updateCategories,deleteCategories as deleteCategoriesAction } from '../../actions';
import Input from '../../components/UI/Input';
import Modal from '../../components/UI/Modal';
import CheckboxTree from 'react-checkbox-tree';
import {IoIosCheckmarkCircleOutline,IoIosCheckbox,IoIosArrowDown,IoIosArrowForward,IoIosCheckboxOutline, IoIosTrash,IoIosAdd,IoIosCl
import 'react-checkbox-tree/lib/react-checkbox-tree.css';
import AddCategoryModal from './components/AddCategoryModal';
const Category = (props) => {
    const category=useSelector(state=>state.category);
    const [show, setShow] = useState(false);
    const [categoryName,setCategoryName]=useState('');
    const [parentCategoryId,setParentCategoryId]=useState('');
    const [categoryImage,setCategoryImage]=useState('');
    const [checked,setChecked]=useState([]);
    const [expanded,setExpanded]=useState([]);
    const [checkedArray,setCheckedArray]=useState([]);
    const [name,setName]=useState('');
    const [expandedArray,setExpandedArray]=useState([]);
    const [updateCategoryModal,setUpdateCategoryModal]=useState(false);
    const [deleteCategoryModal,setDeleteCategoryModal]=useState(false);
    const dispatch=useDispatch();

useEffect(()=>{
if(!category.loading){
    setShow(false);
}
},[category.loading]);
    const handleClose = () =>{

        const form =new FormData();
        if(categoryName === ""){
```



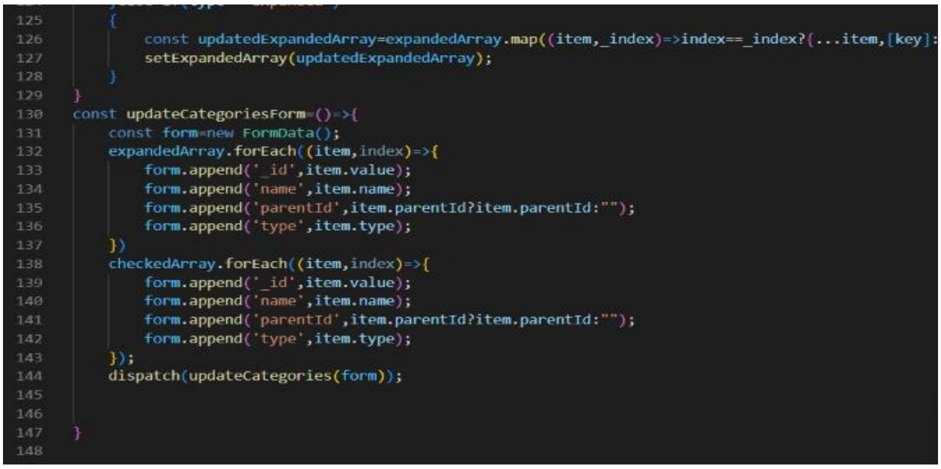
```
        if(categoryName === ""){
            alert('Category Name is required');
            setShow(false);
            return ;
        }

        form.append('name',categoryName);
        form.append('parentId',parentCategoryId);
        form.append('categoryImage',categoryImage);
        dispatch(addCategory(form));
        setCategoryName('');
        setParentCategoryId('');
        // const cat={


        //     // categoryName,
        //     // parentCategoryId,
        //     // categoryImage
        // };

        setShow(false);
    }
    const handleShow = () => setShow(true);
    const renderCategories=(categories)=>
    {
        let myCategories=[];
        for(let category of categories){
            myCategories.push(
                {
                    label:category.name,
                    value:category._id,
```



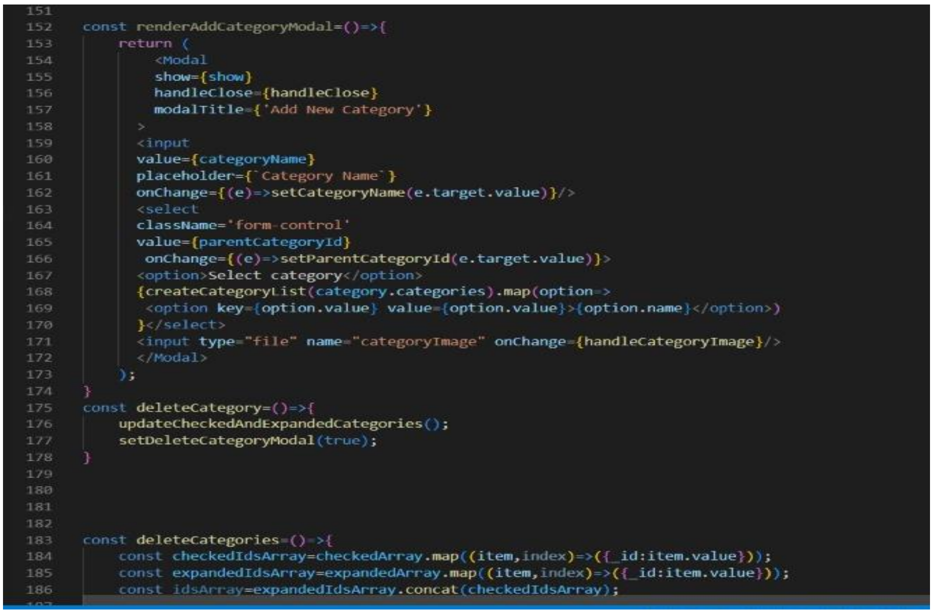
```
                    type:category.type,
                    children:category.children.length>0&& renderCategories(category.children),
                }

            );
        }
        return myCategories;
    }

    const createCategoryList=(categories,options=[])=>{
        for(let category of categories){
            options.push({value:category._id,
                name:category.name,
                parentId:category.parentId,
                type:category.type,

        });

            if(category.children.length>0){
                createCategoryList(category.children,options)
            }
        }
        return options;

    }
const handleCategoryImage=(e)=>{
    setCategoryImage(e.target.files[0]);
}
const updateCategory=()=>{
    updateCheckedAndExpandedCategories();
```



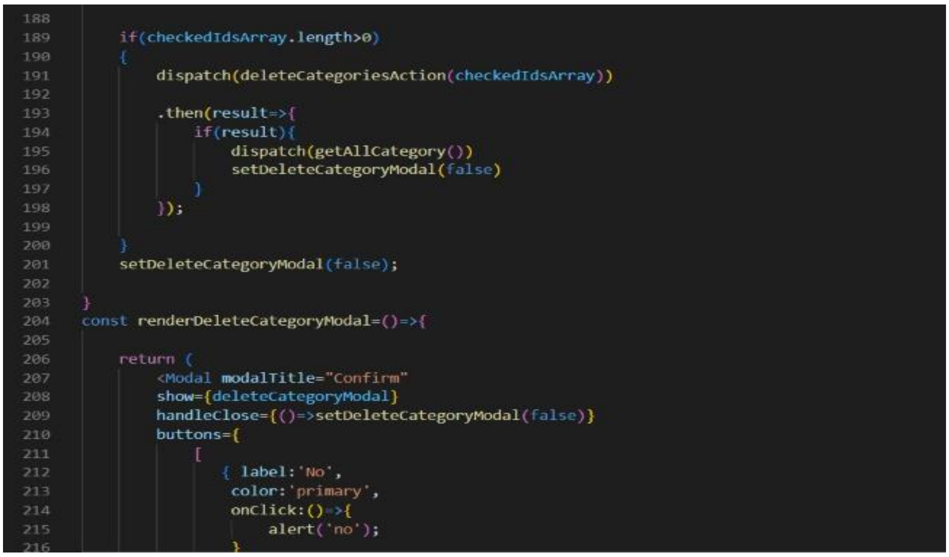
```
    setUpdateCategoryModal(true);

}
const updateCheckedAndExpandedCategories=()=>{
    const categories=createCategoryList(category.categories);
    const checkedArray=[];
    const expandedArray=[];
    checked.length>0&&checked.forEach((categoryId,index)=>{
        const category=categories.find((category,_index)=>categoryId==category.value);
        category&&checkedArray.push(category);
    })
    expanded.length>0&&expanded.forEach((categoryId,index)=>{
        const category=categories.find((category,_index)=>categoryId==category.value);
        category&&expandedArray.push(category);
    })
    setCheckedArray(checkedArray);
    setExpandedArray(expandedArray);


}
const handleCategoryInput=(key,value,index,type)=>{
    if(type=="checked")
    {
        const updatedCheckedArray=checkedArray.map((item,_index)=>index==_index?{...item,[key
        setCheckedArray(updatedCheckedArray);

    }else if(type=="expanded")
    {
        const updatedExpandedArray=expandedArray.map((item,_index)=>index==_index?{...item,[k
        setExpandedArray(updatedExpandedArray);
```



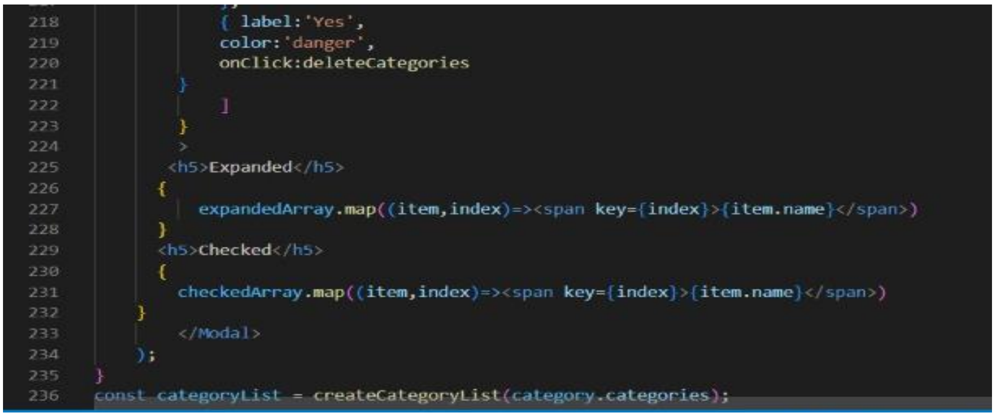
```
        {
            const updatedExpandedArray=expandedArray.map((item,_index)=>index==_index?{...item,[key]:
            setExpandedArray(updatedExpandedArray);
        }
}
const updateCategoriesForm=()=>{
        const form=new FormData();
        expandedArray.forEach((item,index)=>{
            form.append('_id',item.value);
            form.append('name',item.name);
            form.append('parentId',item.parentId?item.parentId:"");
            form.append('type',item.type);
        })
        checkedArray.forEach((item,index)=>{
            form.append('_id',item.value);
            form.append('name',item.name);
            form.append('parentId',item.parentId?item.parentId:"");
            form.append('type',item.type);
        });
        dispatch(updateCategories(form));


}

```



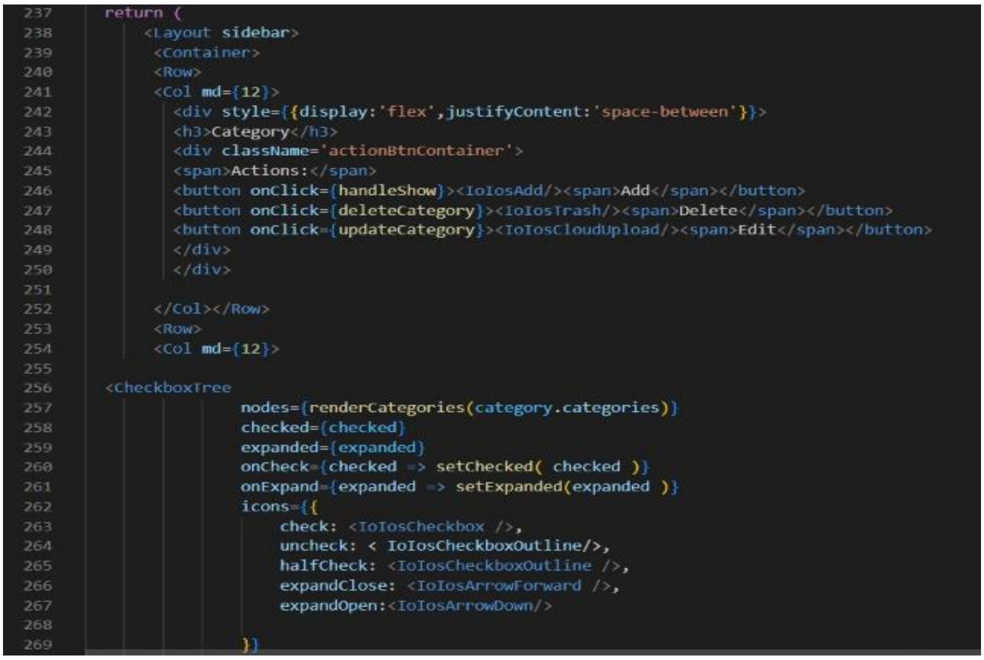
```

const renderAddCategoryModal=()=>{
        return (
            <Modal
            show={show}
            handleClose={handleClose}
            modalTitle={'Add New Category'}
          >
          <input
          value={categoryName}
          placeholder={`Category Name`}
          onChange={(e)=>setCategoryName(e.target.value)}/>
          <select
          className='form-control'
          value={parentCategoryId}
           onChange={(e)=>setParentCategoryId(e.target.value)}>
          <option>Select category</option>
          {createCategoryList(category.categories).map(option=>
           <option key={option.value} value={option.value}>{option.name}</option>)
          }</select>
          <input type="file" name="categoryImage" onChange={handleCategoryImage}/>
          </Modal>
        );
}
const deleteCategory=()=>{
        updateCheckedAndExpandedCategories();
        setDeleteCategoryModal(true);
}




const deleteCategories=()=>{
        const checkedIdsArray=checkedArray.map((item,index)=>({_id:item.value}));
        const expandedIdsArray=expandedArray.map((item,index)=>({_id:item.value}));
        const idsArray=expandedIdsArray.concat(checkedIdsArray);
```



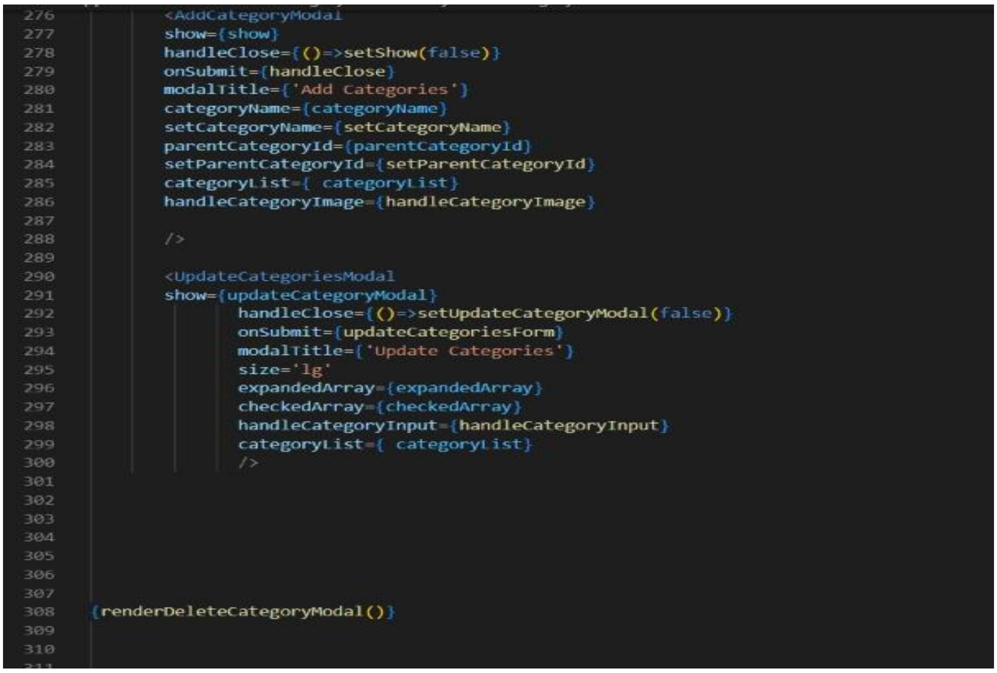
```

    if(checkedIdsArray.length>0)
    {
        dispatch(deleteCategoriesAction(checkedIdsArray))

        .then(result=>{
            if(result){
                dispatch(getAllCategory())
                setDeleteCategoryModal(false)
            }
        });

    }
    setDeleteCategoryModal(false);

}
const renderDeleteCategoryModal=()=>{

    return (
        <Modal modalTitle="Confirm"
        show={deleteCategoryModal}
        handleClose={()=>setDeleteCategoryModal(false)}
        buttons={
            [
               { label:'No',
                color:'primary',
                onClick:()=>{
                    alert('no');
                }
```



```
                { label:'Yes',
                color:'danger',
                onClick:deleteCategories
            }
                ]
            }
            >
          <h5>Expanded</h5>
        {
          expandedArray.map((item,index)=><span key={index}>{item.name}</span>)
        }
        <h5>Checked</h5>
        {
          checkedArray.map((item,index)=><span key={index}>{item.name}</span>)
      }
          </Modal>
      );
  }
  const categoryList = createCategoryList(category.categories);
```

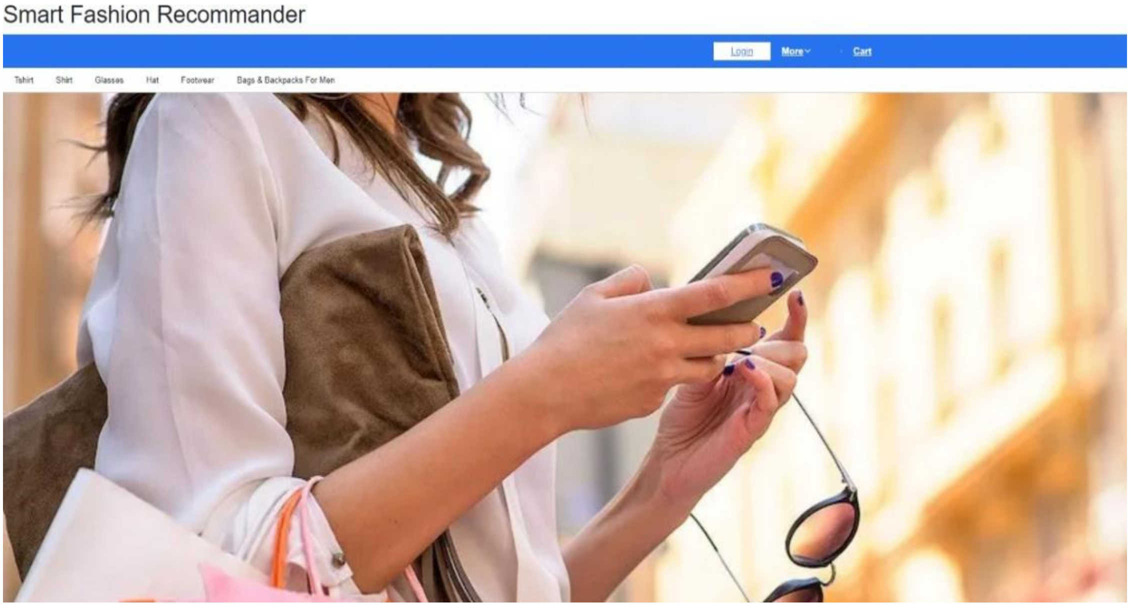


```
   return (
       <Layout sidebar>
        <Container>
        <Row>
        <Col md={12}>
          <div style={{display:'flex',justifyContent:'space-between'}}>
          <h3>Category</h3>
          <div className='actionBtnContainer'>
          <span>Actions:</span>
          <button onClick={handleShow}><IoIosAdd/><span>Add</span></button>
          <button onClick={deleteCategory}><IoIosTrash/><span>Delete</span></button>
          <button onClick={updateCategory}><IoIosCloudUpload/><span>Edit</span></button>
          </div>
          </div>

        </Col></Row>
        <Row>
        <Col md={12}>

   <CheckboxTree
                 nodes={renderCategories(category.categories)}
                 checked={checked}
                 expanded={expanded}
                 onCheck={checked => setChecked( checked )}
                 onExpand={expanded => setExpanded(expanded )}
                 icons={{
                     check: <IoIosCheckbox />,
                     uncheck: < IoIosCheckboxOutline/>,
                     halfCheck: <IoIosCheckboxOutline />,
                     expandClose: <IoIosArrowForward />,
                     expandOpen:<IoIosArrowDown/>

                 }}
```

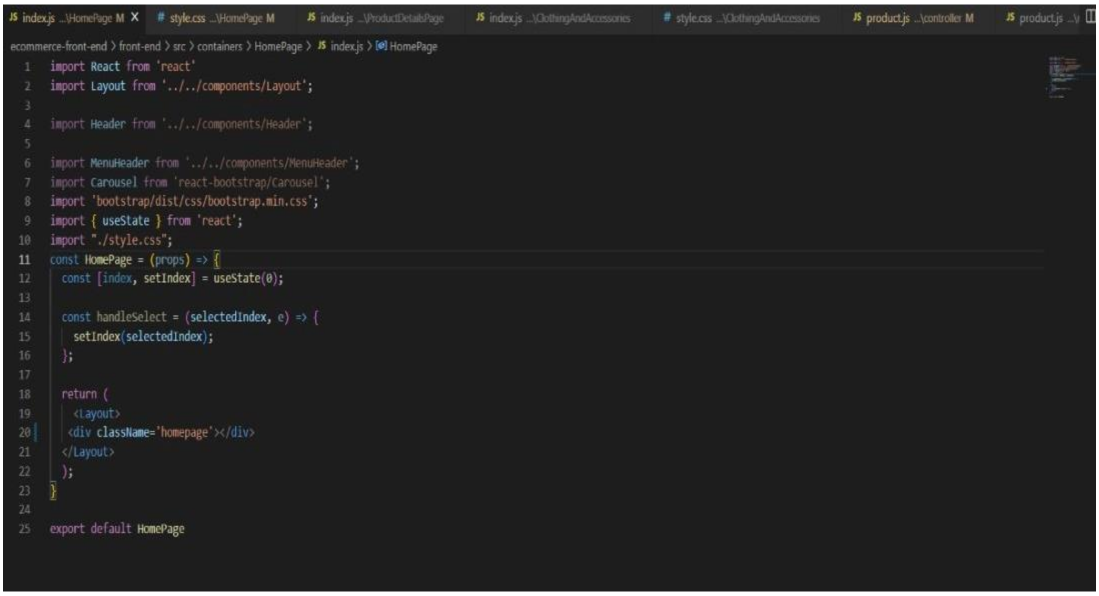


```
        <AddCategoryModal
        show={show}
        handleClose={()=>setShow(false)}
        onSubmit={handleClose}
        modalTitle={'Add Categories'}
        categoryName={categoryName}
        setCategoryName={setCategoryName}
        parentCategoryId={parentCategoryId}
        setParentCategoryId={setParentCategoryId}
        categoryList={ categoryList}
        handleCategoryImage={handleCategoryImage}

        />

        <UpdateCategoriesModal
        show={updateCategoryModal}
               handleClose={()=>setUpdateCategoryModal(false)}
               onSubmit={updateCategoriesForm}
               modalTitle={'Update Categories'}
               size='lg'
               expandedArray={expandedArray}
               checkedArray={checkedArray}
               handleCategoryInput={handleCategoryInput}
               categoryList={ categoryList}
               />







{renderDeleteCategoryModal()}
```

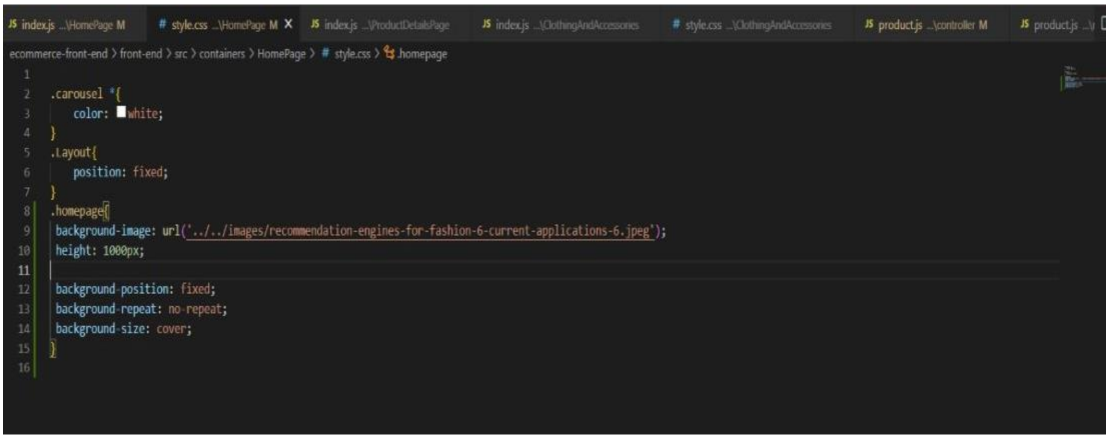
HOME PAGE:



CODE:


index.js ...\HomePage M
style.css ...\HomePage M
index.js ...\ProductDetailsPage
index.js ...\ClothingAndAccessories
style.css ...\ClothingAndAccessories
product.js ...\controller M
product.js ...\
ecommerce-front-end > front-end > src > containers > HomePage > index.js > HomePage

```
import React from 'react'
import Layout from '../../components/Layout';

import Header from '../../components/Header';

import MenuHeader from '../../components/MenuHeader';
import Carousel from 'react-bootstrap/Carousel';
import 'bootstrap/dist/css/bootstrap.min.css';
import { useState } from 'react';
import "./style.css";
const HomePage = (props) => {
  const [index, setIndex] = useState(0);

  const handleSelect = (selectedIndex, e) => {
    setIndex(selectedIndex);
  };

  return (
    <Layout>
   <div className='homepage'></div>
  </Layout>
  );
}

export default HomePage
```


index.js ...\HomePage M
style.css ...\HomePage M
index.js ...\ProductDetailsPage
index.js ...\ClothingAndAccessories
style.css ...\ClothingAndAccessories
product.js ...\controller M
product.js ...\
ecommerce-front-end > front-end > src > containers > HomePage > style.css > .homepage

```

.carousel *{
    color: white;
}
.Layout{
    position: fixed;
}
.homepage{
 background-image: url('../../images/recommendation-engines-for-fashion-6-current-applications-6.jpeg');
 height: 1000px;

 background-position: fixed;
 background-repeat: no-repeat;
 background-size: cover;
}

```