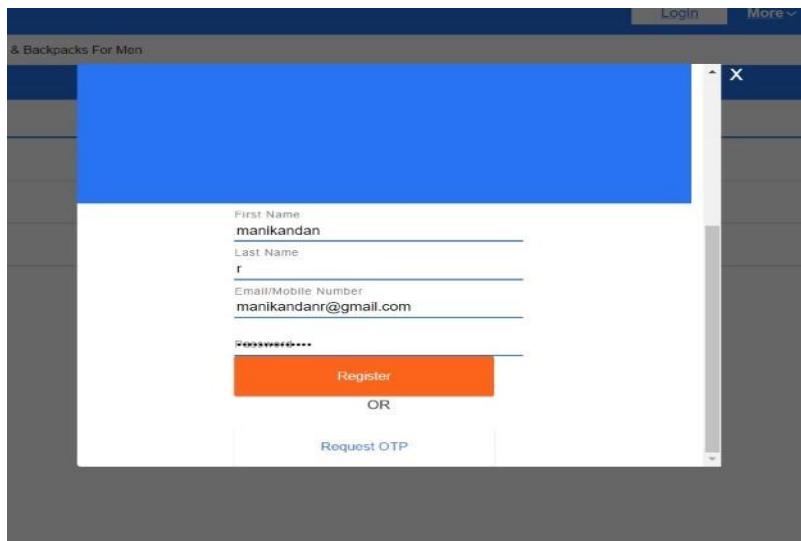


# PROJECT DEVELOPMENT PHASE

## SPRINT – 1

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

LOGIN :



CODE:

```
1 import React, { useEffect, useState } from "react";
2 import "./style.css";
3
4 import goldenStar from "../../images/logo/golden-star.png";
5 import { IoIosArrowDown, IoIosCart, IoIosSearch } from "react-icons/io";
6 import {
7   Modal,
8   MaterialInput,
9   MaterialButton,
10   DropdownMenu,
11 } from "../MaterialUI";
12 import { useDispatch, useSelector } from "react-redux";
13 import { login, signout, signup as _signup } from "../../actions";
14 import Cart from "../UI/Card";
15
16 /**
17  * @author
18  * @function Header
19  */
20
21 const Header = (props) => {
22   const [loginModal, setLoginModal] = useState(false);
23   const [signup, setSignup] = useState(false);
24   const [firstName, setFirstName] = useState("");
25   const [lastName, setLastName] = useState("");
26   const [email, setEmail] = useState("");
27   const [password, setPassword] = useState("");
28   const [fullName, setFullName] = useState("");
29   const [error, setError] = useState("");
30   const auth = useSelector((state) => state.auth);
31   const dispatch = useDispatch();
32
33   // state cart value
34   const cart = useSelector((state) => state.cart);
35
36   const userSignup = () => {
37     const user = { firstName, lastName, email, password, fullName;
```

```

34 const cart = useSelector((state) => state.cart);
35
36 const userSignup = () => {
37   const user = { firstName, lastName, email, password, fullName };
38   if (
39     firstName === "" ||
40     lastName === "" ||
41     email === "" ||
42     password === ""
43   ) {
44     return;
45   }
46
47   dispatch(_signup(user));
48 };
49
50 const userLogin = () => {
51   if (signup) {
52     userSignup();
53   } else {
54     dispatch(login({ email, password }));
55   }
56 };
57
58 const logout = () => {
59   dispatch(signout());
60 };
61
62 useEffect(() => {
63   if (auth.authenticate) {
64     setLoginModal(false);
65   }
66 }, [auth.authenticate]);
67
68 // useEffect(() => {
69 //   dispatch(getCartItems());
70 // }, []);

```

```

71 const smart=()=>{
72   return(
73     <h1>Smart Fashion Recommender</h1>
74   )
75 }
76
77 const renderLoggedInMenu = () => {
78   return (
79     <DropDownMenu
80       menu={
81         <a className="fullName">{auth.user.fullName}</a>
82       }
83       menus={[
84         { label: "My Profile", href: "", icon: null },
85         { label: "SuperCoin Zone", href: "", icon: null },
86         { label: "Plus Zone", href: "", icon: null },
87         {
88           label: "Orders",
89           href: "/account/orders",
90           icon: null,
91         },
92         { label: "Wishlist", href: "", icon: null },
93         { label: "My Chats", href: "", icon: null },
94         { label: "Coupons", href: "", icon: null },
95         { label: "Rewards", href: "", icon: null },
96         { label: "Notifications", href: "", icon: null },
97         { label: "Gift Cards", href: "", icon: null },
98         { label: "Logout", href: "", icon: null, onClick: logout },
99       ]}
100     </DropDownMenu>
101   );
102 };
103
104 const renderNonLoggedInMenu = () => {
105   return (
106     <DropDownMenu
107       menu={
108         <a
109           className="loginButton"

```

```

101
102 const renderNonLoggedInMenu = () => {
103   return (
104     <DropDownMenu
105       menu={
106         <a
107           className="loginButton"
108           onClick={() => {
109             setSignup(false);
110             setLoginModal(true);
111           }}
112         >
113           Login
114         </a>
115       )
116     </div>
117     <div className="firstmenu">
118       <span>New Customer?</span>
119       <a
120         className="firstmenu"
121         href="#"
122         onClick={() => {
123           setSignup(false);
124           setLoginModal(true);
125         }}
126       >
127         New Customer?
128       </a>
129     </div>
130   );
131 }
132
133 const renderLoggedInMenu = () => {
134   return (
135     <DropDownMenu
136       menu={
137         <a
138           className="loginButton"
139           href="#"
140           onClick={() => {
141             setSignup(false);
142             setLoginModal(true);
143           }}
144         >
145           Login
146         </a>
147       )
148     </div>
149     <div className="firstmenu">
150       <span>New Customer?</span>
151       <a
152         className="firstmenu"
153         href="#"
154         onClick={() => {
155           setSignup(false);
156           setLoginModal(true);
157         }}
158       >
159         New Customer?
160       </a>
161     </div>
162   );
163 }

```

```

132 .leftspace {
133   width: 40%;
134   height: 100%;
135   color: #fff;
136   letter-spacing: 1px;
137
138   background-color: #2874f0;
139   background-repeat: no-repeat;
140   background-position: center 85%;
141   padding: 40px 33px;
142   box-sizing: border-box;
143 }
144
145 .rightspace {
146   display: flex;
147   flex: 1;
148   flex-direction: column;
149   align-items: center;
150 }
151
152 .LoginInputContainer {
153   width: 100%;
154   box-sizing: border-box;
155   padding: 0 30px;
156   margin: 40px 0;
157 }
158

```

```

170         label="First Name"
171         value={firstName}
172         onChange={(e) => setFirstName(e.target.value)}
173     />
174 )}
175 {signup && (
176     <MaterialInput
177         type="text"
178         label="Last Name"
179         value={lastName}
180         onChange={(e) => setLastName(e.target.value)}
181     />
182 )}
183
184 <MaterialInput
185     type="text"
186     label="Email/Mobile Number"
187     value={email}
188     onChange={(e) => setEmail(e.target.value)}
189 />
190 <MaterialInput
191     type="password"
192     label="Password"
193     value={password}
194     onChange={(e) => setPassword(e.target.value)}
195     // rightElement={<a href="#">Forgot?</a>}
196 />
197 <MaterialButton
198     title={signup ? "Register" : "Login"}
199     bgColor="#fb641b"
200     textColor="ffffff"
201     style={{
202         margin: "40px 0 20px 0",
203     }}
204     onClick={userLogin}
205 />
206 <p style={{ textAlign: "center" }}>OR</p>

```

```

206 <p style={{ textAlign: "center" }}>OR</p>
207 <MaterialButton
208     title="Request OTP"
209     bgColor="ffffff"
210     textColor="#2874f0"
211     style={{
212         margin: "20px 0",
213     }}
214 />
215 </div>
216 </div>
217 </div>
218 </div>
219 </Modal>
220 <div className="subHeader">
221     { /* Logo */ }
222 {
223     { /* Logo ends here */ }

```

```

226 <div
227     style={{
228         padding: "0 10px",
229     }}
230 >
231 {
232 </div>
233 { /* search component ends here */ }
234
235 { /* right side menu */ }

```

```

236 <div className="rightMenu">
237 |
238   {auth.authenticate ? renderLoggedInMenu() : renderNonLoggedInMenu()}
239   <DropDownMenu
240     menu={
241       <a className="more">
242         <span>More</span>
243         <IoIosArrowDown />
244       </a>
245     }
246     menus={[
247       { label: "Notification Preference", href: "", icon: null },
248       { label: "Sell on", href: "", icon: null },
249       { label: "24x7 Customer Care", href: "", icon: null },
250       { label: "Advertise", href: "", icon: null },
251       { label: "Download App", href: "", icon: null },
252     ]}
253   />
254   <div>
255     <a href={` /cart`} className="cart">
256       <Cart count={Object.keys(cart.cartItems).length} />
257       <span style={{ margin: "0 10px" }}>Cart</span>
258     </a>
259   </div>
260 </div>
261 { /* right side menu ends here */}
262 </div>
263 </div>
264 );
265 };
266
267 export default Header;

```

```

1  .header {
2    width: 100%;
3    height: 56px;
4    background: #2874f0;
5  }
6 |
7  .subHeader {
8    max-width: 1100px;
9    width: 100%;
10   height: 100%;
11   display: flex;
12   margin: 0 auto;
13   align-items: center;
14 }
15
16 .logo {
17   width: 137px;
18   height: 100%;
19   overflow: hidden;
20   display: flex;
21   flex-direction: column;
22   justify-content: center;
23   align-items: flex-end;
24 }
25 |
26 .logo .logoimage {
27   width: 75px;
28 }
29 |
30 .exploreText {
31   font-size: 10px;
32   font-style: italic;
33 }
34 |
35 .plusText {
36   font-size: 10px;
37   margin: 0 2px;

```

```

38     text-transform: capitalize;
39     font-style: italic;
40     color: yellow;
41 }
42
43 .logo .goldenStar {
44     width: 10px;
45     height: 10px;
46 }
47
48 .searchInputContainer {
49     width: 560px;
50     position: relative;
51     background: white;
52     height: 36px;
53     border-radius: 1px;
54     overflow: hidden;
55     display: flex;
56 }
57
58 .searchInput {
59     width: 518px;
60     border: 0;

```

```

61     height: 36px;
62     outline: none;
63     box-sizing: border-box;
64     padding: 0 10px;
65 }
66
67 .searchIconContainer {
68     display: flex;
69     flex: 1;
70     justify-content: center;
71     align-items: center;
72 }
73

```

```

74 .rightMenu {
75     display: flex;
76     flex: 1;
77     justify-content: flex-end;
78     align-items: center;
79 }
80
81
82 .loginButton {
83     display: block;
84     width: 100px;
85     height: 30px;
86     background: white;
87     text-align: center;
88     line-height: 30px;
89     color: #2874f0;
90     font-weight: 500;

```



```

95     .more,
96     .fullName {
97         font-weight: bold;
98         color: ■ #fff;
99         display: flex;
100         align-items: center;
101         margin: 0 20px;
102         cursor: pointer;
103     }
104
105     .cart {
106         font-weight: bold;
107         color: ■ #fff;
108         display: flex;
109         align-items: center;
110         justify-content: center;
111         margin: 0 20px;
112     }
113
114     .authContainer {
115         color: □ #212121;
116         margin: 0 auto;
117         border-radius: 2px;
118         height: 528px;
119         max-width: 750px;
120         min-width: 650px;
121     }
122
123     .row {
124         width: 100%;
125         display: flex;
126         flex-flow: row wrap;
127         height: 528px;
128         max-width: 750px;
129         min-width: 650px;
130     }

```

```

132     .leftspace {
133         width: 40%;
134         height: 100%;
135         color: ■ #fff;
136         letter-spacing: 1px;
137
138         background-color: ■ #2874f0;
139         background-repeat: no-repeat;
140         background-position: center 85%;
141         padding: 40px 33px;
142         box-sizing: border-box;
143     }
144
145     .rightspace {
146         display: flex;
147         flex: 1;
148         flex-direction: column;
149         align-items: center;
150     }
151
152     .LoginInputContainer {
153         width: 100%;
154         box-sizing: border-box;
155         padding: 0 30px;
156         margin: 40px 0;
157     }
158

```

```

1  const mongoose=require('mongoose');
2
3  const userSchema=new mongoose.Schema({
4    firstName:{
5      type:String,
6      required:true,
7      trim:true,
8      min:3,
9      max:20,
10   },
11   lastName:{
12     type:String,
13     required:true,
14     min:3,
15     max:20,
16   },
17   username:{
18     type:String,
19     required:true,
20     trim:true,
21     unique:true,
22     index:true,
23     lowercase:true,
24   },
25   },
26   email:{
27     type:String,
28     required:true,
29     trim:true,
30     unique:true,
31     lowercase:true,
32   },
33   password:{
34     type:String,
35     required:true,
36   },
37   role:{

```

```

44  }
45  exports.signin=(req,res)=>{
46    User.findOne({email:req.body.email})
47    .exec((error,user)=>{
48      if(error) return res.status(400).json({error});
49      if(user)
50      {
51        if(user.role==='user')
52        {
53          const token=jwt.sign({_id:user._id,role:user.role},process.env.JWT_SECRET,{expiresIn:'2d'});
54          const {firstName,lastName,email,role,fullName}=user;
55          res.status(200).json({
56            token,
57            user:{
58              firstName,lastName,email,role,fullName
59            }
60          });
61        }
62        else{
63          return res.status(400).json({
64            message:'Something went wrong'
65          });
66        }
67      }
68      else{
69        return res.status(400).json({
70          message:'Something went wrong'
71        });
72      }
73    });
74  });
75  });
76  }

```



## SIGNUP:

Admin Dashboard

SigninSignup

First Name

Last Name

First Name

Last Name

Email

Email

Password

Password

Submit

## CODE:

```
1  .modal-header {
2    padding: 5px;
3    background-color: #333;
4  }
5  .modal-title {
6    font-size: 18px;
7  }
8  .modal-footer {
9    padding: 5px 15px;
10 }
11
12 .actionBtnContainer {
13   display: flex;
14   align-items: center;
15 }
16
17 .actionBtnContainer > button {
18   border: 0;
19   margin: 0 5px;
20   font-size: 14px;
21 }
22
```

```
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, IoIosClose } from 'react-ionicons';
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 === '') {

```

```

37     if(categoryName === ""){
38         alert('Category Name is required');
39         setShow(false);
40         return ;
41     }
42
43     form.append('name',categoryName);
44     form.append('parentId',parentCategoryId);
45     form.append('categoryImage',categoryImage);
46     dispatch(addCategory(form));
47     setCategoryName('');
48     setParentCategoryId('');
49     // const cat={
50
51
52     //     // categoryName,
53     //     // parentCategoryId,
54     //     // categoryImage
55     // };
56
57     setShow(false);
58 }
59 const handleShow = () => setShow(true);
60 const renderCategories=(categories)=>
61 {
62     let myCategories=[];
63     for(let category of categories){
64         myCategories.push(
65             {
66                 label:category.name,
67                 value:category._id,

```

```

68                 type:category.type,
69                 children:category.children.length>0&& renderCategories(category.children),
70             }
71         );
72     }
73     return myCategories;
74 }
75
76
77 const createCategoryList=(categories,options=[])=>{
78     for(let category of categories){
79         options.push({value:category._id,
80             name:category.name,
81             parentId:category.parentId,
82             type:category.type,
83         });
84     }
85
86     if(category.children.length>0){
87         createCategoryList(category.children,options)
88     }
89     return options;
90 }
91
92
93 const handleCategoryImage=(e)=>{
94     setCategoryImage(e.target.files[0]);
95 }
96 const updateCategory=(e)=>{
97     updateCheckedAndExpandedCategories();

```

```

98     setUpdateCategoryModal(true);
99 }
100
101 const updateCheckedAndExpandedCategories=(e)=>{
102     const categories=createCategoryList(category.categories);
103     const checkedArray=[];
104     const expandedArray=[];
105     checked.length>0&&checked.forEach((categoryId,index)=>{
106         const category=categories.find((category,_index)=>categoryId==category.value);
107         category&&checkedArray.push(category);
108     })
109     expanded.length>0&&expanded.forEach((categoryId,index)=>{
110         const category=categories.find((category,_index)=>categoryId==category.value);
111         category&&expandedArray.push(category);
112     })
113     setCheckedArray(checkedArray);
114     setExpandedArray(expandedArray);
115 }
116
117
118 const handleCategoryInput=(key,value,index,type)=>{
119     if(type=="checked")
120     {
121         const updatedCheckedArray=checkedArray.map((item,_index)=>index==_index?{...item,[key]:value}:{...item});
122         setCheckedArray(updatedCheckedArray);
123     }
124     else if(type=="expanded")
125     {
126         const updatedExpandedArray=expandedArray.map((item,_index)=>index==_index?{...item,[key]:value}:{...item});
127         setExpandedArray(updatedExpandedArray);

```

```

125     }, { type: 'expanded' }
126     {
127       const updatedExpandedArray=expandedArray.map((item, _index)=>index==_index?...item, [key]:
128     }
129   }
130   const updateCategoriesForm=()=>{
131     const form=new FormData();
132     expandedArray.forEach((item, index)=>{
133       form.append('_id', item.value);
134       form.append('name', item.name);
135       form.append('parentId', item.parentId?item.parentId:"");
136       form.append('type', item.type);
137     })
138     checkedArray.forEach((item, index)=>{
139       form.append('_id', item.value);
140       form.append('name', item.name);
141       form.append('parentId', item.parentId?item.parentId:"");
142       form.append('type', item.type);
143     });
144     dispatch(updateCategories(form));
145   }
146 }
147 }
148

```

```

151
152 const renderAddCategoryModal=()=>{
153   return (
154     <Modal
155       show={show}
156       handleClose={handleClose}
157       modalTitle={'Add New Category'}
158     >
159     <input
160       value={categoryName}
161       placeholder={'Category Name'}
162       onChange={(e)=>setCategoryName(e.target.value)}//>
163     <select
164       className='form-control'
165       value={parentCategoryId}
166       onChange={(e)=>setParentCategoryId(e.target.value)}>
167       <option>Select category</option>
168       {createCategoryList(category.categories).map(option=>
169         <option key={option.value} value={option.value}>{option.name}</option>
170       )}</select>
171     <input type="file" name="categoryImage" onChange={handleCategoryImage}/>
172     </Modal>
173   );
174 }
175 const deleteCategory=()=>{
176   updateCheckedAndExpandedCategories();
177   setDeleteCategoryModal(true);
178 }
179
180
181
182
183 const deleteCategories=()=>{
184   const checkedIdsArray=checkedArray.map((item, index)=>({_id:item.value}));
185   const expandedIdsArray=expandedArray.map((item, index)=>({_id:item.value}));
186   const idsArray=expandedIdsArray.concat(checkedIdsArray);
187 }

```

```

188
189   if(checkedIdsArray.length>0)
190   {
191     dispatch(deleteCategoriesAction(checkedIdsArray))
192
193     .then(result=>{
194       if(result){
195         dispatch(getAllCategory())
196         setDeleteCategoryModal(false)
197       }
198     });
199   }
200   setDeleteCategoryModal(false);
201 }
202
203 }
204 const renderDeleteCategoryModal=()=>{
205
206   return (
207     <Modal modalTitle="Confirm"
208       show={deleteCategoryModal}
209       handleClose={()=>setDeleteCategoryModal(false)}
210       buttons={
211         [
212           { label:'No',
213             color:'primary',
214             onClick:()=>{
215               alert('no');
216             }
217           }
218         ]
219       }
220     >

```

```

218     { label: 'Yes',
219       color: 'danger',
220       onClick: deleteCategories
221     }
222   ]
223 }
224 >
225 <h5>Expanded</h5>
226 {
227   expandedArray.map((item,index)=><span key={index}>{item.name}</span>)
228 }
229 <h5>Checked</h5>
230 {
231   checkedArray.map((item,index)=><span key={index}>{item.name}</span>)
232 }
233 </Modal>
234 );
235 }
236 const categoryList = createCategoryList(category.categories);

```

```

237 return (
238   <Layout sidebar>
239     <Container>
240       <Row>
241         <Col md={12}>
242           <div style={{display: 'flex', justifyContent: 'space-between'}}>
243             <h3>Category</h3>
244             <div className='actionBtnContainer'>
245               <span>Actions:</span>
246               <button onClick={handleShow}><IoIosAdd/><span>Add</span></button>
247               <button onClick={deleteCategory}><IoIosTrash/><span>Delete</span></button>
248               <button onClick={updateCategory}><IoIosCloudUpload/><span>Edit</span></button>
249             </div>
250           </div>
251         </Col></Row>
252       </Row>
253       <Col md={12}>
254         <CheckboxTree
255           nodes={renderCategories(category.categories)}
256           checked={checked}
257           expanded={expanded}
258           onCheck={checked => setChecked( checked )}
259           onExpand={expanded => setExpanded(expanded )}
260           icons={{
261             check: <IoIosCheckbox />,
262             uncheck: < IoIosCheckboxOutline/>,
263             halfCheck: <IoIosCheckboxOutline />,
264             expandClose: <IoIosArrowForward />,
265             expandOpen: <IoIosArrowDown/>
266           }}
267         />
268       </Col>
269     </Container>
270   </Layout>
271 );

```

```

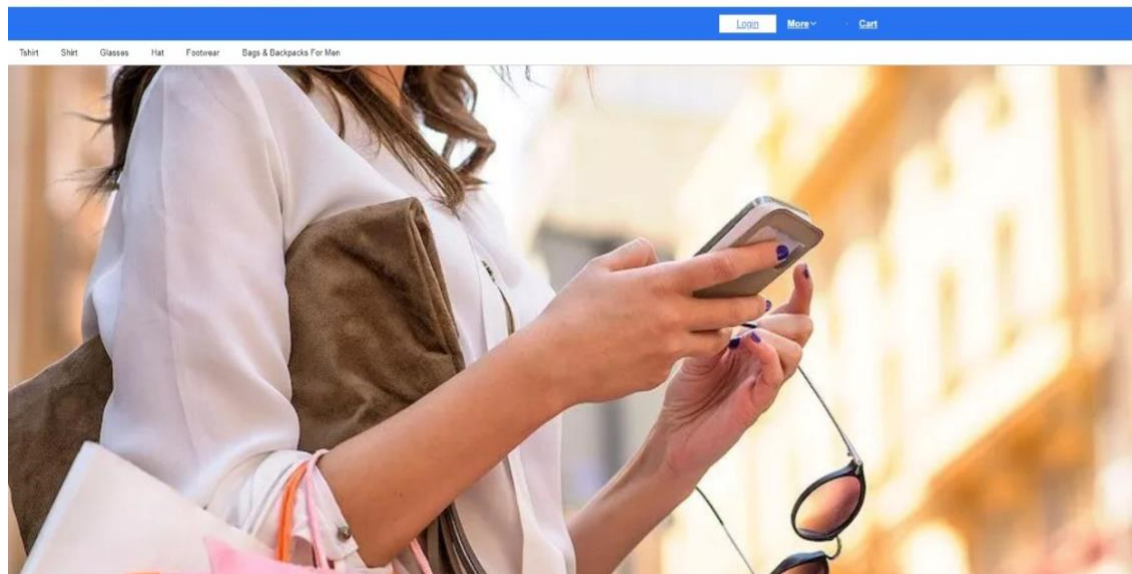
276 <AddCategoryModal
277   show={show}
278   handleClose={()=>setShow(false)}
279   onSubmit={handleClose}
280   modalTitle='Add Categories'
281   categoryName={categoryName}
282   setCategoryName={setCategoryName}
283   parentCategoryId={parentCategoryId}
284   setParentCategoryId={setParentCategoryId}
285   categoryList={categoryList}
286   handleCategoryImage={handleCategoryImage}
287 />
288
289 <UpdateCategoriesModal
290   show={updateCategoryModal}
291   handleClose={()=>setUpdateCategoryModal(false)}
292   onSubmit={updateCategoriesForm}
293   modalTitle='Update Categories'
294   size='lg'
295   expandedArray={expandedArray}
296   checkedArray={checkedArray}
297   handleCategoryInput={handleCategoryInput}
298   categoryList={categoryList}
299 />
300
301
302
303
304
305
306
307 {renderDeleteCategoryModal()}
308
309
310
311

```



# HOME PAGE:

## Smart Fashion Recommender



# CODE:

```
JS index.js .../HomePage M X # style.css .../HomePage M JS index.js .../ProductDetailsPage JS index.js .../ClothingAndAccessories # style.css .../ClothingAndAccessories JS product.js .../controller M JS product.js .../v

ecommerce-front-end > front-end > src > containers > HomePage > JS index.js > [60] HomePage
1 import React from 'react'
2 import Layout from '../components/Layout';
3
4 import Header from '../components/Header';
5
6 import MenuHeader from '../components/MenuHeader';
7 import Carousel from 'react-bootstrap/carousel';
8 import 'bootstrap/dist/css/bootstrap.min.css';
9 import { useState } from 'react';
10 import './style.css';
11 const HomePage = (props) => {
12   const [index, setIndex] = useState(0);
13
14   const handleSelect = (selectedIndex, e) => {
15     setIndex(selectedIndex);
16   };
17
18   return (
19     <layout>
20     <div className="homepage"></div>
21     </Layout>
22   );
23 }
24
25 export default HomePage
```

```
JS index.js .../HomePage M # style.css .../HomePage M X JS index.js .../ProductDetailsPage JS index.js .../ClothingAndAccessories # style.css .../ClothingAndAccessories JS product.js .../controller M JS product.js .../v

ecommerce-front-end > front-end > src > containers > HomePage > # style.css > homepage
1
2 .carousel *{
3   color: white;
4 }
5 .layout{
6   position: fixed;
7 }
8 .homepage{
9   background-image: url('../images/recommendation-engines-for-fashion-6-current-applications-6.jpeg');
10  height: 1000px;
11
12  background-position: fixed;
13  background-repeat: no-repeat;
14  background-size: cover;
15 }
16
```

