

## PROJECT DEVELOPMENT PHASE

## SPRINT – 4

DATE	18 NOVEMBER 2022
TEAM ID	PNT2022TMID02427
PROJECT NAME	SMART FASHION RECOMMENDER APPLICATION

**ADDRESS :**

# Smart Fashion Recommender

manikandan

More

Cart

TshirtShirtGlassesHatFootwearBags & Backpacks For Men

1

LOGIN

manikandan\_m1@gmail.com

2

DELIVERY ADDRESS

☐ manikandan

HOME

8778008026

1/13 Rajaji salai, Kadambathur,  
tamil nadu - 631203

+

ADD NEW ADDRESS

Name

Pincode

Address

City/District/ town

Landmark (Optional)

Address Type

☐ Home ☐ Work

10-digit mobile number

Locality

State

Alternate Phone (Optional)

SAVE AND DELIVER HERE

**CODE:**

ecommerce-front-end > front-end > src > containers > CheckoutPage > JS AddressForm.js > ...

```
1 import React, { useEffect, useState } from "react";
2 import { useDispatch, useSelector } from "react-redux";
3
4 import { addAddress } from "../../actions";
5 import { MaterialButton, MaterialInput } from "../../components/MaterialUI";
6
7 /**
8  * @author
9  * @function AddressForm
10  */
11
12 const AddressForm = (props) => {
13   const { initialData } = props;
14   const [name, setName] = useState(initialData ? initialData.name : "");
15   const [mobileNumber, setMobileNumber] = useState(
16     initialData ? initialData.mobileNumber : ""
17   );
18   const [pinCode, setPinCode] = useState(
19     initialData ? initialData.pinCode : ""
20   );
21   const [locality, setLocality] = useState(
22     initialData ? initialData.locality : ""
23   );
24   const [address, setAddress] = useState(
25     initialData ? initialData.address : ""
26   );
27   const [cityDistrictTown, setCityDistrictTown] = useState(
28     initialData ? initialData.cityDistrictTown : ""
29   );
30   const [state, setState] = useState("");
31   const [landmark, setLandmark] = useState(
32     initialData ? initialData.landmark : ""
33   );
34   const [alternatePhone, setAlternatePhone] = useState(
35     initialData ? initialData.alternatePhone : ""
36   );
37   const [addressType, setAddressType] = useState(
38     initialData ? initialData.addressType : ""
39   );
40   const dispatch = useDispatch();
41   const user = useSelector((state) => state.user);
42   const [submitFlag, setSubmitFlag] = useState(false);
43   const [id, setId] = useState(initialData ? initialData._id : "");
44 }
```

mern-back-end > src > models > JS address.js > ...

```
44   required: true,
45 },
46 landmark: {
47   type: String,
48   min: 10,
49   max: 100,
50 },
51 alternatePhone: {
52   type: String,
53 },
54 addressType: {
55   type: String,
56   required: true,
57   enum: ["home", "work"],
58   required: true,
59 },
60 });
61
62 // B
63 const userAddressSchema = new mongoose.Schema(
64   {
65     user: {
66       type: mongoose.Schema.Types.ObjectId,
67       required: true,
68       ref: "User",
69     },
70     address: [addressSchema],
71   },
72   { timestamps: true }
73 );
74
75 mongoose.model("Address", addressSchema);
76 module.exports = mongoose.model("UserAddress", userAddressSchema);
```

ecommerce-front-end > front-end > src > containers > CheckoutPage > JS AddressForm.js > ...

```
64   };
65
66   useEffect(() => {
67     console.log("addressCount", user.address);
68     if (submitFlag) {
69       console.log("where are we", user);
70       let _address = {};
71       if (id) {
72         address = {
73           _id: id,
74           name,
75           mobileNumber,
76           pinCode,
77           locality,
78           address,
79           cityDistrictTown,
80           state,
81           landmark,
82           alternatePhone,
83           addressType,
84         };
85       } else {
86         _address = user.address.slice(user.address.length - 1)[0];
87       }
88
89       props.onSubmitForm(_address);
90     }
91   }, [user.address]);
92
```

ecommerce-front-end > front-end > src > containers > CheckoutPage > JS AddressForm.js > ...

```
91   }, [user.address]);
92
93   const renderAddressForm = () => {
94     return (
95       <>
96         <div className="flexRow">
97           <div style={inputContainer}>
98             <MaterialInput
99               label="Name"
100               value={name}
101               onChange={(e) => setName(e.target.value)}
102             />
103           </div>
104           <div style={inputContainer}>
105             <MaterialInput
106               label="10-digit mobile number"
107               value={mobileNumber}
108               onChange={(e) => setMobileNumber(e.target.value)}
109             />
110           </div>
111         </div>

```

ecommerce-front-end > front-end > src > containers > CheckoutPage > JS AddressForm.js > ...

```
111 </div>
112 <div className="flexRow">
113   <div style={inputContainer}>
114     <MaterialInput
115       label="Pincode"
116       value={pinCode}
117       onChange={(e) => setPinCode(e.target.value)}
118     />
119   </div>
120   <div style={inputContainer}>
121     <MaterialInput
122       label="Locality"
123       value={locality}
124       onChange={(e) => setLocality(e.target.value)}
125     />
126   </div>
127 </div>
128 <div className="flexRow">
129   <div style={inputContainer}>
130     <MaterialInput
131       label="Address"
132       value={address}
133       onChange={(e) => setAddress(e.target.value)}
134     />
135   </div>
136 </div>
137
138 <div className="flexRow">
139   <div style={inputContainer}>
140     <MaterialInput
141       label="City/District/Town"
142       value={cityDistrictTown}
143       onChange={(e) => setCityDistrictTown(e.target.value)}
144     />
145   </div>
146   <div style={inputContainer}>
```

```
147     <MaterialInput
148       label="State"
149       value={state}
150       onChange={(e) => setState(e.target.value)}
151     />
152   </div>
153 </div>
154 <div className="flexRow">
155   <div style={inputContainer}>
156     <MaterialInput
157       label="Landmark (Optional)"
158       value={landmark}
159       onChange={(e) => setLandmark(e.target.value)}
160     />
161   </div>
162   <div style={inputContainer}>
163     <MaterialInput
164       label="Alternate Phone (Optional)"
165       value={alternatePhone}
166       onChange={(e) => setAlternatePhone(e.target.value)}
167     />
168   </div>
169 </div>
170 <div>
171   <label>Address Type</label>
172   <div className="flexRow">
173     <div>
174       <input
175         type="radio"
176         onClick={() => setAddressType("home")}
177         name="addressType"
178         value="home"
179       />
180       <span>Home</span>
181     </div>
```

ecommerce-front-end > front-end > src > containers > CheckoutPage > JS AddressForm.js > ...

```
180       <span>Home</span>
181     </div>
182     <div>
183       <input
184         type="radio"
185         onClick={() => setAddressType("work")}
186         name="addressType"
187         value="work"
188       />
189       <span>Work</span>
190     </div>
191   </div>
192 </div>
193 <div className="flexRow">
194   <MaterialButton
195     title="SAVE AND DELIVER HERE"
196     onClick={onAddressSubmit}
197     style={{
198       width: "250px",
199       margin: "20px 0",
200     }}
201   />
202 </div>
203 </>
204 );
205 };
206
207 if (props.withoutLayout) {
208   return <div>{renderAddressForm()}</div>;
209 }
210
211 return (
212   <div className="checkoutStep" style={{ background: "#f5faff" }}>
213     <div className={"checkoutHeader"}>
214       <div>
215         <span className="stepNumber">+</span>
216         <span className="stepTitle">{"ADD NEW ADDRESS"}</span>
```

```
217       </div>
218     </div>
219     <div
220       style={{
221         padding: "0 60px",
222         paddingBottom: "20px",
223         boxSizing: "border-box",
224       }}
225     >
226       {renderAddressForm()}
227     </div>
228   </div>
229 );
230 };
231
232 export default AddressForm;
```



ecommerce-front-end > front-end > src > containers > CheckoutPage > JS index.js > [0] CheckoutStep

```
1  import React,{useEffect,useState} from 'react';
2  import { useDispatch,useSelector } from 'react-redux';
3  import { getAddress, getAllCategory, getCartItems,addOrder} from '../actions';
4  import Layout from '../components/Layout';
5  import { Anchor,MaterialButton, MaterialInput } from '../components/MaterialUI';
6  import Card from '../components/UI/Card';
7  import AddressForm from './AddressForm';
8  import './style.css';
9  import PriceDetails from '../components/PriceDetails.js';
10 import CartPage from '../CartPage';
11
12 const CheckoutStep=(props)=>{
13   return (
14     <div className='checkoutStep'>
15       <div onClick={props.onClick} className={`checkoutHeader ${props.active && 'active'}`>
16         <div>
17           <span className='stepNumber'>{props.stepNumber}</span>
18           <span className='stepTitle'>{props.title}</span></div></div>
19           {props.body&& props.body}
20         </div>
21       )
22   }
23
24   const Address=({adr,selectAddress,enableAddressEditForm,confirmDeliveryAddress,onAddressSubmit})=>{
25     return(
26       <div className="flexRow addressContainer">
27         <div>
28           <input name="address" onClick={() => selectAddress(adr)} type="radio" />
29           </div>
30           <div className="flexRow sb addressinfo">
31             {!adr.edit ? (
32               <div style={{ width: "100%" }}>
33                 <div className="addressDetail">
34                   <div>
35                     <span className="addressName">{adr.name}</span>
36                     <span className="addressType">{adr.addressType}</span>
37                     <span className="addressMobileNumber">{adr.mobileNumber}</span>
38                   </div>
39                   {adr.selected && (
40                     <Anchor
41                       name="EDIT"
42                       onClick={() => enableAddressEditForm(adr)}
43                       style={{
44                         fontWeight: "500",
45                         color: "#2874f0",
46                       }}
47                     )
48                   }
49                   <div className="fullAddress">
50                     {adr.address} <br /> {`${adr.state} - ${adr.pinCode}`}
51                   </div>
52                   {adr.selected && (
53                     <MaterialButton
54                       title="DELIVERY HERE"
55                       onClick={() => confirmDeliveryAddress(adr)}
56                       style={{
57                         width: "200px",
58                         margin: "10px 0",
59                       }}
60                     )
61                   }
62                 </div>
63               ) : (
64                 <AddressForm
65                   withoutLayout={true}
66                   onSubmitForm={onAddressSubmit}
67                   initialData={adr}
68                   onCancel={() => {}}
69                 />
70               )
71             )
72           </div>
73         </div>
74       )
75     )
76   }
77 }
```

```

134     useEffect(()=>{
135         const address=user.address.map(adr=>({...adr,selected:false,edit:false}));
136         setAddress(address);
137     },[user.address]);
138     const selectAddress=(addr)=>{
139         // console.log(addr);
140         const updatedAddress= address.map(adr=>
141             adr._id===addr._id?({...adr,selected:true}:{...adr,selected:false});
142         setAddress(updatedAddress)
143     }
144
145     const userOrderConfirmation=()=>{
146         setOrderConfirmation(true);
147         setOrderSummary(false);
148         setpaymentOption(true);
149     }
150     const confirmDeliveryAddress=(addr)=>{
151         setSelectedAddress(addr);
152         setConfirmAddress(true);
153         setOrderSummary(true);
154     }
155     if(confirmOrder)
156     {
157         return(
158             <Layout>
159             <Card>
160             <div>
161                 Thank You</div></Card></Layout>
162         )
163     }

```

```

164     return (
165         <Layout>
166         <div className='cartContainer' style={{alignItems:'flex-start'}}>
167             <CheckoutStep
168                 stepNumber={'1'}
169                 title={'LOGIN'}
170                 active={!auth.authenticate}
171                 body={
172
173                     auth.authenticate?
174                     <div className='loggedInId'>
175                         <span style={{fontWeight:500}}>{auth.user.firstName}</span>
176                         <span style={{margin:'0 5px'}}>{auth.user.email}</span>
177                     </div>:
178                     <div>
179                         <MaterialInput
180                             Label="Email"/>
181                     </div>
182                 }
183             </div>
184
185             <CheckoutStep
186                 stepNumber={'2'}
187                 title={"DELIVERY ADDRESS"}
188                 active={!confirmAddress && auth.authenticate}
189                 body={
190
191                     <>
192                         {confirmAddress ? (
193                             <div className="stepCompleted">['${selectedAddress.name} ${selectedAddress.address} - ${selectedAddress.pinCode}']
194                         ) : (

```

```

ecommerce-front-end > front-end > src > containers > CheckoutPage > JS index.js > [60] CheckoutStep
197     address.map((adr) => (
198       <Address
199         selectAddress={selectAddress}
200         enableAddressEditForm={enableAddressEditForm}
201         confirmDeliveryAddress={confirmDeliveryAddress}
202         onAddressSubmit={onAddressSubmit}
203         adr={adr}
204       />
205     ))
206   ))
207 }
208 </>
209 ]
210 />
211
212 {/* AddressForm */}
213 {confirmAddress ? null : newAddress ? (
214   <AddressForm onSubmitForm={onAddressSubmit} onCancel={() => {}} />
215 ) : auth.authenticate ? (
216   <CheckoutStep
217     stepNumber={"+"}
218     title={"ADD NEW ADDRESS"}
219     active={false}
220     onClick={() => setNewAddress(true)}
221   />
222 ) : null}
223
224
225 <CheckoutStep
226   stepNumber={"3"}
227   title={"ORDER SUMMARY"}
228   active={orderSummary}
229   body={orderSummary ? (<CartPage onlyCartItems={true} />):orderConfirmation?<div className='stepCompleted'>{Object.key
230   />
231   {
232     orderSummary&&<Card style={{

```

```

224
225 <CheckoutStep
226   stepNumber={"3"}
227   title={"ORDER SUMMARY"}
228   active={orderSummary}
229   body={orderSummary ? (<CartPage onlyCartItems={true} />):orderConfirmation?<div className='stepCompleted'>{Object.keys(c
230   />
231   {
232     orderSummary&&<Card style={{
233       margin:'20px 0'
234     }}>
235     <div className='flexRow sb' style={{
236       padding:'20px',
237       alignItems:'center'
238     }}>
239     <p style={{fontSize:"12px"}}>Order confirmation email will be sent to <strong>{auth.user.email} </strong> </p>
240     <MaterialButton
241       title="CONTINUE"
242       onClick={userOrderConfirmation}
243       style={{
244         width:'200px'}}/>
245     </div></Card>
246   }
247
248
249
250 <CheckoutStep
251   stepNumber={"4"}
252   title={"PAYMENT OPTION"}
253   active={paymentOption}
254   body={
255     paymentOption && <div className='stepCompleted'>
256     <div className='flexRow'
257       style={{
258         alignItems:"center",
259         padding:"20px"

```



```

258         alignItems:"center",
259         padding:"20px"
260     })>
261     <input type="radio" name="paymentOption" value="cod"/>
262     <div>Cash on Delivery</div>
263 </div>
264 <MaterialButton
265   title="Confirm Order"
266   onClick={onConfirmOrder}
267   style={{
268     width:"200px",
269     margin: '0 0 20px 20px'}}/>
270 </div>
271   }
272 </div>
273 </div>
274 <PriceDetails
275   totalItem={Object.keys(cart.cartItems).reduce(function (qty, key) {
276     return qty + cart.cartItems[key].qty;
277   }, 0)}
278   totalPrice={Object.keys(cart.cartItems).reduce((totalPrice, key) => {
279     const { price, qty } = cart.cartItems[key];
280     return totalPrice + price * qty;
281   }, 0)}
282 </div>
283 </div>
284 </Layout>
285
286 }
287 export default CheckoutPage;

```

```

1  .checkoutContainer {
2    width: calc(100% - 400px);
3  }
4  .checkoutStep {
5    box-sizing: border-box;
6    background: #fff;
7    box-shadow: 0 1px 1px 0px rgba(0, 0, 0, 0.2);
8    margin-bottom: 10px;
9  }
10 .checkoutHeader {
11   width: 100%;
12   box-sizing: border-box;
13   padding: 10px 20px;
14 }
15 .checkoutHeader.active {
16   background: #2874f0;
17 }
18 .checkoutHeader .stepNumber {
19   display: inline-block;
20   text-align: center;
21   background: #eee;
22   color: #2874f0;
23   border-radius: 3px;
24   font-size: 12px;
25   box-sizing: border-box;
26   padding: 3px 7px;
27 }
28 .checkoutHeader.active .stepNumber {
29   background: #fff;
30 }
31 .stepTitle {
32   display: inline-block;
33   font-size: 14px;
34   font-weight: 500;
35   color: #878787;
36   margin-left: 15px;
37 }

```

ecommerce-front-end > front-end > src > containers > CheckoutPage > # style.css > .checkoutContainer

```
37 }
38 .checkoutHeader.active .stepTitle {
39   color: #fff;
40 }
41 .loggedInId {
42   font-size: 12px;
43   margin: 0px 0 10px 60px;
44 }
45
46 .addressContainer {
47   box-sizing: border-box;
48   padding: 20px;
49   background: #f5faff;
50 }
51 .addressinfo {
52   padding-left: 15px;
53   box-sizing: border-box;
54   width: 100%;
55 }
56 .addressDetail {
57   display: flex;
58   justify-content: space-between;
59   width: 100%;
60   box-sizing: border-box;
61   padding-bottom: 20px;
62   font-weight: 600;
63   font-family: "Roboto";
64   font-size: 12px;
65   color: #333;
66 }
67 .addressType {
68   display: inline-block;
69   margin: 0 10px;
70   text-transform: uppercase;
71   background: #eee;
72   font-weight: 500;
73   padding: 1px 5px;
74   font-size: 10px;
```

```
74   font-size: 10px;
75   border-radius: 2px;
76   color: #777777;
77 }
78 .fullAddress {
79   font-size: 12px;
80   letter-spacing: 1px;
81   line-height: 18px;
82 }
83 .stepCompleted {
84   padding: 0 0 20px 60px;
85   font-size: 12px;
86 }
87 .stepCompleted {
88   padding: 0 0 20px 60px;
89   font-size: 12px;
90 }
```

mern-back-end > src > controller > JS address.js > addAddress > addAddress > \$set

```
1  const address = require('../models/address');
2  const UserAddress=require('../models/address');
3
4  exports.addAddress=(req,res)=>{
5      const {payload}=req.body;
6      if(payload.address){
7          if(payload.address._id){
8              UserAddress.findOneAndUpdate(
9                  {user:req.user._id,"address._id":payload.address._id},
10                 {
11                     $set:{
12                         "address.$":payload.address,
13                     },
14                 }
15             ).exec((error,address)=>{
16                 if(error) return res.status(400).json({error});
17                 if(address){
18                     res.status(201).json({address});
19                 }
20             });
21         }
22     }
23 }
24 else{
25     UserAddress.findOneAndUpdate(
26         {
27             user:req.user._id,
28             {
29                 $push:{
30                     address:payload.address,
31                 },
32             },
33             {new:true,upsert:true}
34         ).exec((error,address)=>{
35             if(error) return res.status(400).json({error});
36             if(address){
37                 res.status(201).json({address});
```

```
38     }
39     });
40 }
41
42 }
43 else{
44     res.status(400).json({error:'Params address required'});
45 }
46 };
47
48 exports.getAddress=(req,res)=>{
49     UserAddress.findOne({user:req.user._id})
50     .exec((error,userAddress)=>{
51         if(error) return res.status(400).json({error});
52         if(userAddress){
53             res.status(200).json({userAddress});
54         }
55     });
56 }
```

mern-back-end > src > models > JS address.js > ...

```
1  const mongoose = require("mongoose");
2
3  // C
4  const addressSchema = new mongoose.Schema({
5    name: {
6      type: String,
7      required: true,
8      trim: true,
9      min: 3,
10     max: 50,
11   },
12   mobileNumber: {
13     type: String,
14     required: true,
15     trim: true,
16   },
17   pinCode: {
18     type: String,
19     required: true,
20     trim: true,
21   },
22   locality: {
23     type: String,
24     required: true,
25     trim: true,
26     min: 10,
27     max: 100,
28   },
29   address: {
30     type: String,
31     required: true,
32     trim: true,
33     min: 10,
34     max: 100,
35   },
36   cityDistrictTown: {
37     type: String,
```

```
38     required: true,
39     trim: true,
40   },
41   state: {
42     type: String,
43     required: true,
44     required: true,
45   },
46   landmark: {
47     type: String,
48     min: 10,
49     max: 100,
50   },
51   alternatePhone: {
52     type: String,
53   },
54   addressType: {
55     type: String,
56     required: true,
57     enum: ["home", "work"],
58     required: true,
59   },
60 });
61
62 // B
63 const userAddressSchema = new mongoose.Schema(
64   {
65     user: {
66       type: mongoose.Schema.Types.ObjectId,
67       required: true,
```

```

68     ref: "User",
69   },
70   address: [addressSchema],
71 },
72 { timestamps: true }
73 );
74
75 mongoose.model("Address", addressSchema);
76 module.exports = mongoose.model("UserAddress", userAddressSchema);

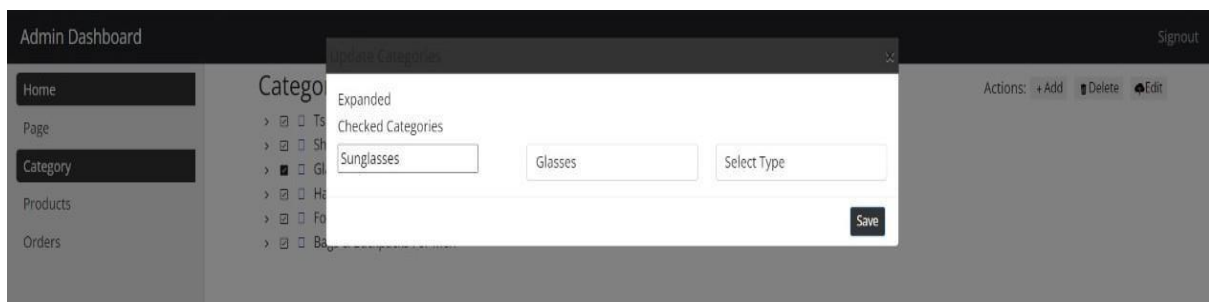
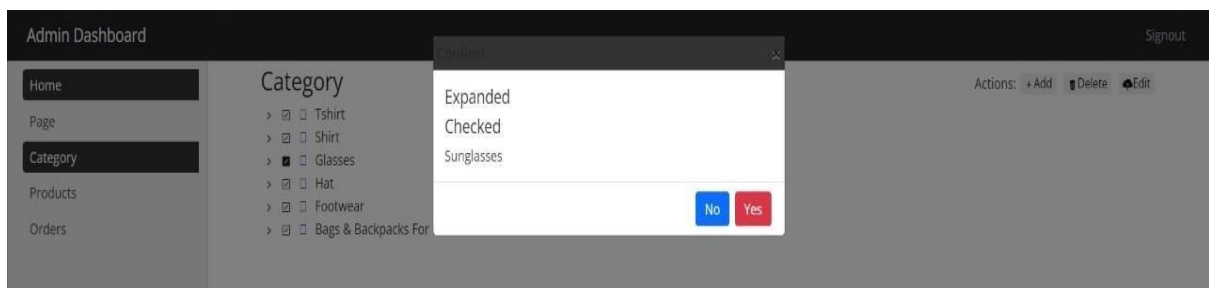
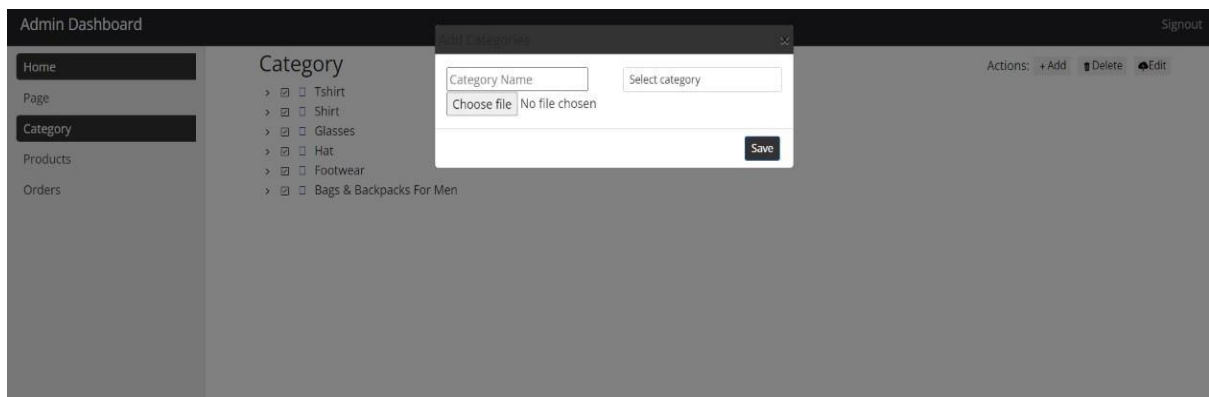
```

```

mern-back-end > src > routes > JS address.js > ...
1  const express=require('express');
2  const {requiresignin,userMiddleware}=require('../common-middleware')
3  const {addAddress,getAddress}=require('../controller/address')
4  const router=express.Router();
5  router.post('/user/address/create',requiresignin,userMiddleware,addAddress);
6  router.post('/user/getaddress',requiresignin,userMiddleware,getAddress);
7
8  module.exports=router;

```

## CATEGORY:





```

1  const express = require('express');
2  const { signup, signin, signout } = require('../controller/admin/auth');
3  const { validateSignupRequest, isRequestValidated, validateSigninRequest } = require('../validators/auth');
4  const router = express.Router();
5  const { requireSignin } = require('../common-middleware');
6
7
8  router.post('/admin/signup', validateSignupRequest, isRequestValidated, signup);
9  router.post('/admin/signin', validateSigninRequest, isRequestValidated, signin);
10 router.post('/admin/signout', signout);
11
12 // router.post('/profile', requireSignin, (req, res) => {
13 //   res.status(200).json({ user: 'profile' })
14 // });
15
16 module.exports = router;

```

```

mern-back-end > src > controller > admin > 25 auth.js > ...
1  const User = require('../models/user');
2  const jwt = require("jsonwebtoken");
3
4  const shortid = require('shortid');
5  const { validationResult } = require('express-validator');
6  exports.signup = (req, res) => {
7
8    User.findOne({ email: req.body.email })
9    .exec(async (error, user) => {
10      if (user)
11        return res.status(400).json({
12          message: 'Admin already registered'
13        });
14      const {
15        firstName,
16        lastName,
17        email,
18        password
19      } = req.body;
20
21      const _user = new User({
22        firstName,
23        lastName,
24        email,
25
26        password,
27
28        username: shortid.generate(),
29        role: 'admin'
30      });
31
32      user.save((error, data) => {
33        if (error) {
34          console.log(error);
35          return res.status(400).json({
36            message: 'something went wrong'
37          });

```

```

54      if (user.role === 'admin')
55      {
56        const token = jwt.sign({ _id: user._id, role: user.role }, process.env.JWT_SECRET, { expiresIn: '2d' });
57        const { firstName, lastName, email, role, fullName, password } = user;
58        if (ispassword === password)
59        {
60          res.status(200).json({
61            token,
62            user: {
63              firstName, lastName, email, role, fullName
64            }
65          });
66        }
67        else {
68          return res.status(400).json({
69            message: 'Something went wrong'
70          });
71        }
72      }
73
74    }
75    else {
76      return res.status(400).json({
77        message: 'Something went wrong'
78      });
79    }
80  });
81
82  });
83 }
84 exports.signout = (req, res) => {
85   res.clearCookie('token');
86   res.status(200).json({
87     message: 'Signout successfully...!'
88   });
89 }

```

# ADMIN DASHBOARD:

Admin Dashboard

SigninSignup

Email

Password

Submit

```
1 const express = require('express');
2 const { signup, signin, signout } = require('../controller/admin/auth');
3 const { validateSignupRequest, isRequestValidated, validateSigninRequest } = require('../validators/auth');
4 const router = express.Router();
5 const { requireSignin } = require('../common-middleware');
6
7
8 router.post('/admin/signup', validateSignupRequest, isRequestValidated, signup);
9 router.post('/admin/signin', validateSigninRequest, isRequestValidated, signin);
10 router.post('/admin/signout', signout);
11
12 // router.post('/profile', requireSignin, (req, res) => {
13 //   res.status(200).json({ user: 'profile' })
14 // });
15
16 module.exports = router;
```

```
1 import React, { useState } from 'react';
2 import Layout from '../components/Layout';
3 import { Container, Form, Row, Col, Button } from 'react-bootstrap';
4 import Input from '../components/UI/Input';
5 import { login } from '../actions';
6 import { useDispatch, useSelector } from 'react-redux';
7 import { Redirect } from 'react-router-dom';
8 import { useEffect } from 'react';
9
10
11 const Signin = (props) => {
12   const [email, setEmail] = useState('');
13   const [password, setPassword] = useState('');
14   const [error, setError] = useState('');
15   const auth = useSelector(state => state.auth);
16   const dispatch = useDispatch();
17
18
19   const userLogin = (e) => {
20     e.preventDefault();
21     const user = {
22       email, password
23     };
24     dispatch(login(user));
25   }
26
27   if (auth.authenticate) {
28     return <Redirect to="/" />
29   }
30
31
32
33
34
35   return (
36     <Layout>
37     <Container>
```

```

38
39     <Row style={{marginTop:'50px'}}>
40     <Col md={{span:6,offset:3}}>
41     <Form onSubmit={userLogin}>
42     <Input
43     label="Email"
44     placeholder="Email"
45     value={email}
46     type="email"
47     onChange={(e)=>setEmail(e.target.value)}
48     />
49     <Input
50     label="Password"
51     placeholder="Password"
52     value={password}
53     type="password"
54     onChange={(e)=>setPassword(e.target.value)}
55     />
56
57
58
59     <Button variant="primary" type="submit">
60     Submit
61     </Button>
62   </Form>
63 </Col></Row>
64 </Container>
65 </Layout>
66
67 )
68
69 export default Signin

```

```

1  import React,{useState} from 'react';
2  import Layout from '../components/Layout';
3  import {Container,Form,Row,Col,Button} from 'react-bootstrap';
4  import Input from '../components/UI/Input';
5  import{Redirect} from'react-router-dom';
6  import { useDispatch, useSelector } from 'react-redux';
7  import { signup } from '../actions';
8  import { useEffect } from 'react';
9  const Signup=(props)=>{
10     const [firstName, setFirstName] = useState("");
11     const [lastName, setLastName] = useState("");
12     const [email, setEmail] = useState("");
13     const [password, setPassword] = useState("");
14     const [error, setError] = useState("");
15     const auth = useSelector((state) => state.auth);
16     const user=useSelector((state)=>state.user);
17     const dispatch=useDispatch();

```

```

19
20 const userSignup=(e)=>{
21   e.preventDefault();
22
23   const user={
24     firstName,
25     lastName,
26     email,
27     password
28   };
29   dispatch(signup(user));
30 };
31 if(auth.authenticate){
32   return <Redirect to={'/'}/>
33 }
34 if(user.loading){
35   return <p>Loading...!</p>
36 }

```

```

37
38
39 return(
40   <Layout>
41     <Container>
42
43
44       <Row style={{marginTop:'50px'}}>
45         <Col md={{span:6,offset:3}}>
46           <Form onSubmit={userSignup}>
47             <Row>
48               <Col md={6}>
49                 <Input
50                   label="First Name"
51                   placeholder="First Name"
52                   value={firstName}
53                   type="text"
54                   onChange={(e)=>setFirstName(e.target.value)}
55                 />
56               </Col>
57               <Col md={6}>
58                 <Input
59                   label="Last Name"
60                   placeholder="Last Name"
61                   value={lastName}
62                   type="text"
63                   onChange={(e)=>setLastName(e.target.value)}
64                 />

```

```

64     />
65     </Col>
66   </Row>
67
68   <Input
69     label="Email"
70     placeholder="Email"
71     value={email}
72     type="email"
73     onChange={(e)=>setEmail(e.target.value)}
74   />
75   <Input
76     label="Password"
77     placeholder="Password"
78     value={password}
79     type="password"
80     onChange={(e)=>setPassword(e.target.value)}
81 />
82
83
84
85
86   <Button variant="primary" type="submit">
87     Submit
88   </Button>
89 </Form>
90 </Col></Row>

```

```

87   Submit
88 </Button>
89 </Form>
90 </Col></Row>
91 </Container>
92 </Layout>
93 )
94 }
95 export default Signup

```

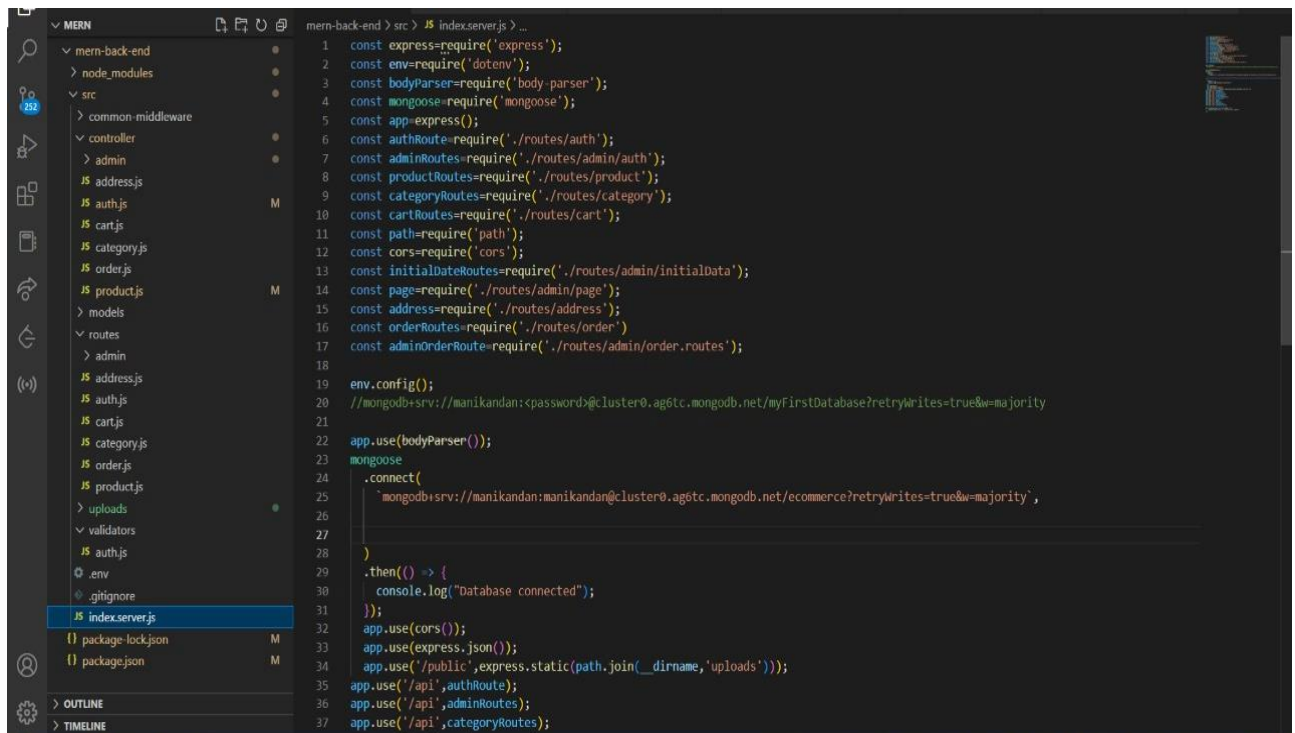
```

1  const express = require('express');
2  const { signup, signin, signout } = require('../controller/admin/auth');
3  const { validateSignupRequest, isRequestValidated, validateSigninRequest } = require('../validators/auth');
4  const router = express.Router();
5  const { requireSignin } = require('../common-middleware');
6
7
8  router.post('/admin/signup', validateSignupRequest, isRequestValidated, signup);
9  router.post('/admin/signin', validateSigninRequest, isRequestValidated, signin);
10 router.post('/admin/signout', signout);
11
12 // router.post('/profile', requireSignin, (req, res) => {
13 //   res.status(200).json({ user: 'profile' })
14 // });
15
16 module.exports = router;

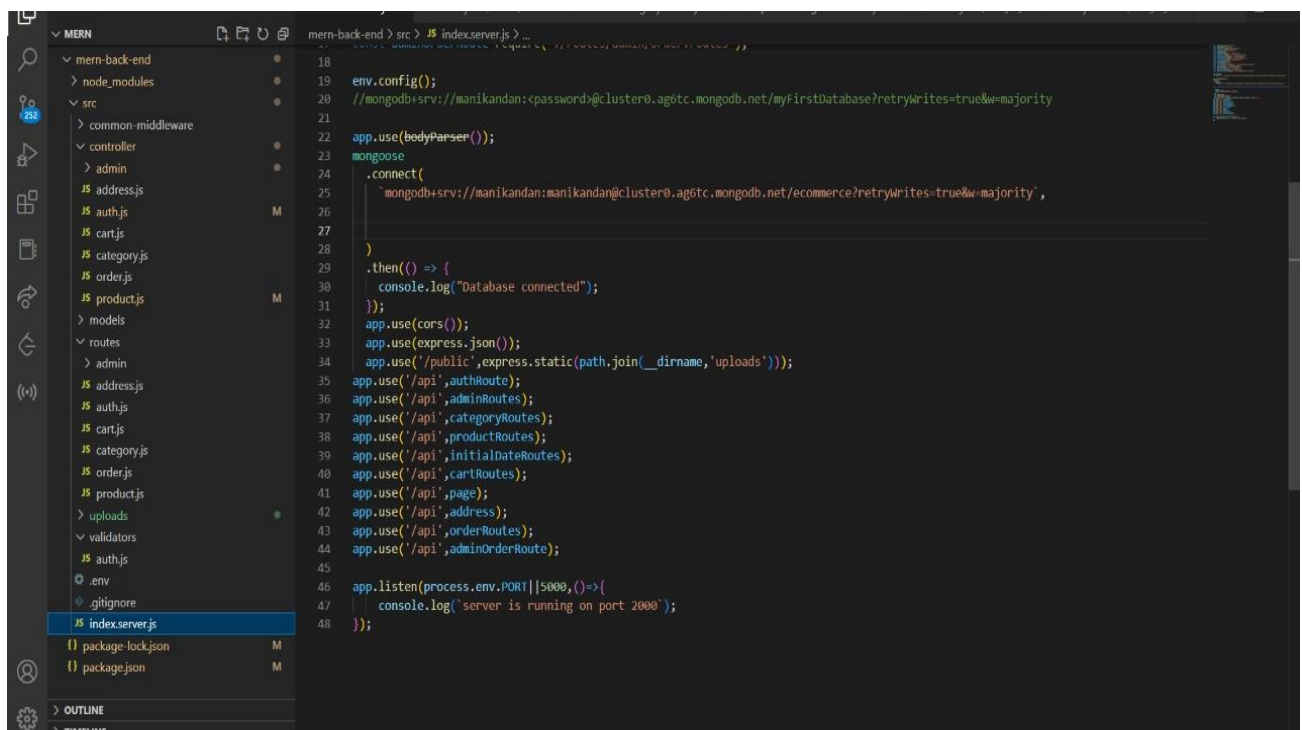
```



## Backend code for running:



```
1 const express=require('express');
2 const env=require('dotenv');
3 const bodyParser=require('body-parser');
4 const mongoose=require('mongoose');
5 const app=express();
6 const authRoute=require('./routes/auth');
7 const adminRoutes=require('./routes/admin/auth');
8 const productRoutes=require('./routes/product');
9 const categoryRoutes=require('./routes/category');
10 const cartRoutes=require('./routes/cart');
11 const path=require('path');
12 const cors=require('cors');
13 const initialDateRoutes=require('./routes/admin/initialData');
14 const page=require('./routes/admin/page');
15 const address=require('./routes/address');
16 const orderRoutes=require('./routes/order');
17 const adminOrderRoute=require('./routes/admin/order.routes');
18
19 env.config();
20 //mongodb+srv://manikandan:password@cluster0.ag6tc.mongodb.net/myFirstDatabase?retryWrites=true&w=majority
21
22 app.use(bodyParser());
23 mongoose
24 .connect(
25   'mongodb+srv://manikandan:manikandan@cluster0.ag6tc.mongodb.net/ecommerce?retryWrites=true&w=majority',
26 )
27
28
29 .then(() => {
30   console.log("Database connected");
31 });
32 app.use(cors());
33 app.use(express.json());
34 app.use('/public',express.static(path.join(__dirname,'uploads')));
35 app.use('/api',authRoute);
36 app.use('/api',adminRoutes);
37 app.use('/api',categoryRoutes);
```



```
18
19 env.config();
20 //mongodb+srv://manikandan:password@cluster0.ag6tc.mongodb.net/myFirstDatabase?retryWrites=true&w=majority
21
22 app.use(bodyParser());
23 mongoose
24 .connect(
25   'mongodb+srv://manikandan:manikandan@cluster0.ag6tc.mongodb.net/ecommerce?retryWrites=true&w=majority',
26 )
27
28
29 .then(() => {
30   console.log("Database connected");
31 });
32 app.use(cors());
33 app.use(express.json());
34 app.use('/public',express.static(path.join(__dirname,'uploads')));
35 app.use('/api',authRoute);
36 app.use('/api',adminRoutes);
37 app.use('/api',categoryRoutes);
38 app.use('/api',productRoutes);
39 app.use('/api',initialDateRoutes);
40 app.use('/api',cartRoutes);
41 app.use('/api',page);
42 app.use('/api',address);
43 app.use('/api',orderRoutes);
44 app.use('/api',adminOrderRoute);
45
46 app.listen(process.env.PORT||5000,()=>{
47   console.log("server is running on port 2000");
48 });
```