



Web App Design with React Final Project

Ayushjav – Sep-30-2022-fesd-burger_online_order-project-
originally due: **10/01/2022**

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

1. Using an online API of your choice (or if no API can be found, using an array for in-memory storage is okay as well), create a React project of your choice. You will be working on this for the next three weeks.
2. Project must meet the following criteria:
 - a. Use React Router and have at least 3 pages
 - b. Use React Bootstrap or an alternative styling library
 - c. Contain at least 10 components
 - d. Allow for all CRUD operations



PROMINEO TECH

Description: My Web API Burger-Online-Order, is a working API.

1. My goal is to understand the essence of React, understand what processes are behind it, and want to master it. First of all, I didn't want to use ready-made information, I just wanted to write the code myself. The idea of this project is that the user directly registers and becomes a member of our restaurant, chooses his favorite burger ingredients, sees the price of the burger, the user only monitors his order, and provides delivery information to the restaurant. I used google firebase. All restaurant orders(data) are based on it. Right now it's working without any problems. I will further develop and perfect the project.

2. My React Project Burger-Online -Order

- **6 pages:** App, BurgerPage, LoginPage, OrderPage, ShippingPage & SignupPage.
- **Uses:** React Routerv6, Redux & the React Bootstrap library
- **Contains 15 components :** BuildControl, BildControls, Burger, BurgerIngredient, ContactData, General, HamburgerMenu, Logo, Menu, MenuItem, Order, OrderSummary, SideBar & Toolbar.
- **Contains (redux) 1 actions & 6 js files:** burgerActions,loginActions, orderActions & signupActions.
- **Contains 1 reducers & 3 js files:** burgerReducer, orderReducer & signupLoginReducers.
- **Implements all 4 GRUD Oreations:** Create, Read, Update & Delete

URL to GitHub Repository:

https://github.com/AyushjavBat/order_burger

Visit this URL :

<https://ayushjavbat.github.io/login>

Screenshots of Code:

Index.js



```
js index.js X
src > JS index.js > [?] reducers
  1 import React from "react"; 6.9k (gzipped: 2.7k)
  2 import ReactDOM from "react-dom/client"; 513 (gzipped: 319)
  3 import "./index.css";
  4 import App from "./pages/App";
  5 import reportWebVitals from "./reportWebVitals";
  6 import { BrowserRouter } from "react-router-dom"; 21.5k (gzipped: 7
  7
  8 import { createStore, applyMiddleware, combineReducers,
  9 compose } from "redux"; 4k (gzipped: 1.5k)
 10 import { Provider } from "react-redux"; 6.9k (gzipped: 2.7k)
 11
 12 import thunk from "redux-thunk"; 427 (gzipped: 269)
 13 import burgerReducer from "./redux/reducer/burgerReducer";
 14 import orderReducer from "./redux/reducer/orderReducer";
 15 import signupReducer from "./redux/reducer/signupLoginReducer";
 16
 17 const loggerMiddleware = store => {
 18   return next => {
 19     return action => {
 20       console.log("MyLoggerMiddleware : Dispatching ==> ",
 21         action);
 22       console.log("MyLoggerMiddleware : State BEFORE ==> ",
 23         store.getState());
 24       const result = next(action);
 25       console.log("MyLoggerMiddleware : State AFTER ==> ", store.
 26         getState());
 27       return result;
 28     };
 29   };
 30 };
 31
 32 const composeEnhancers = window.
 33   __REDUX_DEVTOOLS_EXTENSION_COMPOSE__ || compose;
 34
 35 const reducers = combineReducers({
 36   burgerReducer,
 37   orderReducer,
 38   signupReducer,
 39 });
 40
 41 const middlewares = [loggerMiddleware, thunk];
 42
 43 const store = createStore(
 44   reducers,
 45   composeEnhancers(applyMiddleware(...middlewares))
 46 );
 47
 48 const root = ReactDOM.createRoot(document.getElementById
 49   ("root"));
 50 root.render(
 51   <Provider store={store}>
 52     <BrowserRouter>
 53       <React.StrictMode>
 54         | <App />
 55       </React.StrictMode>
 56     </BrowserRouter>
 57   </Provider>
 58 );
 59
 60 // If you want to start measuring performance in your app, pass
 61 // a function
 62 // to log results (for example: reportWebVitals(console.log))
 63 // or send to an analytics endpoint. Learn more: https://bit.ly/
```



Axios.-orders.js (api)

```
s axios-orders.js ×
src > JS axios-orders.js > ...
1 import axios from "axios"; 38.7k (gzipped: 13.3k)
2
3 const instance = axios.create({
4   baseURL: "https://burger-a1f10-default.firebaseio.com/",
5 });
6
7 export default instance;
8 |
```

Pages:

App.js:



```
js index.js X
src > pages > App > js index.js > App > componentDidMount
1 import React, { Component } from "react"; 6.9k (gzipped: 2.7k)
2 import { Redirect } from "react-router-dom"; 15.3k (gzipped: 5.8k)
3 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
4 import css from "./style.module.css";
5
6 import Toolbar from "../../components/Toolbar";
7 import BurgerPage from "../../BurgerPage";
8 import SideBar from "../../components/SideBar";
9 import OrderPage from "../../OrderPage";
10 import { Route, Switch } from "react-router-dom"; 14.8k (gzipped: 5
11 import ShippingPage from "../../ShippingPage";
12 import LoginPage from "../../LoginPage";
13 import SignupPage from "../../SignupPage";
14 import Logout from "../../components/Logout";
15 import * as actions from "../../redux/actions/loginActions";
16 import * as signupActions from "../../redux/actions/
signupActions";
17
18 class App extends Component {
19   state = {
20     showSidebar: false,
21   };
22
23   toggleSideBar = () => {
24     this.setState(prevState => {
25       return { showSidebar: !prevState.showSidebar };
26     });
27   };
28
29   componentDidMount = () => {
30     const token = localStorage.getItem("token");
31     const userId = localStorage.getItem("userId");
32     const expireDate = new Date(localStorage.getItem
("expireDate"));
33
34     const refreshToken = localStorage.getItem("refreshToken");
35
36     if (token) {
37       if (expireDate > new Date()) {
38         //auto logging and not expired Token
39         this.props.autoLogin(token, userId);
40
41         //counted Token time and auto logout
42         this.props.autologoutAfterMillisec(
43           expireDate.getTime() - new Date().getTime()
44         );
45       } else {
46         //Expired token time
47         this.props.logout();
48       }
49     }
50   };
51
52   render() {
53     return (
54       <div>
55         <Toolbar toggleSideBar={this.toggleSideBar} />
56         <SideBar
57           showSidebar={this.state.showSidebar}
58           toggleSideBar={this.toggleSideBar}
59         />
60         <main className={css.Content}>
```



```
61      <main className={css.Content}>
62        {this.props.userId ? (
63          <Switch>
64            <Route path='/logout' component={Logout} />
65            <Route path='/orders' component={OrderPage} />
66            <Route path='/ship' component={ShippingPage} />
67            <Route path='/' component={BurgerPage} />
68          </Switch>
69        ) : (
70          <Switch>
71            <Route path='/signup' component={SignupPage} />
72            <Route path='/login' component={LoginPage} />
73            <Redirect to='/login' />
74          </Switch>
75        )}
76      </main>
77    </div>
78  );
79}
80
81
82 const mapStateToProps = state => {
83   return {
84     userId: state.signupReducer.userId,
85   };
86 };
87
88 const mapDispatchToProps = dispatch => {
89   return {
90     autoLogin: (token, userId) =>
91       dispatch(actions.loginUserSuccess(token, userId)),
92     logout: () => dispatch(signupActions.logout()),
93     autoLogoutAfterMillisec: () =>
94       dispatch(signupActions.autoLogoutAfterMillisec()),
95   };
96 };
97
98 export default connect(mapStateToProps, mapDispatchToProps)(App);
```



PROMINEO TECH

```
src > pages > App > # style.module.css > .Content p
```

```
1  .Content {  
2    margin-top: 72px;  
3    text-align: center;  
4  }  
5  
6  .Content p {  
7    font-size: 25px;  
8  }
```

BurgerPage:

Index.js:



```
JS index.js X
src > pages > BurgerPage > JS index.js > 🏷️ BurgerPage > ⚡ continueOrder
  1 import React, { Component } from "react"; 6.9k (gzipped: 2.7k)
  2
  3 import Burger from "../../components/Burger";
  4 import BuildControls from "../../components/BuildControls";
  5 import Modal from "../../components/General/Modal";
  6 import OrderSummary from "../../components/OrderSummary";
  7 import axios from "../../axios-orders";
  8 import Spinner from "../../components/General/Spinner";
  9 import * as actions from "../../redux/actions/burgerActions";
 10
 11 class BurgerPage extends Component {
 12   state = {
 13     confirmOrder: false,
 14   };
 15
 16   continueOrder = () => {
 17     this.props.history.push("/ship");
 18   };
 19
 20   showConfirmModal = () => {
 21     this.setState({ confirmOrder: true });
 22   };
 23
 24   closeConfirmModal = () => {
 25     this.setState({ confirmOrder: false });
 26   };
 27
 28   render() {
 29     return (
 30       <div>
 31         <Modal
 32           closeConfirmModal={this.closeConfirmModal}
 33           show={this.state.confirmOrder}
 34         >
 35           {this.state.loading ? (
 36             <Spinner />
 37           ) : (
 38             <OrderSummary
 39               onCancel={this.closeConfirmModal}
 40               onContinue={this.continueOrder}
 41             />
 42           )}
 43         </Modal>
 44
 45         <Burger />
 46
 47         <BuildControls
 48           showConfirmModal={this.showConfirmModal}
 49           ortsNemeh={this.props.addIngredient}
 50           ortsHasah={this.props.removeIngredient}
 51         />
 52       </div>
 53     );
 54   }
 55 }
 56
 57 export default BurgerPage;
```



PROMINEO TECH

LoginPage:

Index.js:



PROMINEO TECH

```
src > pages > LoginPage > js index.js > ts Login > render
  1 import React, { Component } from "react"; 6.9k (gzipped: 2.7k)
  2 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
  3
  4 import Button from "../../components/General/Button";
  5 import css from "./style.module.css";
  6 import * as actions from "../../redux/actions/loginActions";
  7 import Spinner from "../../components/General/Spinner";
  8 import { Redirect } from "react-router-dom"; 15.3k (gzipped: 5.8k)
  9
 10 class Login extends Component {
 11   state = {
 12     email: "",
 13     password: "",
 14   };
 15
 16   changeEmail = e => {
 17     this.setState({ email: e.target.value });
 18   };
 19
 20   changePassword = e => {
 21     this.setState({ password: e.target.value });
 22   };
 23
 24   login = () => {
 25     this.props.login(this.state.email, this.state.password);
 26   };
 27
 28   render() {
 29     return (
 30       <div className={css.Login}>
 31         <h1>Login form</h1>
 32
 33         {this.props.userId && <Redirect to="/orders" />}
 34
 35         <input onChange={this.changeEmail} type='email'
 36           placeholder='Email' />
 37
 38         <input
 39           onChange={this.changePassword}
 40           type='password'
 41           placeholder='Password'
 42         />
 43
 44         {this.props.logginIn && <Spinner />}
 45
 46         {this.props.firebaseioError && (
 47           <div style={{ color: "red" }}>
 48             {this.props.firebaseioError} code is : {this.props.firebaseioErrorCode}
 49           </div>
 50         )}
 51
 52         <Button text='LOG IN' btnType='Success' clicked={this.
 53           login} />
 54       </div>
 55     );
 56   }
 57
 58   const mapStateToProps = state => {
 59     return {
 60       logginIn: state.signupReducer.logginIn,
 61       firebaseError: state.signupReducer.firebaseioError,
```



```
return {
  logginIn: state.signupReducer.logginIn,
  firebaseError: state.signupReducer.firebaseioError,
  firebaseErrorCode: state.signupReducer.firebaseioErrorCode,
  userId: state.signupReducer.userId,
};

const mapStateToProps = dispatch => {
  return {
    login: (email, password) => dispatch(actions.loginUser
      (email, password)),
  };
}

export default connect(mapStateToProps, mapDispatchToProps)
(Login);
```

Style.module.css:

```
src > pages > LoginPage > # style.module.css > .Login input
1  .Login {
2    width: 100%;
3    border: 1px solid #eee;
4    box-sizing: border-box;
5    box-shadow: 0 2px 3px #ccc;
6    padding: 10px;
7    text-align: center;
8  }
9  .Login input {
10   display: block;
11   font-size: 18px;
12   padding: 5px;
13   margin: 10px auto;
14 }
```

OrderPage:



PROMINEO TECH

Index.js:



```
> pages > OrderPage > js index.js > OrderPage > render
1 import React from "react"; 6.9k (gzipped: 2.7k)
2 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
3
4 import css from "./style.module.css";
5 import Spinner from "../../components/General/Spinner";
6 import Order from "../../components/Order";
7 import * as actions from "../../redux/actions/orderActions";
8
9 class OrderPage extends React.Component {
10   componentDidMount = () => {
11     this.props.loadOrders(this.props.userId);
12   }
13
14   render() {
15     //console.log("=====", JSON.stringify(this.state.orders));
16     return (
17       <div>
18         {this.props.loading ? (
19           <Spinner />
20         ) : (
21           this.props.orders.map(el => <Order key={el[0]} order={el[1]} />)
22         )}
23       </div>
24     );
25   }
26 }
27
28 const mapStateToProps = state => {
29   return {
30     orders: state.orderReducer.orders,
31     loading: state.orderReducer.loading,
32     userId: state.signupReducer.userId,
33   };
34 };
35
36 const mapDispatchToProps = dispatch => {
37   return {
38     loadOrders: userId => dispatch(actions.loadOrders(userId)),
39   };
40 };
41
42 export default connect(mapStateToProps, mapDispatchToProps)
  (OrderPage);
```



PROMINEO TECH

ShippingPage:

Index.js:



```
8 import ContactData from "../../components/ContactData";
9
10 class ShippingPage extends React.Component {
11   cancelOrder = () => {
12     this.props.history.goBack();
13   };
14   showContactData = () => {
15     this.props.history.replace("/ship/contact");
16   };
17
18   render() {
19     return (
20       <div className={css.ShippingPage}>
21         <p style={{ fontSize: "24px" }}>
22           <strong>We hope your order will be delicious.</strong>
23         </p>
24         <p style={{ fontSize: "24px" }}>
25           <strong>Total Price : {this.props.price}$</strong>
26         </p>
27         <Burger />
28         <Button
29           clicked={this.cancelOrder}
30           btnType='Danger'
31           text='ORDER CANCEL'
32         />
33         <Button
34           clicked={this.showContactData}
35           btnType='Success'
36           text='ENTER SHIPPING INFORMATION'
37         />
38         <Route path='/ship/contact'>
39           <ContactData />
40         </Route>
41       </div>
42     );
43   }
44 }
45
46 const mapStateToProps = state => {
47   return {
48     price: state.burgerReducer.totalPrice,
49   };
50 };
51
52 export default connect(mapStateToProps)(ShippingPage);
53
```



PROMINEO TECH

SignupPage:

Index.js:



PROMINEO TECH

```
is index.js  X
src > pages > SignupPage > js index.js > Signup > render
  1 import React, { Component } from "react"; 6.9k (gzipped: 2.7k)
  2 import Button from "../../components/General/Button";
  3
  4 import css from "./style.module.css";
  5 import * as actions from "../../redux/actions/signupActions";
  6 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
  7 import Spinner from "../../components/General/Spinner";
  8
  9 import { Redirect } from "react-router-dom"; 15.3k (gzipped: 5.8k)
10
11 class Signup extends Component {
12   state = {
13     email: "",
14     password1: "",
15     password2: "",
16     error: ""
17   };
18
19   changeEmail = e => {
20     this.setState({ email: e.target.value });
21   };
22
23   changePassword1 = e => {
24     this.setState({ password1: e.target.value });
25   };
26
27   changePassword2 = e => {
28     this.setState({ password2: e.target.value });
29   };
30
31   signup = () => {
32     if (this.state.password1 === this.state.password2) {
33       this.props.signupUser(this.state.email, this.state.
34         password1);
35     } else {
36       this.setState({ error: "Password does not match !" });
37     }
38   };
39
40   render() {
41     return (
42       <div className={css.Signup}>
43         {this.props.userId && <Redirect to='/' />}
44         <h1>Registration form</h1>
45         <div>Please enter your information</div>
46         <input onChange={this.changeEmail} type='email'
47           placeholder='Email' />
48         <input
49           onChange={this.changePassword1}
50             type='password'
51             placeholder='Enter your password'
52           />
53         <input
54           onChange={this.changePassword2}
55             type='password'
56             placeholder='Re-enter your password'
57           />
58         {this.state.error && (
59           <div style={{ color: "red" }}>{this.state.error}</div>
60         )}
61         {this.props.firebaseioError && (
62           <div style={{ color: "red" }}>{this.props.
```



```
61         <div style={{ color: "red" }}>{this.props.  
62             firebaseError}</div>  
63         )}  
64         {this.props.saving && <Spinner />}  
65     <Button text='REGISTER' btnType='Success' clicked={this.  
66             signup} />  
67     </div>  
68 );  
69 }  
70 }  
71  
72 const mapStateToProps = state => {  
73     return {  
74         saving: state.signupReducer.saving,  
75         firebaseError: state.signupReducer.firebaseioError,  
76         userId: state.signupReducer.userId,  
77     };  
78 };  
79  
80 const mapDispatchToProps = dispatch => {  
81     return {  
82         signupUser: (email, password) =>  
83             dispatch(actions.signupUser(email, password)),  
84     };  
85 };  
86  
87 export default connect(mapStateToProps, mapDispatchToProps)  
88 (Signup);
```

Style.module.css:



```
rc > pages > SignupPage > # style.module.css > .Signup input
1  .Signup {
2    width: 100%;
3    border: 1px solid #eee;
4    box-sizing: border-box;
5    box-shadow: 0 2px 3px #ccc;
6    padding: 10px;
7    text-align: center;
8  }
9  .Signup input {
10   display: block;
11   font-size: 18px;
12   padding: 5px;
13   margin: 10px auto;
14 }
```

Components:

BuildControl:

Index.js



PROMINEO TECH

index.js X

```
rc > components > BuildControl > js index.js > ...
1  import React from "react";  6.9k (gzipped: 2.7k)
2
3  import css from "./style.module.css";
4
5  const BuildControl = props =>
6    <div className={css.BuildControl}>
7      <div className={css.Label}>{props.text}</div>
8      <button
9        disabled={props.disabled[props.type]}
10       onClick={() => props.ortsHasah(props.type)}
11       className={css.Less}
12     >
13       Cansel
14     </button>
15     <button onClick={() => props.ortsNemeh(props.type)}
16       className={css.More}>
17       Add
18     </button>
19   </div>
20
21 );
```

21 export default BuildControl;

Style.module.css:



PROMINEO TECH

```
1  .BuildControl {
2      display: flex;
3      justify-content: space-between;
4      align-items: center;
5      margin: 5px 0;
6  }
7
8  .BuildControl button {
9      display: block;
10     font: inherit;
11     padding: 5px;
12     margin: 0 5px;
13     width: 80px;
14     border: 1px solid #aa6817;
15     cursor: pointer;
16     outline: none;
17 }
18
19 .BuildControl button:disabled{
20     background-color: #ac9980;
21     border: 1px solid #7e7365;
22     color: #ccc;
23     cursor: default;
24 }
25
26 .BuildControl button:hover:disabled{
27     background-color: #ac9980;
28     color: #ccc;
29     cursor: not-allowed;
30 }
31 .Label {
32     padding: 10px;
33     font-weight: bold;
34     width: 100px;
35     text-align: right;
36 }
37
38 .BuildControl .Less {
39     background-color: #d39952;
40     color: white;
41 }
42
43 .BuildControl .More {
44     background-color: #8f5e1e;
45     color: white;
46 }
```



PROMINEO TECH

Buildcontrols:

Index.js:



PROMINEO TECH

```
src > components > BuildControls > JS index.js > mapStateToProps > ingredientNames
1 import React from "react"; 6.9k (gzipped: 2.7k)
2 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
3 import * as actions from "../../redux/actions/burgerActions";
4
5 import BuildControl from "../BuildControl";
6 import css from "./style.module.css";
7
8 const BuildControls = props => {
9   const disabledIngredients = { ...props.burgerIngredients };
10
11   for (let key in disabledIngredients) {
12     disabledIngredients[key] = disabledIngredients[key] <= 0;
13   }
14
15   return (
16     <div className={css.BuildControls}>
17       <p>
18         | Burger's price : <strong>{props.price}$</strong>
19       </p>
20
21       {Object.keys(props.ingredientNames).map(el => (
22         <BuildControl
23           | key={el}
24           | ortsHasah={props.ortsHasah}
25           | ortsNemeh={props.ortsNemeh}
26           | disabled={disabledIngredients}
27           | type={el}
28           | text={props.ingredientNames[el]}
29         />
30       ))}
31
32       <button
33         | onClick={props.showConfirmModal}
34         | disabled={!props.purchasing}
35         | className={css.OrderButton}
36         |>
37           | To Order
38         </button>
39     </div>
40   );
41 }
42 const mapStateToProps = state => {
43   return {
44     burgerIngredients: state.burgerReducer.ingredients,
45     price: state.burgerReducer.totalPrice,
46     purchasing: state.burgerReducer.purchasing,
47     ingredientNames: state.burgerReducer.ingredientNames,
48   };
49 }
50
51 const mapDispatchToProps = dispatch => {
52   return {
53     ortsNemeh: ingredientName =>
54       dispatch(actions.addIngredient(ingredientName)),
55     ortsHasah: ingredientName =>
56       dispatch(actions.removeIngredient(ingredientName)),
57   };
58 }
59 export default connect(mapStateToProps, mapDispatchToProps)
  (BuildControls);
```



PROMINEO TECH

Style.module.css:



PROMINEO TECH

```
# style.module.css x
src > components > BuildControls > # style.module.css > @keyframes enable
1  .BuildControls {
2    width: 100%;
3    background-color: #cf8f2e;
4    display: flex;
5    flex-flow: column;
6    align-items: center;
7    box-shadow: 0 2px 1px #ccc;
8    margin: auto;
9    padding: 10px 0;
10 }
11
12 .BuildControls p {
13   font-size: 28px;
14 }
15 }
16
17 .OrderButton {
18   background-color: #dad735;
19   outline: none;
20   cursor: pointer;
21   border: 1px solid #966909;
22   color: #966909;
23   font-family: inherit;
24   font-size: 1.2em;
25   padding: 15px 30px;
26   box-shadow: 2px 2px 2px #966909;
27 }
28 .OrderButton:hover,
29 .OrderButton:active {
30   background-color: #a0db41;
31   border: 1px solid #966909;
32   color: #966909;
33 }
34
35 .OrderButton:disabled {
36   background-color: #c7c6c6;
37   cursor: not-allowed;
38   border: 1px solid #ccc;
39   color: #888888;
40 }
41 .OrderButton:not(:disabled) {
42   animation: enable 0.3s linear;
43 }
44
45 @keyframes enable {
46   0% {
47     transform: scale(1);
48   }
49   60% {
50     transform: scale(1.1);
51   }
52   100% {
53     transform: scale(1);
54   }
55 }
```



PROMINEO TECH

Burger:

Index.js:



```
src > components > Burger > index.js > [e] mapStateToProps > ↗ ingredient
1 import React from "react"; 6.9k (gzipped: 2.7k)
2 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
3 import BurgerIngredient from "../BurgerIngredient";
4 import css from "./style.module.css";
5 import { withRouter } from "react-router-dom"; 14.1k (gzipped:
6
7 const Burger = props => {
8     // {bacon: 2, cheese: 2, meat: 1, salad: 1}
9     const items = Object.entries(props.ingredient);
10
11    let content = [];
12
13    items.map(el => {
14        for (let i = 0; i < el[1]; i++)
15            content.push(<BurgerIngredient key={`${el[0]} ${i + 1}`} type={el[0]} />);
16    });
17    if (content.length === 0)
18        content = <p>Choose an ingredient for a side dish?</p>;
19
20    return (
21        <div className={css.Burger}>
22            <BurgerIngredient type='bread-top' />
23            {content}
24            <BurgerIngredient type='bread-bottom' />
25        </div>
26    );
27 };
28
29 const mapStateToProps = state => {
30     return {
31         ingredient: state.burgerReducer.ingredients,
32     };
33 };
34
35 export default connect(mapStateToProps)(withRouter(Burger));
36
```



PROMINEO TECH

Style.module.css:

```
# style.module.css X  
src > components > Burger > # style.module.css > .Burger  
1  .Burger{  
2    width: 100%;  
3    height: 280px;  
4    margin: auto;  
5    text-align: center;  
6    overflow: scroll;  
7    font-weight: bold;  
8    font-size: 24px;  
9  }  
10  
11  @media (min-width: 800px) and (min-height: 400px) {  
12    .Burger {  
13      width: 700px;  
14      height: 500px;  
15    }  
16  }
```

BurgerIngredient:

Index.js:



```
src > components > BurgerIngredient > JS index.js > [?] BurgerIngredient
1 import React from "react"; 6.9k (gzipped: 2.7k)
2 import css from "./style.module.css";
3
4 const BurgerIngredient = props => {
5   if (props.type === "bread-top")
6     return (
7       <div className={css.BreadTop}>
8         <div className={css.Seed}></div>
9         <div className={`${css.Seed} ${css.Second}`}></div>
10        <div className={`${css.Seed} ${css.Third}`}></div>
11        <div className={`${css.Seed} ${css.Fourth}`}></div>
12     </div>
13   );
14
15   if (props.type === "salad") return <div className={css.Salad}></div>;
16   if (props.type === "bacon") return <div className={css.Bacon}></div>;
17   if (props.type === "meat") return <div className={css.Meat}></div>;
18   if (props.type === "cheese") return <div className={css.Cheese}></div>;
19   if (props.type === "bread-bottom")
20     return <div className={css.BreadBottom}></div>;
21
22   return <div>...</div>;
23 };
24
25 export default BurgerIngredient;
```

Style.module.css



```
1 .BreadTop{
2     width: 80%;
3     height: 20%;
4     margin: auto;
5     background: linear-gradient( #bc581e, #e27b36);
6     border-radius: 50% 50% 0 0;
7     position: relative;
8 }
9 .Seed{
10    position: absolute;
11    width: 10%;
12    height: 15%;
13    background-color: #fff;
14    top: 50%;
15    left: 30%;
16    border-radius: 40%;
17    transform: rotate(-20deg);
18    box-shadow: inset -2px -3px #c9c9c9;
19 }
20 .Second{
21    top: 25%;
22    left: 50%;
23    transform: rotate(20deg);
24 }
25 .Third{
26    top: 35%;
27    left: 78%;
28    transform: rotate(-195deg);
29 }
30 .Fourth{
31    top: 45%;
32    left: 10%;
33    transform: rotate(-170deg);
34 }
35 .Salad{
36    width: 90%;
37    height: 7%;
38    margin: 2% auto;
39    background: linear-gradient( #228c1d, #91ce50);
40    border-radius: 20px;
41 }
42 .Meat{
43    width: 80%;
44    height: 8%;
45    background: linear-gradient( #7f3608, #702e05);
46    margin: 2% auto;
47    border-radius: 14px;
48 }
49 .Cheese{
50    width: 91%;
51    height: 4.5%;
52    background: linear-gradient( #f4d004, #d6bb22);
53    margin: 2% auto;
54    border-radius: 20px;
55 }
56 .Bacon{
57    width: 80%;
58    height: 3%;
59    background: linear-gradient( #bf3813, #c45e38);
60    margin: 2% auto;
61 }
62 }
63 }
```



PROMINEO TECH

```
52
53 }
54 .BreadBottom{
55   width: 80%;
56   height: 10%;
57   margin: auto;
58   background: red;
59   background: linear-gradient(bc581e, e27b36);
60   border-radius: 0 0 20px 20px;
61 }
```

ContactData:

Index.js:



PROMINEO TECH

```
js index.js X
src > components > ContactData > js index.js > mapStateToProps
1 import React from "react"; 6.9k (gzipped: 2.7k)
2
3 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
4 import css from "./style.module.css";
5 import Button from "../General/Button";
6 import Spinner from "../General/Spinner";
7 import axios from "../../axios-orders";
8 import { withRouter } from "react-router-dom"; 14.1k (gzipped: 5
9 import * as actions from "../../redux/actions/orderActions";
10
11 class ContactData extends React.Component {
12   state = {
13     name: null,
14     city: null,
15     street: null,
16     unit: null,
17   };
18
19   changeName = e => {
20     this.setState({ name: e.target.value });
21   };
22   changeCity = e => {
23     this.setState({ city: e.target.value });
24   };
25   changeStreet = e => {
26     this.setState({ street: e.target.value });
27   };
28   changeUnit = e => {
29     this.setState({ unit: e.target.value });
30   };
31
32   componentDidUpdate() {
33     if (
34       this.props.newOrderStatus.finished &&
35       !this.props.newOrderStatus.error
36     ) {
37       this.props.history.replace("/orders");
38     }
39   }
40
41   saveOrder = () => {
42     const newOrder = {
43       userId: this.props.userId,
44       ingredients: this.props.ingredients,
45       total: this.props.price,
46       address: {
47         name: this.state.name,
48         city: this.state.city,
49         street: this.state.street,
50         unit: this.state.unit,
51       },
52     };
53
54     this.props.saveOrderAction(newOrder);
55   };
56   render() {
57     console.log(this.props);
58     return (
59       <div className={css.ContactData}>
60         Price : {this.props.price}
61         <div>
62           {this.props.newOrderStatus.error &&
63             <div>
64               <h3>Error!</h3>
65               {this.props.newOrderStatus.error}
66             </div>
67           }
68       </div>
69     );
70   }
71 }
72
73 export default ContactData;
```



PROMINEO TECH

```
62      {this.props.newOrderStatus.error &&
63       `An error occurred while saving the order : ${this.
64       props.newOrderStatus.error}`}
65   </div>
66   {this.props.newOrderStatus.saving ? (
67    <Spinner />
68   ) : (
69    <div>
70      <input
71        onChange={this.changeName}
72        type='text'
73        name='name'
74        placeholder='Your Name'
75      />
76      <input
77        onChange={this.changeCity}
78        type='text'
79        name='city'
80        placeholder='City'
81      />
82      <input
83        onChange={this.changeStreet}
84        type='text'
85        name='street'
86        placeholder='Street Name'
87      />
88      <input
89        onChange={this.changeUnit}
90        type='text'
91        name='unit'
92        placeholder='Unit'
93      />
94      <Button text='CONFIRM' btnType='Success' clicked=
95        {this.saveOrder} />
96    </div>
97  );
98 }
99 }
100
101 const mapStateToProps = state => {
102   return {
103     price: state.burgerReducer.totalPrice,
104     ingredients: state.burgerReducer.ingredients,
105     newOrderStatus: state.orderReducer.newOrder,
106     userId: state.signupReducer.userId,
107   };
108 };
109
110 const mapDispatchToProps = dispatch => {
111   return {
112     saveOrderAction: newOrder => dispatch(actions.saveOrder
113       (newOrder)),
114   };
115 };
116
117 export default connect(
118   mapStateToProps,
119   mapDispatchToProps
120 )(withRouter(ContactData));
```



PROMINEO TECH

Style.module.css:

```
# style.module.css X  
src > components > ContactData > # style.module.css > ...  
1 .ContactData {  
2   width: 100%;  
3   border: 1px solid #eee;  
4   box-sizing: border-box;  
5   box-shadow: 0 2px 3px #ccc;  
6   padding: 10px;  
7   text-align: center;  
8 }  
9 .ContactData input {  
10  display: block;  
11  font-size: 18px;  
12  padding: 5px;  
13  margin: 10px auto;  
14 }
```

General:

Button:

Index.js



PROMINEO TECH

```
rc > components > General > Button > js index.js > [?] Button
  1 import React from "react"; 6.9k (gzipped: 2.7k)
  2 import css from "./style.module.css";
  3
  4 const Button = props => (
  5   <button
  6     onClick={props.clicked}
  7     className={`${css.Button} ${css[props.btnType]}`}
  8   >
  9     {props.text}{ " " }
 10   </button>
 11 );
 12
 13 export default Button;
```

Style.module.css:



PROMINEO TECH

```
1 .Button {  
2   background-color: transparent;  
3   border: none;  
4   color: white;  
5   outline: none;  
6   cursor: pointer;  
7   font: inherit;  
8   padding: 10px;  
9   margin: 10px;  
10  font-weight: bold;  
11 }  
12 .Button:first-of-type {  
13   margin-left: 0;  
14   padding-left: 0;  
15 }  
16 .Button:disabled {  
17   color: #ccc;  
18   cursor: not-allowed;  
19 }  
20 .Success {  
21   color: #5C9210;  
22 }  
23 .Danger {  
24   color: #944317;  
25 }  
26 }
```

Modal:

Index.js:



```
src > components > General > Modal > js index.js > [modal]
1  import React from "react"; 6.9k (gzipped: 2.7k)
2  import css from "./style.module.css";
3  import Shadow from "../Shadow";
4
5  const modal = props =>
6    <div>
7      <Shadow show={props.show} darahad={props.closeConfirmModal} />
8      <div
9        style={{
10          transform: props.show ? "translateY(0)" : "translateY(-100vh)",
11          opacity: props.show ? "1" : "0",
12        }}
13        className={css.Modal}
14      >
15        {props.children}
16      </div>
17    </div>
18  );
19
20  export default modal;
```

Style.module.css



src > components > General > Modal > # style.module.css > .Modal

```
1  .Modal {  
2    position: fixed;  
3    z-index: 500;  
4    background-color: white;  
5    width: 70%;  
6    border: 1px solid #ccc;  
7    box-shadow: 1px 1px 1px black;  
8    padding: 16px;  
9    left: 15%;  
10   top: 30%;  
11   transition: all 0.3s ease-out;  
12 }  
13  
14 @media (min-width: 600px) {  
15   .Modal {  
16     width: 500px;  
17     left: calc(50% - 250px);  
18   }  
19 }
```

Shadow:

Index.js:

```
JS index.js X  
src > components > General > Shadow > JS index.js > ...  
1  import React from "react"; 6.9k (gzipped: 2.7k)  
2  import css from "./style.module.css";  
3  
4  const Shadow = props => {  
5    return props.show ? (  
6      <div onClick={props.darahad} className={css.Shadow}></div>  
7    ) : null;  
8  };  
9  
10 export default Shadow;
```

Style.module.css



PROMINEO TECH

```
1 .Shadow {  
2   position: fixed;  
3   left: 0;  
4   top: 0;  
5   z-index: 100;  
6   width: 100%;  
7   height: 100%;  
8   background-color: rgba(0, 0, 0, 0.5);  
9 }
```

Spinner:

Index.js:

```
1 import React from "react"; 6.9k (gzipped: 2.7k)  
2  
3 import css from "./style.module.css";  
4  
5 const Spinner = () => <div className={css.Loader}></div>;  
6  
7 export default Spinner;
```

Style.module.css



PROMINEO TECH

```
1 .Loader,
2 .Loader:before,
3 .Loader:after {
4   background: ■#d94b1c;
5   -webkit-animation: load1 1s infinite ease-in-out;
6   animation: load1 1s infinite ease-in-out;
7   width: 1em;
8   height: 4em;
9 }
10 .Loader {
11   color: ■#d94b1c;
12   text-indent: -9999em;
13   margin: 88px auto;
14   position: relative;
15   font-size: 11px;
16   -webkit-transform: translateZ(0);
17   -ms-transform: translateZ(0);
18   transform: translateZ(0);
19   -webkit-animation-delay: -0.16s;
20   animation-delay: -0.16s;
21 }
22 .Loader:before,
23 .Loader:after {
24   position: absolute;
25   top: 0;
26   content: "";
27 }
28 .Loader:before {
29   left: -1.5em;
30   -webkit-animation-delay: -0.32s;
31   animation-delay: -0.32s;
32 }
33 .Loader:after {
34   left: 1.5em;
35 }
36 @-webkit-keyframes load1 {
37   0%,
38   80%,
39   100% {
40     box-shadow: 0 0;
41     height: 4em;
42   }
43   40% {
44     box-shadow: 0 -2em;
45     height: 5em;
46   }
47 }
48 @keyframes load1 {
49   0%,
50   80%,
51   100% {
52     box-shadow: 0 0;
53     height: 4em;
54   }
55   40% {
56     box-shadow: 0 -2em;
57     height: 5em;
58   }
59 }
```



PROMINEO TECH

Hamburger:

Index.js:

A screenshot of a code editor showing the file `index.js`. The code defines a `HamburgerMenu` component using React. It imports React and a CSS module. The component is a div with an onClick handler that toggles a sidebar. Inside the div, there are three nested divs. The code ends with an export statement. The code editor interface shows the file path `src > components > HamburgerMenu > index.js`, the file size `6.9k (gzipped: 2.7k)`, and some navigation icons.

```
js index.js  X
src > components > HamburgerMenu > js index.js > HamburgerMenu
1 import React from "react"; 6.9k (gzipped: 2.7k)
2
3 import css from "./style.module.css";
4
5 const HamburgerMenu = props => (
6   <div onClick={props.toggleSideBar} className={css.
HamburgerMenu}>
7     <div></div>
8     <div></div>
9     <div></div>
10    </div>
11  );
12
13 export default HamburgerMenu;
```

Style.module.css



PROMINEO TECH

style.module.css X

```
src > components > HamburgerMenu > # style.module.css > {} @media (min-width: 600px) > .HamburgerMenu {  
1   .HamburgerMenu {  
2     width: 40px;  
3     height: 100%;  
4     display: flex;  
5     flex-flow: column;  
6     justify-content: space-around;  
7     padding: 10px 0;  
8     box-sizing: border-box;  
9   }  
10  .HamburgerMenu div {  
11    background-color: white;  
12    height: 3px;  
13  }  
14  
15  @media (min-width: 600px) {  
16    .HamburgerMenu {  
17      display: none;  
18    }  
19  }
```

Logo:

Index.js:

js index.js X

```
src > components > Logo > js index.js > Logo  
1  import React from "react"; 6.9k (gzipped: 2.7k)  
2  
3  import logoImage from "../../assets/images/burger-logo.png";  
4  import css from "./style.module.css";  
5  
6  const Logo = () => (  
7    <div className={css.Logo}>  
8      <img src={logoImage}/>  
9    </div>);  
10  
11 export default Logo;
```



PROMINEO TECH

Style.module.css:

```
.Logo {  
    background-color: #rgb(216, 39, 23);  
    padding: 8px;  
    height: 80%;  
    box-sizing: border-box;  
    border-radius: 5px;  
  
}  
.Logo img{  
    height: 100%;  
}
```

Assets.images.burger-logo.png:





PROMINEO TECH

Logout:

Index.js:

```
js index.js  X
src > components > Logout > js index.js > Logout > render
1 import React from "react"; 6.9k (gzipped: 2.7k)
2
3 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
4 import * as actions from "../../redux/actions/signupActions";
5 import { Redirect } from "react-router-dom"; 15.3k (gzipped: 5.8k)
6
7 class Logout extends React.Component {
8   componentDidMount = () => {
9     this.props.logout();
10   }
11
12   render() {
13     return <Redirect to="/" />;
14   }
15 }
16
17 const mapDispatchToProps = dispatch => {
18   return {
19     logout: () => dispatch(actions.logout()),
20   };
21 };
22
23 export default connect(null, mapDispatchToProps)(Logout);
```

Menu:

Index.js



PROMINEO TECH

s index.js X

```
src > components > Menu > js index.js > [x] mapStateToProps
1 import React, { Fragment } from "react"; 6.9k (gzipped: 2.7k)
2 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
3 import css from "./style.module.css";
4 import MenuItem from "../MenuItem";
5
6 const Menu = props => (
7   <div>
8     <ul className={css.Menu}>
9       {props.userId ? (
10         <Fragment>
11           <MenuItem exact link="/">NEW ORDER
12           </MenuItem>
13           <MenuItem link="/orders">ORDERS</MenuItem>
14           <MenuItem link="/logout">LOG OUT</MenuItem>
15         </Fragment>
16       ) : (
17         <Fragment>
18           <MenuItem link="/login">LOG IN</MenuItem>
19           <MenuItem link="/signup">SIGN UP</MenuItem>
20         </Fragment>
21       )}
22     </ul>
23   </div>
24 );
25
26
27 const mapStateToProps = state => {
28   return [
29     {userId: state.signupReducer.userId},
30   ];
31 };
32
33 export default connect(mapStateToProps)(Menu);
```

Style.module.css:



PROMINEO TECH

style.module.css X

```
src > components > Menu > # style.module.css > {} @media (min-width: 500px)
1   .Menu {
2     list-style: none;
3     display: flex;
4     flex-flow: column;
5     margin: 0px;
6     padding: 0px;
7     height: 100%;
8     align-items: center;
9   }
10  @media (min-width: 500px) {
11    .Menu{
12      flex-flow: row ;
13    }
14  }
```

MenuItem:

Index.js:

```
JS index.js X
src > components > MenuItem > JS index.js > ...
1 import React from "react"; 6.9k (gzipped: 2.7k)
2 import { NavLink } from "react-router-dom"; 21.1k (gzipped: 7.7k)
3
4 import css from "./style.module.css";
5
6 const MenuItem = props => (
7   <li className={css.MenuItem}>
8     <NavLink exact={props.exact} activeClassName={css.active} to=
9       {props.link}>
10      {props.children}
11     </NavLink>
12   </li>
13 );
14 export default MenuItem;
```



PROMINEO TECH

Style.module.css:



PROMINEO TECH

```
# style.module.css X

src > components > MenuItem > # style.module.css > .MenuItem a:hover

1  .MenuItem {
2    margin: 0;
3    display: block;
4    width: 100%;
5    align-items: center;
6  }
7  .MenuItem a {
8    color: #8f5c2c;
9    text-decoration: none;
10   height: 100%;
11   display: block;
12   padding: 16px 10px;
13 }
14 .MenuItem a:hover,
15 .MenuItem a:active,
16 .MenuItem a.active {
17   color: color;
18 }
19
20 @media (min-width: 500px) {
21   .MenuItem {
22     margin: 0;
23     display: flex;
24     height: 100%;
25     width: auto;
26     align-items: center;
27   }
28   .MenuItem a {
29     color: white;
30     text-decoration: none;
31     height: 100%;
32     padding: 16px 10px;
33     border-bottom: 4px solid transparent;
34   }
35   .MenuItem a:hover,
36   .MenuItem a:active,
37   .MenuItem a.active {
38     background-color: #8f5c2c;
39     border-bottom: 4px solid #40a4c8;
40   }
41 }
```



Order:

Index.js:

```
rc > components > Order > JS index.js > Order
  1 import React from "react"; 6.9k (gzipped: 2.7k)
  2 import css from "./style.module.css";
  3
  4 const Order = props => {
  5   console.log(props.order);
  6   return (
  7     <div className={css.Order}>
  8       <p>
  9         <strong>Address : </strong>
 10        {props.order.address.name} | {props.order.address.unit}
 11        {props.order.address.street} | {props.order.address.city}
 12      </p>
 13      <p>
 14        <strong>Ingredients : </strong>
 15        Bacon : {props.order.ingredients.bacon} Cheese :
 16        {props.order.ingredients.cheese} Meat : {props.order.
 17          ingredients.meat}
 18        Salad : {props.order.ingredients.salad}
 19      </p>
 20      <p>
 21        <strong>Total amount : {props.order.total} $</strong>
 22      </p>
 23    </div>
 24  );
 25};
 26
 27 export default Order;
```

Style.module.css:



PROMINEO TECH

```
src > components > Order > # style.module.css > .Order
```

```
1  .Order {  
2    width: 80%;  
3    border: 1px solid #eee;  
4    box-shadow: 0 2px 3px #ccc;  
5    padding: 10px;  
6    margin: 10px auto;  
7  }
```

OrderSummary:

Index.js:



```
src > components > OrderSummary > JS index.js > mapStateToProps > price
1 import React from "react"; 6.9k (gzipped: 2.7k)
2 import { connect } from "react-redux"; 11.8k (gzipped: 4.4k)
3
4 import Button from "../General/Button";
5
6 const OrderSummary = props => {
7   return (
8     <div>
9       <h3>Your order</h3>
10      <p>Ingredients of your choice : </p>
11      <ul>
12        <Object.keys(props.ingredients).map(el => (
13          <li key={el}>
14            | {props.ingredientNames[el]} : {props.ingredients[el]}
15            </li>
16          )>
17        </ul>
18
19        <p>
20          | <strong>Order amount: {props.price}$</strong>
21        </p>
22
23        <p>Do you want to continue?</p>
24        <Button clicked={props.onCancel} btnType='Danger'
25          text='CANCEL' />
26        <Button clicked={props.onContinue} btnType='Success'
27          text='CONTINUE' />
28      </div>
29    );
30  };
31
32  const mapStateToProps = state => {
33    return {
34      ingredients: state.burgerReducer.ingredients,
35      ingredientNames: state.burgerReducer.ingredientNames,
36      price: state.burgerReducer.totalPrice,
37    };
38  };
39
40  export default connect(mapStateToProps)(OrderSummary);
```



PROMINEO TECH

SideBar:

Index.js:

```
src > components > SideBar > JS index.js > [ej] SideBar
1  import React from "react";  6.9k (gzipped: 2.7k)
2
3  import css from "./style.module.css";
4  import Logo from "../Logo";
5  import Menu from "../Menu";
6  import Shadow from "../General/Shadow";
7
8  const SideBar = props => {
9    let classes = [css.SideBar, css.Close];
10
11   if (props.showSidebar) {
12     classes = [css.SideBar, css.Open];
13   }
14
15   return (
16     <div>
17       <Shadow show={props.showSidebar} darahad={props.
18         toggleSideBar} />
19
20       <div onClick={props.toggleSideBar} className={classes.join
21         (" ")}>
22         <div className={css.Logo}>
23           <Logo />
24         </div>
25         <Menu />
26       </div>
27     );
28
29   export default SideBar;
```

Style.module.css:



PROMINEO TECH

src > components > SideBar > # style.module.css > ...

```
1  .SideBar {  
2    width: 280px;  
3    background-color: white;  
4    position: fixed;  
5    top: 0;  
6    left: 0;  
7    height: 100%;  
8    z-index: 200;  
9    padding: 32px 16px;  
10   transition: transform 0.3s ease-out;  
11 }  
12  
13 @media (min-width: 500px) {  
14   .SideBar {  
15     display: none;  
16   }  
17 }  
18  
19 .Open {  
20   transform: translateX(0);  
21 }  
22  
23 .Close {  
24   transform: translateX(-100%);  
25 }  
26  
27 .Logo {  
28   height: 15%;  
29 }
```

Toolbar:

Index.js:



PROMINEO TECH

src > components > Toolbar > **index.js** > [x] Toolbar

```
1 import React from "react"; 6.9k (gzipped: 2.7k)
2
3 import css from "./style.module.css";
4 import Logo from "../Logo";
5 import Menu from "../Menu";
6 import HamburgerMenu from "../HamburgerMenu";
7
8 const Toolbar = props => (
9   <header className={css.Toolbar}>
10    <HamburgerMenu toggleSideBar={props.toggleSideBar} />
11    <Logo />
12    <nav className={css.HideOnMobile}>
13      <Menu />
14    </nav>
15  </header>
16);
17
18 export default Toolbar;
```

Style.module.css:



PROMINEO TECH

style.module.css X

```
src > components > Toolbar > # style.module.css > {} @media (max-width: 500px)
1  .Toolbar {
2    height: 56px;
3    width: 100%;
4    position: fixed;
5    top: 0;
6    left: 0;
7    background-color: #703b09;
8    display: flex;
9    justify-content: space-between;
10   align-items: center;
11   padding: 0 20px;
12   box-sizing: border-box;
13   z-index: 90;
14   color: white;
15   font-size: 20px;
16 }
17 @media (max-width: 500px) {
18   .HideOnMobile{
19     display: none;
20   }
21 }
```

Redux:

Actions:

burgerActions:



PROMINEO TECH

js burgerActions.js X

```
src > redux > actions > js burgerActions.js > ...
1  export const addIngredient = ingredientName => {
2    return {
3      type: "ADD_INGREDIENT",
4      ingredientName,
5    };
6  };
7  export const removeIngredient = ingredientName => {
8    return {
9      type: "REMOVE_INGREDIENT",
10     ingredientName,
11   };
12 };
```

LoginActions:



```
s loginActions.js X
src > redux > actions > JS loginActions.js > [x] loginUser > <function>
1  import axios from "axios"; 38.7k (gzipped: 13.3k)
2  import * as actions from "./signupActions";
3
4  export const loginUser = (email, password) => {
5    return function (dispatch) {
6      dispatch(loginUserStart());
7
8      const data = {
9        email,
10       password,
11       returnSecureToken: true,
12     };
13
14     axios
15       .post(
16         "https://identitytoolkit.googleapis.com/v1/
17         accounts:signInWithPassword?
18         key=AIzaSyDsX2K9dJm_HrzmuZvPOM54c08yUn2jAek",
19         data
20       )
21       .then(result => {
22         //Save to local storage
23         const token = result.data.idToken;
24         const userId = result.data.localId;
25         const expiresIn = result.data.expiresIn;
26         const expireDate = new Date(new Date().getTime() +
27           expiresIn * 1000);
28         const refreshToken = result.data.refreshToken;
29
30         localStorage.setItem("token", token);
31         localStorage.setItem("userId", userId);
32         localStorage.setItem("expireDate", expireDate);
33         localStorage.setItem("refreshToken", refreshToken);
34
35         //Refresh token time
36         dispatch(loginUserSuccess(token, userId));
37         dispatch(actions.autoLogoutAfterMillisec(expiresIn *
38           1000));
39       }
40     );
41   };
42
43  export const loginUserStart = () => {
44    return {
45      type: "LOGIN_USER_START",
46    };
47
48  export const loginUserSuccess = (token, userId) => {
49    return {
50      type: "LOGIN_USER_SUCCESS",
51      token,
52      userId,
53    };
54  };
55
56  export const loginUserError = error => {
```



PROMINEO TECH

```
56  export const loginUserError = error => {
57    return {
58      type: "LOGIN_USER_ERROR",
59      error,
60    };
61  };
```

orderActions.js:



PROMINEO TECH

```
s orderActions.js X
src > redux > actions > js orderActions.js > saveOrder > ⊕ <function>
1 import axios from "../../../axios-orders";
2
3 export const loadOrders = userId => {
4   return function (dispatch, getState) {
5     //Orders start (pull)
6     //Next working a spinner
7     dispatch(loadOrdersStart());
8
9     const token = getState().signupReducer.token;
10
11    axios
12      .get(`orders.json?auth=${token}&orderBy="userId"&
13        equalTo="${userId}"`)
14      .then(responce => {
15        const loadedOrders = Object.entries(responce.data).
16          reverse();
17        dispatch(loadOrdersSuccess(loadedOrders));
18      })
19      .catch(err => dispatch(loadOrdersError(err)));
20  };
21
22  export const loadOrdersStart = () => {
23    return {
24      type: "LOAD_ORDERS_START",
25    };
26  };
27
28  export const loadOrdersSuccess = loadedOrders => {
29    return {
30      type: "LOAD_ORDERS_SUCCESS",
31      orders: loadedOrders,
32    };
33  };
34
35  export const loadOrdersError = error => {
36    return {
37      type: "LOAD_ORDERS_ERROR",
38      error,
39    };
40  };
41  //Saving Order
42
43  export const saveOrder = newOrder => {
44    return function (dispatch, getState) {
45      // start Spinner
46      dispatch(saveOrderStart());
47
48      const token = getState().signupReducer.token;
49
50      // saving to FireBase
51      axios
52        .post(`orders.json?auth=${token}`, newOrder)
53        .then(response => {
54          dispatch(saveOrderSuccess());
55        })
56        .catch(error => {
57          dispatch(saveOrderError(error));
58        });
59    };
60  };
61
```



PROMINEO TECH

```
61
62  export const saveOrderStart = () => {
63    return {
64      type: "SAVE_ORDER_START",
65    };
66  };
67
68  export const saveOrderSuccess = () => {
69    return {
70      type: "SAVE_ORDER_SUCCESS",
71    };
72  };
73
74  export const saveOrderError = error => {
75    return {
76      type: "SAVE_ORDER_ERROR",
77      errorMessage: error,
78    };
79  };

```

signupActions.js:



```
js signupActions.js ×
src > redux > actions > JS signupActions.js > [o] autoLogoutAfterMillisec > [o] <function>
1 import axios from "axios"; 38.7k (gzipped: 13.3k)
2 import { loginUserSuccess } from "./loginActions";
3
4 export const signupUser = (email, password) => {
5   return function (dispatch) {
6     dispatch(signupUserStart());
7
8     const data = {
9       email,
10      password,
11      returnSecureToken: true,
12    };
13
14     axios
15       .post(
16         "https://identitytoolkit.googleapis.com/v1/
accounts:signUp?
key=AIzaSyDsX2K9dJm_HrzmuZvPOM54c08yUn2jAek",
17         data
18       )
19       .then(result => {
20         //Save to local storage
21         const token = result.data.idToken;
22         const userId = result.data.localId;
23
24         localStorage.setItem("token", token);
25         localStorage.setItem("userId", userId);
26
27         dispatch(signupUserSuccess(token, userId));
28       })
29       .catch(err => {
30         dispatch(signupUserError(err));
31       });
32     );
33   );
34
35 export const signupUserStart = () => {
36   return {
37     type: "SIGNUP_USER_START",
38   };
39 };
40
41 export const signupUserSuccess = (token, userId) => {
42   return {
43     type: "SIGNUP_USER_SUCCESS",
44     token,
45     userId,
46   };
47 };
48
49 export const signupUserError = error => {
50   return {
51     type: "SIGNUP_USER_ERROR",
52     error,
53   };
54 };
55
56 export const logout = () => {
57   localStorage.removeItem("token");
58   localStorage.removeItem("userId");
59   localStorage.removeItem("expireDate");
60   localStorage.removeItem("refreshToken");
61   return {

```



```
61    return {
62      type: "LOGOUT",
63    };
64  };
65
66  export const autoLogoutAfterMillisec = ms => {
67    return function (dispatch) {
68      // refresh Token time
69      // axios
70      // .post(
71      //   "https://securetoken.googleapis.com/v1/token?
key=AIzaSyDsX2K9dJm_HrzmuvPOM54c08yUn2jAek",
72      //   {
73      //     grant_type: "refresh_token",
74      //     refresh_token: localStorage.getItem("refresh_token"),
75      //   }
76      // )
77      // .then(result => {
78      //   const token = result.data.id_token;
79      //   const userId = result.data.user_id;
80      //   const token = result.data.idToken;
81      //   const userId = result.data.localId;
82      //   const expiresIn = result.data.expiresIn;
83      //   const expireDate = new Date(new Date().getTime() +
expiresIn * 1000);
84      //   const refreshToken = result.data.refreshToken;
85      //   localStorage.setItem("token", token);
86      //   localStorage.setItem("userId", userId);
87      //   localStorage.setItem("expireDate", expireDate);
88      //   localStorage.setItem("refreshToken", refreshToken);
89      //   dispatch(loginUserSuccess(token, userId));
90      // })
91      // .catch(err => {
92      //   dispatch(signupUserError(err));
93      // });
94      //autoLogout
95      // setTimeout(() => {
96      //   dispatch(logout());
97      // }, ms);
98    };
99  };

```



PROMINEO TECH

Reducer:

burgerReducer.js:



PROMINEO TECH

```
1 const initialState = [
2   ingredients: {
3     salad: 0,
4     bacon: 0,
5     cheese: 0,
6     meat: 0,
7   },
8
9   totalPrice: 10,
10  purchasing: false,
11  ingredientNames: {
12    salad: "Salad",
13    bacon: "Bacon",
14    cheese: "Cheese",
15    meat: "Meat",
16  },
17];
18
19 const INGREDIENT_PRICES = { salad: 2, cheese: 3, bacon: 4, meat: 5 };
20
21 const reducer = (state = initialState, action) => {
22   if (action.type === "ADD_INGREDIENT") {
23     return {
24       ...state,
25       ingredients: {
26         ...state.ingredients,
27         [action.ingredientName]: state.ingredients[action.
28           ingredientName] + 1,
29       },
30       totalPrice: state.totalPrice + INGREDIENT_PRICES[action.
31           ingredientName],
32       purchasing: true,
33     };
34   } else if (action.type === "REMOVE_INGREDIENT") {
35     const newPrice =
36       state.totalPrice - INGREDIENT_PRICES[action.
37           ingredientName];
38     return {
39       ...state,
40       ingredients: {
41         ...state.ingredients,
42         [action.ingredientName]: state.ingredients[action.
43           ingredientName] - 1,
44       },
45       totalPrice: newPrice,
46       purchasing: newPrice > 10,
47     };
48   }
49   return state;
50 };
51
52 export default reducer;
```



PROMINEO TECH

orderReducer.js:



PROMINEO TECH

src > redux > reducer > **JS** orderReducer.js > **reducer**

```
1 const initialState = {
2   //Load order
3   orders: [],
4   loading: false,
5   error: null,
6
7   //save order
8
9   newOrder: {
10     saving: false,
11     finished: false,
12     error: null,
13   },
14 };
15 const reducer = (state = initialState, action) => {
16   switch (action.type) {
17     case "LOAD_ORDERS_START":
18       return {
19         ...state,
20         loading: true,
21       };
22     case "LOAD_ORDERS_SUCCESS":
23       return {
24         ...state,
25         loading: false,
26         orders: action.orders,
27       };
28     case "LOAD_ORDERS_ERROR":
29       return {
30         ...state,
31         loading: false,
32         error: action.error,
33       };
34     case "SAVE_ORDER_START":
35       return {
36         ...state,
37         newOrder: {
38           ...state.newOrder,
39           saving: true,
40         },
41       };
42     case "SAVE_ORDER_SUCCESS":
43       return {
44         ...state,
45         newOrder: {
46           ...state.newOrder,
47           saving: false,
48           finished: true,
49           error: null,
50         },
51       };
52     case "SAVE_ORDER_ERROR":
53       return {
54         ...state,
55         newOrder: {
56           ...state.newOrder,
57           saving: false,
58           finished: true,
59           error: action.errorMessage,
60         },
61       };
62     default:
63       return state;
```



PROMINEO TECH

```
63     || return state;
64   }
65 };
66
67 export default reducer;
68
```

signupLoginreducer.js:



```
js signupLoginReducer.js X
src > redux > reducer > js signupLoginReducer.js > [red] reducer > ↗ userId
1  const initialState = {
2    saving: false,
3    loginIn: false,
4    firebaseError: null,
5    firebaseErrorCode: null,
6    token: null,
7    userId: null,
8  };
9
10 const reducer = (state = initialState, action) => {
11   switch (action.type) {
12     case "SIGNUP_USER_START":
13       return {
14         ...state,
15         saving: true,
16       };
17
18     case "SIGNUP_USER_ERROR":
19       return {
20         ...state,
21         saving: false,
22         firebaseError: action.error.response.data.error.message,
23       };
24
25     case "SIGNUP_USER_SUCCESS":
26       return {
27         ...state,
28         saving: false,
29         token: action.token,
30         userId: action.userId,
31       };
32     case "LOGIN_USER_START":
33       return {
34         ...state,
35         loginIn: true,
36       };
37     case "LOGIN_USER_ERROR":
38       return {
39         ...state,
40         loginIn: false,
41         firebaseError: action.error.response.data.error.message,
42         firebaseErrorCode: action.error.response.data.error.code,
43       };
44
45     case "LOGIN_USER_SUCCESS":
46       return {
47         ...state,
48         loginIn: false,
49         token: action.token,
50         userId: action.userId,
51       };
52
53     case "LOGOUT":
54       return {
55         ...state,
56         token: null,
57         userId: null,
58         firebaseError: null,
59         firebaseErrorCode: null,
60       };
61
62     default:
63       return state;
64   }
65 }
```

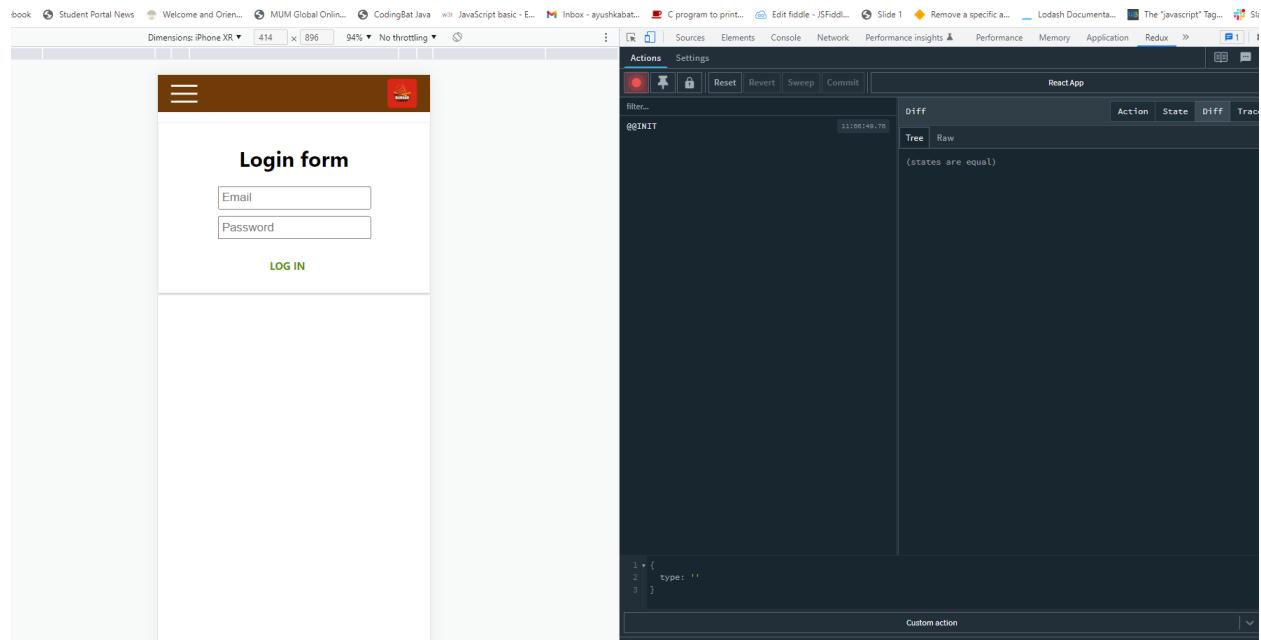


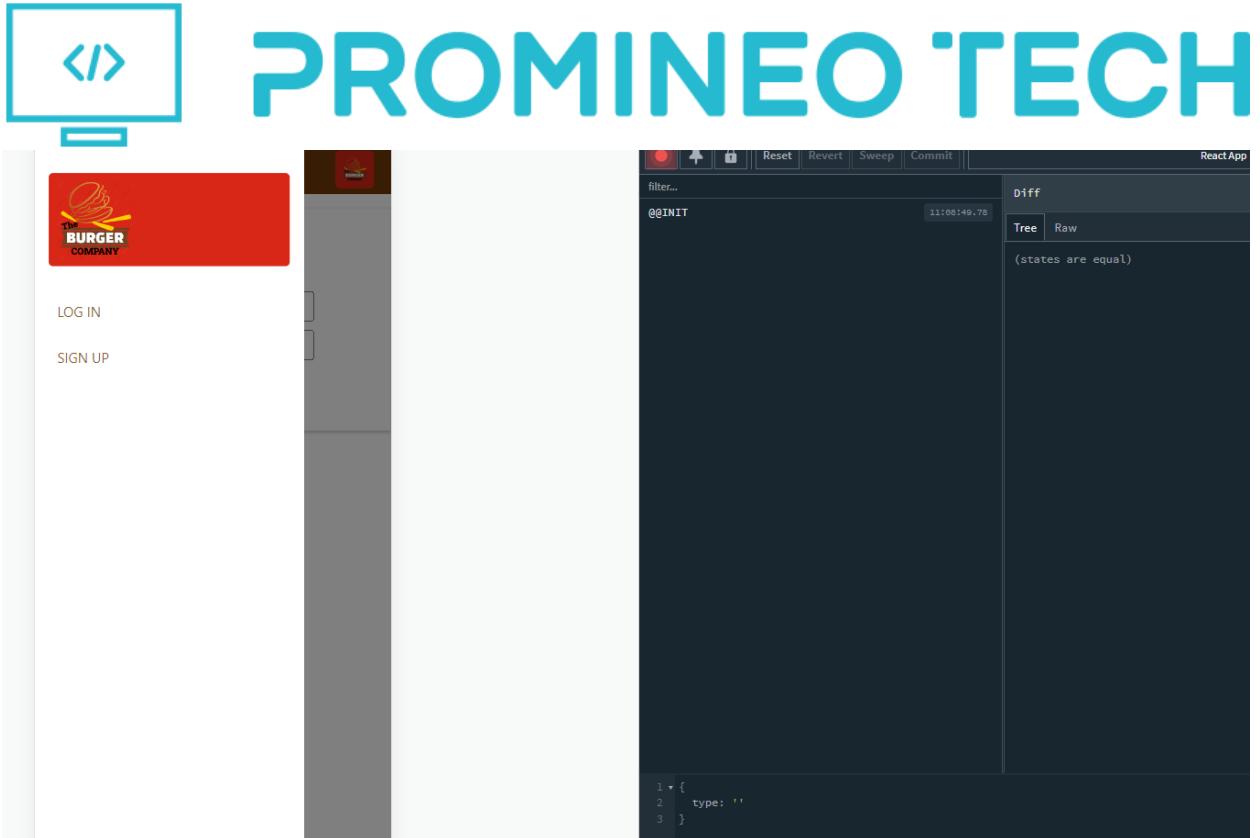
PROMINEO TECH

```
64      }
65  };
66
67  export default reducer;
68
```

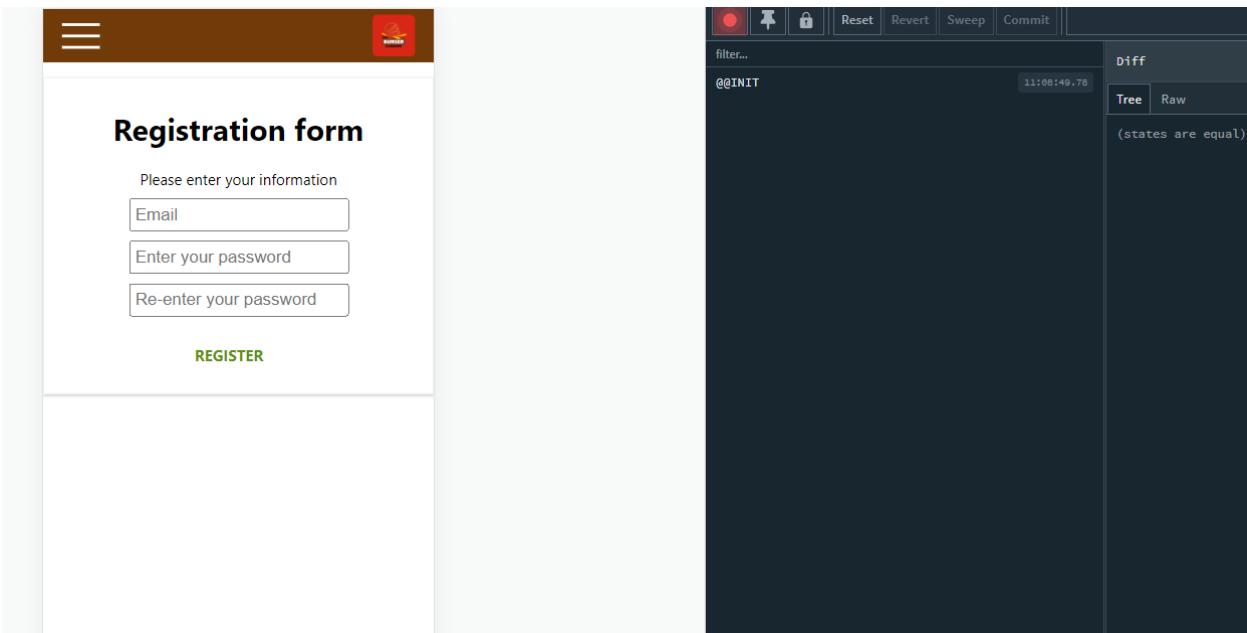
Screenshots of Running Application:

In smart phone:





To register:



To order:



PROMINEO TECH

The screenshot shows a mobile application interface for ordering a burger. At the top, there is a large image of a burger. Below it, the text "Burger's price : 24\$". Underneath, there are four ingredient selection buttons: "Salad" (disabled), "Bacon", "Cheese", and "Meat". Each button has "Cancel" and "Add" options. A large yellow "To Order" button is at the bottom.

Actions Settings

Diff

Tree Raw

burgerReducer (pin)
▶ ingredients (pin): { meat: 1 }
totalPrice (pin): 39 => 24

Action	State
@INIT	11:08:49.78
SIGNUP_USER_START	+00:59.34
SIGNUP_USER_SUCCESS	+00:00.80
ADD_INGREDIENT	+00:14.25
ADD_INGREDIENT	+00:01.23
ADD_INGREDIENT	+00:00.98
ADD_INGREDIENT	+00:01.11

1 + {
2 type: ..
3 }

The screenshot shows the same application after adding all ingredients. The total price is now 24\$. A modal dialog box is displayed with the title "Your order" and the text "Ingredients of your choice: Salad : 1, Bacon : 1, Cheese : 1, Meat : 1". Below this, it says "Order amount: 24\$". At the bottom, it asks "Do you want to continue?" with "CANCEL" and "CONTINUE" buttons.

Actions Settings

Diff

Tree Raw

burgerReducer (pin)
▶ ingredients (pin): { meat: 1 }
totalPrice (pin): 39 => 24

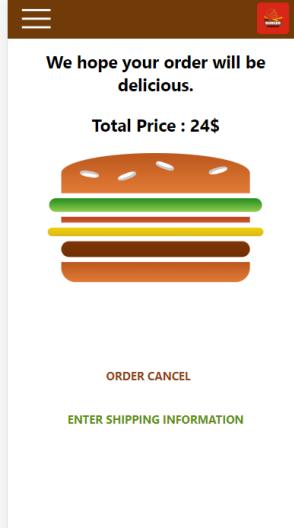
Action	State
@INIT	11:08:49.78
SIGNUP_USER_START	+00:59.34
SIGNUP_USER_SUCCESS	+00:00.80
ADD_INGREDIENT	+00:14.25
ADD_INGREDIENT	+00:01.23
ADD_INGREDIENT	+00:00.98
ADD_INGREDIENT	+00:01.11

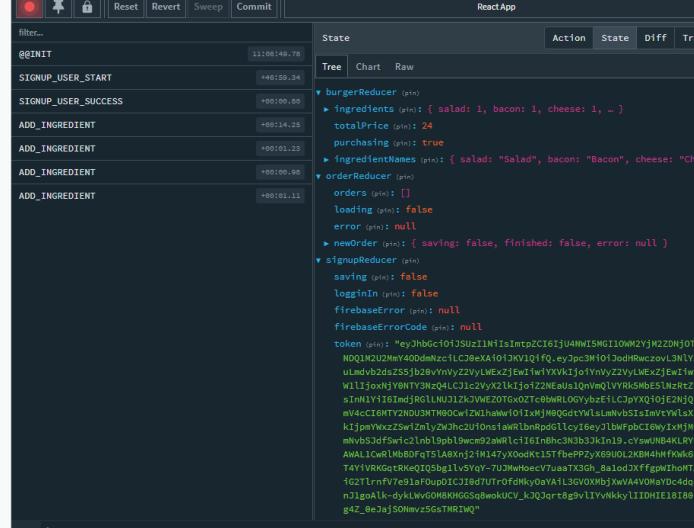
1 + {
2 type: ..
3 }

Contenu:



PROMINEO TECH



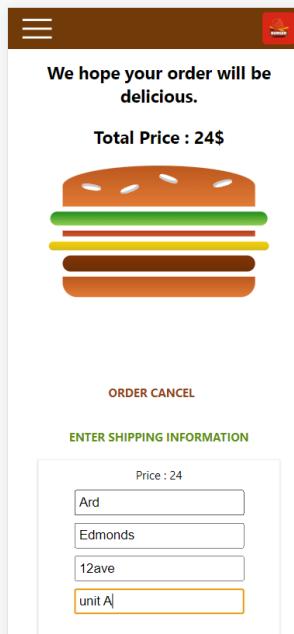


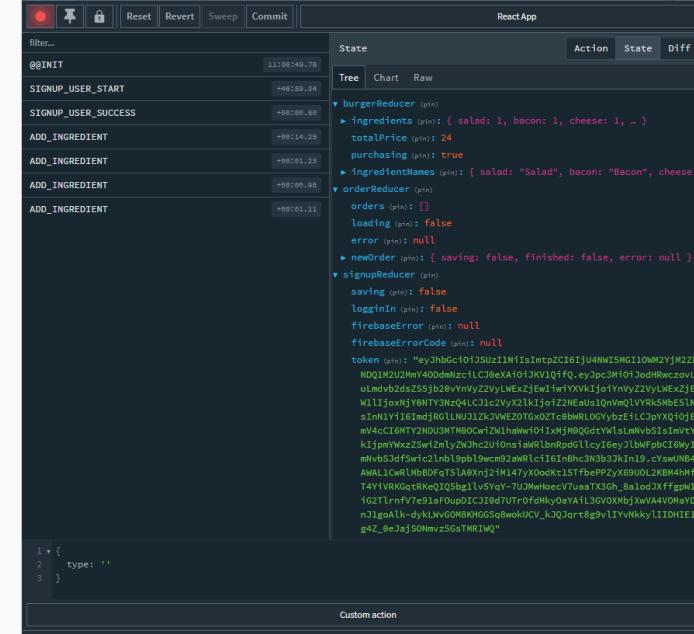
The screenshot shows the React DevTools State tab with the following state structure:

```
@@INIT
SIGNUP_USER_START
SIGNUP_USER_SUCCESS
ADD_INGREDIENT
ADD_INGREDIENT
ADD_INGREDIENT
ADD_INGREDIENT
ADD_INGREDIENT
```

Action column shows actions corresponding to each state entry.

Shipping info:





The screenshot shows the React DevTools State tab with the following state structure:

```
@@INIT
SIGNUP_USER_START
SIGNUP_USER_SUCCESS
ADD_INGREDIENT
ADD_INGREDIENT
ADD_INGREDIENT
ADD_INGREDIENT
ADD_INGREDIENT
ADD_INGREDIENT
```

A new entry 'saving' is present in the Action column.



Address : Ard | unit A | 12ave | Edmonds

Ingredients : Bacon : 1
Cheese : 1 Meat : 1 Salad : 1

Total amount : 24 \$

React App State

```

Actions Settings
@INIT 11:08:49.78
SIGNUP_USER_START +00:59.54
SIGNUP_USER_SUCCESS +00:00.00
ADD_INGREDIENT +00:14.25
ADD_INGREDIENT +00:01.23
ADD_INGREDIENT +00:00.98
ADD_INGREDIENT +00:01.11
SAVE_ORDER_START +00:15.05
SAVE_ORDER_SUCCESS +00:00.73
LOAD_ORDERS_START +00:00.01
LOAD_ORDERS_START +00:00.00
LOAD_ORDERS_SUCCESS +00:00.00
LOAD_ORDERS_SUCCESS +00:00.10

```

Data to firebase:

Realtime Database

```

https://burger-a1f10-default-rtdb.firebaseio.com
- NDBPoQPoChIC1fTVv
- NDB_1z36467_NDoPct9
- NDB1PxXR0YHgX1fW9d
- NDCIntuAvp7x890pDyX
- NDEq_VcXnBmJzaqAE
  address
    city: "Edmonds"
    name: "Ard"
    street: "12ave"
    unit: "unit A"
  ingredients
    bacon: 1
    cheese: 1
    meat: 1
    salad: 1
  total: 24
  userId: "gcDi5BufBUxFNLN74mdk8f2o1"

```

Logout menu:



The screenshot shows a mobile application interface for "The BURGER COMPANY". On the left, there's a red header bar with the company logo. Below it, a navigation menu includes "NEW ORDER", "ORDERS", and "LOG OUT". The main content area displays a "Save" button and some numerical data (1, 1, 5). On the right, the Redux DevTools extension is open, showing the state tree for the "ReactApp" store. The state contains actions like "@INIT", "SIGNUP_USER_START", "SIGNUP_USER_SUCCESS", etc., along with reducer logic for "burgerReducer", "orderReducer", and "signupReducer". The state also includes variables such as "totalPrice", "purchasing", "loading", "error", and user-related data.

Use checking in redux:

This screenshot shows a "Login form" with fields for "Email" and "Password", and a "LOG IN" button. To the right is the Redux DevTools state panel for the "ReactApp" store. The state includes actions like "@INIT", "SIGNUP_USER_START", "SIGNUP_USER_SUCCESS", and various reducers for "burgerReducer", "orderReducer", and "signupReducer". The state also tracks variables such as "loading", "error", and user authentication details like "token" and "userId".

App in web app:

This screenshot shows a "Login form" with fields for "Email" and "Password", and a "LOG IN" button. The browser's developer tools are open, showing the "Redux" tab with the state tree for the "ReactApp" store. The state is identical to the previous screenshots, showing the same actions and reducer logic, including the "burgerReducer" for managing a burger order and the "signupReducer" for user authentication.



LOG IN **SIGN UP**

Registration form

Please enter your information

Email
Enter your password
Re-enter your password

REGISTER

Actions Settings Reset Revert Sweep Commit ReactApp

```
State Action State Diff
Tree Chart Raw
@@INIT 11:00:49.76
SIGNUP_USER_START +@00:00:00
SIGNUP_USER_SUCCESS +@00:00:00
ADD_INGREDIENT +@00:14:28
ADD_INGREDIENT +@00:01:01
ADD_INGREDIENT +@00:00:00
ADD_INGREDIENT +@00:01:11
SAVE_ORDER_START +@00:15:05
SAVE_ORDER_SUCCESS +@00:00:75
LOAD_ORDERS_START +@00:00:00
LOAD_ORDERS_SUCCESS +@00:00:08
LOAD_ORDERS_SUCCESS +@00:00:18
LOGOUT +@00:01:05
LOGOUT +@00:00:00
```

If password error:

LOG IN **SIGN UP**

Login form

ard@gmail.com

EMAIL_NOT_FOUND code is : 400

LOG IN

Actions Settings Reset Revert Sweep Commit ReactApp

```
State Action State Diff
Tree Chart Raw
@@INIT 11:00:49.76
SIGNUP_USER_START +@00:10:34
SIGNUP_USER_SUCCESS +@00:00:00
ADD_INGREDIENT +@00:14:28
ADD_INGREDIENT +@00:01:23
ADD_INGREDIENT +@00:00:00
ADD_INGREDIENT +@00:01:11
SAVE_ORDER_START +@00:15:05
SAVE_ORDER_SUCCESS +@00:00:75
LOAD_ORDERS_START +@00:00:01
LOAD_ORDERS_START +@00:00:00
LOAD_ORDERS_SUCCESS +@00:00:08
LOAD_ORDERS_SUCCESS +@00:00:10
LOGOUT +@00:01:05
LOGOUT +@00:00:00
LOGIN_USER_START +@00:21:96
LOGIN_USER_ERROR +@00:00:11
```

NEW ORDER **ORDERS** **LOG OUT**

Choose an ingredient for a side dish?

Burger's price : 10\$

Salad
 Bacon
 Cheese
 Meat

Actions Settings Reset Revert Sweep Commit ReactApp

```
State Action State Diff
Tree Chart Raw
ADD_INGREDIENT +@00:14:28
ADD_INGREDIENT +@00:01:01
ADD_INGREDIENT +@00:00:00
ADD_INGREDIENT +@00:01:11
SAVE_ORDER_START +@00:15:05
SAVE_ORDER_SUCCESS +@00:00:75
LOAD_ORDERS_START +@00:20:00
LOAD_ORDERS_START +@00:00:00
LOAD_ORDERS_SUCCESS +@00:00:08
LOAD_ORDERS_SUCCESS +@00:00:10
LOGOUT +@00:01:05
LOGOUT +@00:00:00
LOGIN_USER_START +@00:21:96
LOGIN_USER_ERROR +@00:00:11
LOGIN_USER_START +@00:28:06
LOGIN_USER_SUCCESS +@00:00:00
LOGOUT +@00:01:05
REMOVE_INGREDIENT +@00:02:00
REMOVE_INGREDIENT +@00:01:19
REMOVE_INGREDIENT +@00:01:22
REMOVE_INGREDIENT +@00:00:00

1 + {
  2   type: ''
}
3 }
```

Custom action

Inspector Log monitor Chart RTK Query

Console What's New Issues



PROMINEO TECH

NEW ORDER ORDERS LOG OUT

Address : Ard | unit A | 12ave | Edmonds

Ingredients : Bacon : 1 Cheese : 1 Meat : 1 Salad : 1

Total amount : 24 \$

The React DevTools screenshot shows the state tree for the application. The `burgerReducer` contains the `ingredients` object with values: bacon: 1, cheese: 1, meat: 1, salad: 1. The `ordersReducer` contains the `orders` array with one item, which has properties: loading: false, error: null, and `newOrder`: { saving: false, finished: true, error: null }. The `signUpReducer` contains the `signUp` object with properties: saving: false, loginIn: false, and `firebaseError`: { code: 400, message: "EMAIL_NOT_FOUND" }. The `firebaseErrorCode` is also set to 400.

Database in realtime (userId):

Project Overview

Realtime Database

Data Rules Backups Usage

Protect your Realtime Database resources from abuse, such as billing fraud or phishing Configure App Check

https://burger-a1f10-default.firebaseio.com

The Firebase Realtime Database screenshot shows a single node with the following structure:

- NDBPoPouIC1fTVy5
- NDB_1z364G7_NDnPct9
- NDBpXkxR0HgX1fW9d
- NDCIntUaVp7x890pDyX
- NDEq_VcX2n8mJzaqAE
 - address
 - city: "Edmonds"
 - name: "Ard"
 - street: "12ave"
 - unit: "unit A"
 - ingredients
 - bacon: 1
 - cheese: 1
 - meat: 1
 - salad: 1
 - total: 24
- user_id: "gcDIK5BufBUXFNLNe74mdk8f2o1"