Part 5

Fetch Products using Axios

Modify the **Products.js**

```
1.   import axios from "axios";
2.   import { useEffect, useState } from "react";
3.   import "./Products.css";
4.
5.   function Products() {
6.     const [products, setProducts] = useState([]);
7.     useEffect(() => {
8.       axios.get("/products").then((response) => {
9.         console.log(response.data);
10.        setProducts(response.data);
11.      });
12.    }, []);
13.    return (
14.      <div className="Products">
15.        <h1>Products</h1>
16.        <div className="content">
17.          <div>
18.            <button>Add Product</button>
19.          </div>
20.          <div>
21.            <table className="products-table" cellSpacing="0" cellPadding="0">
22.              <thead>
23.                <tr>
24.                  <td>ID</td>
25.                  <td></td>
26.                  <td>Name</td>
27.                  <td>Price</td>
28.                  <td>Size</td>
29.                  <td>Action</td>
30.                </tr>
31.              </thead>
32.              <tbody>
33.                {products !== undefined ? (
34.                  products.map((product, i) => (
35.                    <tr key={i}>
36.                      <td>{product.id}</td>
37.                      <td className="product-thumbnail">
38.                        <img src={`http://localhost/${product.image}`} />
39.                      </td>
40.                      <td>{product.name}</td>
41.                      <td>{product.price}</td>
42.                      <td>{product.size}</td>
43.                      <td>
44.                        <button>Edit</button>
45.                        <button>
46.                          Delete
47.                        </button>
48.                      </td>
49.                    </tr>
50.                  ))
51.                ) : (
52.                  <tr>
53.                    <td colSpan="6">No data available</td>
54.                  </tr>
```

```
55.                    )}
56.               </tbody>
57.             </table>
58.           </div>
59.         </div>
60.       </div>
61.   );
62. }
63.
64. export default Products
```

Modify **Products.css**

```
1.  .products-table {
2.     width: 100%;
3.     border: none;
4.  }
5.  .products-table thead tr {
6.     font-weight: bold;
7.     height: 40px;
8.  }
9.  .products-table tr td {
10.    padding-left: 10px;
11. }
12. .products-table tbody tr:nth-child(odd) {
13.    background-color: lightblue;
14. }
15. .product-thumbnail {
16.    width: 120px;
17.    height: 120px;
18.    overflow: hidden;
19.    padding: 0 10px;
20. }
21. .product-thumbnail img {
22.    display: block;
23.    height: 100px;
24.    width: 100px;
25.    object-fit: cover;
26. }
```

Create a new folder inside **src** and named it as **Contexts.** Inside of Contexts folder, create a new file for **AuthContext.js** and **ProductContext.js.**

AuthContext.js

```
1.  import { createContext, useState } from "react";
2.
3.  export const AuthContext = createContext({
4.    accessToken: "",
5.    authenticate: () => {},
6.    signOut: () => {},
7.    isAuthenticated: false,
8.  });
9.
10. function AuthContextProvider({ children }) {
11.   const [accessToken, setAccessToken] = useState();
12.
13.   function authenticate(token) {
14.     setAccessToken(token);
15.   }
16.
17.   function signOut() {
18.     setAccessToken(null);
19.     localStorage.removeItem("accessToken");
20.   }
21.
22.   const value = {
23.     accessToken: accessToken,
24.     authenticate: authenticate,
25.     isAuthenticated: !!accessToken,
26.     signOut: signOut,
27.   };
28.
29.   return <AuthContext.Provider value={value}>{children}</AuthContext.Provider>;
30. }
31.
32. export default AuthContextProvider;
```

To use the **AuthContext**, modify the following files: **App.js, Login.js** and **Main.js**

App.js

```
1.  import "./App.css";
2.
3.  import { BrowserRouter, Route, Routes, Navigate } from "react-router-dom";
```

```
4.   import Login from "./pages/Login/Login";
5.   import Register from "./pages/Register/Register";
6.   import Main from "./pages/Main/Main";
7.   import Dashboard from "./pages/Main/Dashboard/Dasboard";
8.   import Products from "./pages/Main/Products/Products";
9.
10.  import { useContext, useEffect, useState } from "react";
11.  import AuthContextProvider, { AuthContext } from "./Contexts/AuthContext";
12.
13.  function AuthNavigation() {
14.    return (
15.      <Routes>
16.        <Route exact path="/" element={<Login />} />
17.        <Route exact path="/register" element={<Register />} />
18.        <Route path="*" element={<Navigate replace to="/" />} />
19.      </Routes>
20.    );
21.  }
22.
23.  function AuthenticatedNavigation() {
24.    return (
25.      <Routes>
26.        <Route path="/main" element={<Main />}>
27.          <Route exact path="dashboard" element={<Dashboard />} />
28.          <Route exact path="products" element={<Products />} />
29.        </Route>
30.        <Route path="*" element={<Navigate replace to="/main/dashboard" />} />
31.      </Routes>
32.    );
33.  }
34.
35.  function Navigation() {
36.    const authCtx = useContext(AuthContext);
37.    const accessToken = localStorage.getItem("accessToken");
38.
39.    useEffect(() => {
40.      if (
41.        !authCtx.isAuthenticated &&
42.        (accessToken !== undefined || accessToken !== "" || accessToken !== null)
43.      ) {
44.        authCtx.authenticate(accessToken);
45.      }
46.    }, []);
47.
48.    return (
49.      <BrowserRouter>
50.        {authCtx.isAuthenticated ? (
51.          <AuthenticatedNavigation />
52.        ) : (
53.          <AuthNavigation />
54.        )}
55.      </BrowserRouter>
56.    );
57.  }
58.
59.  function App() {
60.    const handleLogout = () => {
61.      localStorage.removeItem("accessToken");
62.    };
63.
64.    return (
```

```
65.     <div className="App">
66.       <AuthContextProvider>
67.         <Navigation />
68.       </AuthContextProvider>
69.     </div>
70.   );
71. }
72.
73. export default App;
```

Login.js

```
1.   import { useForm } from "react-hook-form";
2.   import "./Login.css";
3.   import axios from "axios";
4.   import { useNavigate } from "react-router-dom";
5.   import { useContext } from "react";
6.   import { AuthContext } from "../../Contexts/AuthContext";
7.
8.   function Login() {
9.     const navigate = useNavigate();
10.    const authCtx = useContext(AuthContext);
11.    const {
12.      register,
13.      handleSubmit,
14.      formState: { errors },
15.    } = useForm({
16.      defaultValues: {
17.        email: "",
18.        password: "",
19.      },
20.    });
21.    const onSubmit = async (data) => {
22.      await axios({
23.        method: "post",
24.        url: "user/login",
25.        data,
26.        headers: { "Access-Control-Allow-Origin": "*" },
27.      })
28.        .then((res) => {
29.          localStorage.setItem("accessToken", res.data.access_token);
30.          authCtx.authenticate(res.data.access_token);
31.          navigate("/main/dashboard");
32.        })
33.        .catch((e) => {
34.          alert(e.response.data.message);
35.        });
36.    };
37.    return (
38.      <div className="Login">
39.        <div className="main-container">
40.          <h3>Login</h3>
41.          <form onSubmit={handleSubmit(onSubmit)}>
42.            <div className="form-container">
43.              <div>
44.                <div className="form-group">
45.                  <label>E-mail:</label>
46.                  <input type="text" {...register("email", { required: true })} />
```

```
47.                </div>
48.                {errors.email && (
49.                   <span className="errors">This field is required</span>
50.                )}
51.              </div>
52.              <div>
53.                <div className="form-group">
54.                  <label>Password:</label>
55.                  <input
56.                    type="password"
57.                    {...register("password", { required: true })}
58.                  />
59.                </div>
60.                {errors.password && (
61.                   <span className="errors">This field is required</span>
62.                )}
63.              </div>
64.              <div className="submit-container">
65.                <button type="submit">Login</button>
66.              </div>
67.              <div className="register-container">
68.                <a href="/register">
69.                   <small>Register</small>
70.                </a>
71.              </div>
72.            </div>
73.          </form>
74.        </div>
75.      </div>
76.    );
77. }
78.
79. export default Login;
```

Modify the **Main.js**

```
1.   import { useEffect } from "react";
2.   import { Link, Outlet, useNavigate } from "react-router-dom";
3.   import "./Main.css";
4.   import { useContext } from "react";
5.   import { AuthContext } from "../../Contexts/AuthContext";
6.
7.   function Main() {
8.     const authCtx = useContext(AuthContext);
9.
10.    const navigate = useNavigate();
11.    const handleLogout = () => {
12.      authCtx.signOut();
13.      navigate("/");
14.    };
15.
16.    useEffect(() => {
17.      if (
18.        authCtx.accessToken === undefined ||
19.        authCtx.accessToken === "" ||
20.        authCtx.accessToken === null
21.      ) {
22.        handleLogout();
23.      }
24.    }, [authCtx.accessToken]);
```

```jsx
25.    return (
26.       <div className="Main">
27.          <div className="container">
28.             <div className="navigation">
29.                <ul>
30.                   <li>
31.                      <Link to="/main/dashboard">Dashboard</Link>
32.                   </li>
33.                   <li>
34.                      <Link to="/main/products">Products</Link>
35.
36.                   </li>
37.                   <li className="logout">
38.                      <a onClick={handleLogout}>Logout</a>
39.                   </li>
40.                </ul>
41.             </div>
42.             <div className="outlet">
43.                <Outlet />
44.             </div>
45.          </div>
46.       </div>
47.    );
48. }
49.
50. export default Main;
```

ProductContext.js (located inside **Context** folder)

```jsx
1.  import { createContext, useState } from "react";
2.
3.  export const ProductContext = createContext({
4.     products: [],
5.     updateProducts: () => {},
6.     addProduct: () => {},
7.     deleteProduct: () => {},
8.  });
9.
10. function ProductContextProvider({ children }) {
11.    const [products, setProducts] = useState([]);
12.
13.    function updateProducts(products) {
14.       setProducts(products);
15.    }
16.
17.    function addProduct(product) {
18.       let currentProducts = products;
19.       currentProducts = [...products, product];
20.       setProducts(currentProducts);
21.    }
22.
23.    function deleteProduct(product) {
24.       //structedClone is javascript function to clone a object
25.       const newSet = structuredClone(products);
26.       const index = products.findIndex((data) => data === product);
27.       //splice is used to remove an element from array
28.       newSet.splice(index, 1);
29.       setProducts(newSet);
30.    }
31.
```

```
32.   const value = {
33.      products: products,
34.      updateProducts: updateProducts,
35.      addProduct: addProduct,
36.      deleteProduct: deleteProduct,
37.   };
38.
39.   return (
40.      <ProductContext.Provider value={value}>{children}</ProductContext.Provider>
41.   );
42. }
43.
44. export default ProductContextProvider;
```

To use the ProductContext, import and put it inside element prop of Route for products.

```
12. import ProductContextProvider from "./Contexts/ProductContext";
```

```
29. <Route
30.      exact
31.      path="products"
32.      element={
33.       <ProductContextProvider>
34.          <Products />
35.       </ProductContextProvider>
36.      }
37. />
```

Modify **Products.js**

```
1.  import axios from "axios";
2.  import { useEffect, useState } from "react";
3.  import "./Products.css";
4.  import { useContext } from "react";
5.  import { ProductContext } from "../../../Contexts/ProductContext";
6.
7.  function Products() {
8.    const [products, setProducts] = useState([]);
9.    const productCtx = useContext(ProductContext);
10.
11.   useEffect(() => {
12.     axios.get("/products").then((response) => {
13.        productCtx.updateProducts(response.data);
14.     });
15.   }, []);
16.
17.   const handleDeleteProduct = (data) => {
18.     productCtx.deleteProduct(data);
19.   };
20.
21.   useEffect(() => {
22.     setProducts(productCtx.products);
23.   }, [productCtx.products]);
24.   return (
```

```jsx
25.      <div className="Products">
26.        <h1>Products</h1>
27.        <div className="content">
28.          <div>
29.            <button>Add Product</button>
30.          </div>
31.          <div>
32.            <table className="products-table" cellSpacing="0" cellPadding="0">
33.              <thead>
34.                <tr>
35.                  <td>ID</td>
36.                  <td></td>
37.                  <td>Name</td>
38.                  <td>Price</td>
39.                  <td>Size</td>
40.                  <td>Action</td>
41.                </tr>
42.              </thead>
43.              <tbody>
44.                {products !== undefined ? (
45.                  products.map((product, i) => (
46.                    <tr key={i}>
47.                      <td>{product.id}</td>
48.                      <td className="product-thumbnail">
49.                        <img src={`http://localhost/${product.image}`} />
50.                      </td>
51.                      <td>{product.name}</td>
52.                      <td>{product.price}</td>
53.                      <td>{product.size}</td>
54.                      <td>
55.                        <button>Edit</button>
56.                        <button onClick={() => handleDeleteProduct(product)}>
57.                          Delete
58.                        </button>
59.                      </td>
60.                    </tr>
61.                  ))
62.                ) : (
63.                  <tr>
64.                    <td colSpan="6">No data available</td>
65.                  </tr>
66.                )}
67.              </tbody>
68.            </table>
69.          </div>
70.        </div>
71.      </div>
72.  );
73. }
74.
75. export default Products;
```

Inside **Products** folder, create a new folder for **ProductForm**, create also **ProductForm.js** and **ProductForm.css** inside the newly folder.

ProductForm.js

```
1.  import "./ProductForm.css";
2.  import { useNavigate, useParams } from "react-router-dom";
3.  import { useEffect, useState } from "react";
4.  import { useForm } from "react-hook-form";
5.  import axios from "axios";
6.  import { useContext } from "react";
7.  import { AuthContext } from "../../../../Contexts/AuthContext";
8.
9.  function ProductForm() {
10.    const authCtx = useContext(AuthContext);
11.
12.    const [selectedFile, setSelectedFile] = useState(undefined);
13.    const [preview, setPreview] = useState(undefined);
14.
15.    const navigate = useNavigate();
16.
17.    let { id } = useParams();
18.    const {
19.      register,
20.      handleSubmit,
21.      formState: { errors },
22.      setValue,
23.    } = useForm({
24.      defaultValues: {
25.        name: "",
26.        description: "",
27.        size: "",
28.        price: "",
29.        file: undefined,
30.      },
31.    });
32.
33.    useEffect(() => {
34.      if (id !== undefined) {
35.        axios.get(`/products/${id}`).then((res) => {
36.          setValue("name", res.data.name);
37.          setValue("description", res.data.description);
38.          setValue("size", res.data.size);
39.          setValue("price", res.data.price);
40.          setValue("is_available", res.data.is_available ? 1 : 0);
41.          setPreview(`http://localhost/php-rest-api-git/${res.data.image}`);
42.        });
43.      }
44.    }, []);
45.
46.    const handleFileSelect = (event) => {
47.      const file = event.target.files[0];
48.      setSelectedFile(file);
49.
50.      const reader = new FileReader();
51.      const url = reader.readAsDataURL(file);
52.
53.      reader.onloadend = () => {
54.        setPreview(reader.result);
55.      };
```

```
56.    };
57.
58.    const onSubmit = async (data) => {
59.      if (id === undefined) {
60.        //create new product
61.        const formData = new FormData();
62.        formData.append("name", data.name);
63.        formData.append("description", data.description);
64.        formData.append("size", data.size);
65.        formData.append("price", data.price);
66.        formData.append("file", selectedFile);
67.        formData.append("is_available", data.is_available);
68.
69.        axios
70.          .post("/products", formData, {
71.            headers: {
72.              "Content-Type": "multipart/form-data",
73.              Authorization: `Bearer ${authCtx.accessToken}`,
74.            },
75.          })
76.          .then((res) => {
77.            if (res) {
78.              alert("Product successfully saved.");
79.              navigate("/main/products");
80.            }
81.          })
82.          .catch((error) => {
83.            alert(error.response.data.message);
84.          });
85.      } else {
86.        //update existing product
87.        axios
88.          .patch(`/products/${id}`, data, {
89.            headers: {
90.              Authorization: `Bearer ${authCtx.accessToken}`,
91.            },
92.          })
93.          .then((res) => {
94.            if (res) {
95.              alert("Product successfully updated.");
96.              navigate("/main/products");
97.            }
98.          })
99.          .catch((error) => {
100.                alert(error.response.data.message);
101.              });
102.      }
103.    };
104.      return (
105.        <div className="ProductForm">
106.          <h1>Product Form</h1>
107.          <div className="content">
108.            <form onSubmit={handleSubmit(onSubmit)}>
109.              <div className="form-container">
110.                <div className="form-group">
111.                  <label>Name:</label>
112.                  <input type="text" {...register("name", { required: true })} />
113.                </div>
114.                {errors.name && (
115.                  <span className="errors">This field is required</span>
116.                )}
```

```
117.                    <div className="form-group">
118.                      <label>Description:</label>
119.                      <textarea
120.                        rows={10}
121.                        {...register("description", { required: true })}
122.                      />
123.                    </div>
124.                    {errors.description && (
125.                      <span className="errors">This field is required</span>
126.                    )}
127.                    <div className="form-group">
128.                      <label>Size:</label>
129.                      <input
130.                        type="number"
131.                        {...register("size", {
132.                          required: true,
133.                          valueAsNumber: true,
134.                          pattern: {
135.                            value: /^[0-9]+$/,
136.                            message: "Please enter a number",
137.                          },
138.                        })}
139.                      />
140.                    </div>
141.                    {errors.size && errors.size.type === "required" && (
142.                      <span className="errors">This field is required</span>
143.                    )}
144.                    {errors.size && errors.size.type === "valueAsNumber" && (
145.                      <span className="errors">This field accepts numbers only</span>
146.                    )}
147.                    <div className="form-group">
148.                      <label>Price:</label>
149.                      <input
150.                        type="number"
151.                        {...register("price", {
152.                          required: true,
153.                          valueAsNumber: true,
154.                          pattern: {
155.                            value: /^[0-9]+$/,
156.                          },
157.                        })}
158.                      />
159.                    </div>
160.                    {errors.price && errors.price.type === "required" && (
161.                      <span className="errors">This field is required</span>
162.                    )}
163.                    {errors.price && errors.price.type === "valueAsNumber" && (
164.                      <span className="errors">This field accepts numbers only</span>
165.                    )}
166.                    <div className="form-group">
167.                      <label>Availability:</label>
168.                      <select {...register("is_available")}>
169.                        <option value="0">Unavailable</option>
170.                        <option value="1">Available</option>
171.                      </select>
172.                    </div>
173.                    {!id ? (
174.                      <input
175.                        type="file"
176.                        {...register("file")}
177.                        onChange={handleFileSelect}
```

```
178.                       />
179.                     ) : null}
180.                   {(selectedFile || preview) && (
181.                     <img className="product-preview" src={preview} />
182.                   )}
183.
184.                   <div className="submit-container">
185.                     <button type="submit">Submit</button>
186.                   </div>
187.                 </div>
188.               </form>
189.             </div>
190.           </div>
191.         );
192.       }
193.     export default ProductForm;
```

ProductForm.css

```
1.   .ProductForm .form-group label {
2.     width: 100px;
3.     margin-right: 8px;
4.     text-align: left;
5.   }
6.
7.   .ProductForm .form-container {
8.     border: 0;
9.     border-radius: 0;
10.    max-width: 50%;
11.    align-items: flex-start;
12.  }
13.
14.  .ProductForm .form-container .form-group input,
15.  .ProductForm .form-container .form-group textarea {
16.    width: 500px;
17.  }
18.
19.  .ProductForm .product-preview {
20.    width: 200px;
21.  }
```

Add the ProductForm to Route of **App.js**

```
1.   import "./App.css";
2.
3.   import { BrowserRouter, Route, Routes, Navigate } from "react-router-dom";
4.   import Login from "./pages/Login/Login";
5.   import Register from "./pages/Register/Register";
6.   import Main from "./pages/Main/Main";
7.   import Dashboard from "./pages/Main/Dashboard/Dasboard";
8.   import Products from "./pages/Main/Products/Products";
9.
10.  import { useContext, useEffect, useState } from "react";
11.  import AuthContextProvider, { AuthContext } from "./Contexts/AuthContext";
12.  import ProductContextProvider from "./Contexts/ProductContext";
13.  import ProductForm from "./pages/Main/Products/ProductForm/ProductForm";
14.
15.  function AuthNavigation() {
16.    return (
```

```
17.     <Routes>
18.       <Route exact path="/" element={<Login />} />
19.       <Route exact path="/register" element={<Register />} />
20.       {/* <Route path="*" element={<Navigate replace to="/" />} /> */}
21.     </Routes>
22.   );
23. }
24.
25. function AuthenticatedNavigation() {
26.   return (
27.     <Routes>
28.       <Route path="/main" element={<Main />}>
29.         <Route exact path="dashboard" element={<Dashboard />} />
30.         <Route
31.           path="products"
32.           element={
33.             <ProductContextProvider>
34.               <Products />
35.             </ProductContextProvider>
36.           }
37.         ></Route>
38.         <Route
39.           path="products/form/:id?"
40.           element={
41.             <ProductContextProvider>
42.               <ProductForm />
43.             </ProductContextProvider>
44.           }
45.         ></Route>
46.       </Route>
47.       <Route path="/" element={<Navigate replace to="/main/dashboard" />} />
48.     </Routes>
49.   );
50. }
51.
52. function Navigation() {
53.   const authCtx = useContext(AuthContext);
54.   const accessToken = localStorage.getItem("accessToken");
55.
56.   useEffect(() => {
57.     if (
58.       !authCtx.isAuthenticated &&
59.       (accessToken !== undefined || accessToken !== "" || accessToken !== null)
60.     ) {
61.       authCtx.authenticate(accessToken);
62.     }
63.   }, []);
64.   return (
65.     <BrowserRouter>
66.       {authCtx.isAuthenticated ? (
67.         <AuthenticatedNavigation />
68.       ) : (
69.         <AuthNavigation />
70.       )}
71.     </BrowserRouter>
72.   );
73. }
74.
75. function App() {
76.   const handleLogout = () => {
77.     localStorage.removeItem("accessToken");
```

```
78.   };
79.
80.   return (
81.     <div className="App">
82.       <AuthContextProvider>
83.         <Navigation />
84.       </AuthContextProvider>
85.     </div>
86.   );
87. }
88.
89. export default App;
```

Modify **Products.js** for create, edit and delete functionality of Product.

```
1.  import axios from "axios";
2.  import { useEffect, useState } from "react";
3.  import "./Products.css";
4.  import { useContext } from "react";
5.  import { ProductContext } from "../../../Contexts/ProductContext";
6.  import { Link } from "react-router-dom";
7.  import { AuthContext } from "../../../Contexts/AuthContext";
8.
9.  function Products() {
10.   const [products, setProducts] = useState([]);
11.   const productCtx = useContext(ProductContext);
12.   const authCtx = useContext(AuthContext);
13.
14.   useEffect(() => {
15.     axios.get("/products").then((response) => {
16.       productCtx.updateProducts(response.data);
17.     });
18.   }, []);
19.
20.   const handleDeleteProduct = (data) => {
21.     axios
22.       .delete(`/products/${data.id}`, {
23.         headers: {
24.           Authorization: `Bearer ${authCtx.accessToken}`,
25.         },
26.       })
27.       .then((res) => {
28.         alert(res.data.message);
29.         productCtx.deleteProduct(data);
30.       })
31.       .catch((error) => {
32.         alert(error.response.data.message);
33.       });
34.   };
35.
36.   useEffect(() => {
37.     setProducts(productCtx.products);
38.   }, [productCtx.products]);
39.   return (
40.     <div className="Products">
41.       <h1>Products</h1>
42.       <div className="content">
43.         <div>
44.           <Link to="/main/products/form">
45.             <button>Add Product</button>
```

```
46.                </Link>
47.             </div>
48.             <div>
49.                <table className="products-table" cellSpacing="0" cellPadding="0">
50.                   <thead>
51.                      <tr>
52.                         <td>ID</td>
53.                         <td></td>
54.                         <td>Name</td>
55.                         <td>Price</td>
56.                         <td>Size</td>
57.                         <td>Action</td>
58.                      </tr>
59.                   </thead>
60.                   <tbody>
61.                      {products !== undefined ? (
62.                        products.map((product, i) => (
63.                           <tr key={i}>
64.                              <td>{product.id}</td>
65.                              <td className="product-thumbnail">
66.                                 <img
67.                                   src={`http://localhost/php-rest-api-git/${product.image}`}
68.                                 />
69.                              </td>
70.                              <td>{product.name}</td>
71.                              <td>{product.price}</td>
72.                              <td>{product.size}</td>
73.                              <td>
74.                                 <Link to={`/main/products/form/${product.id}`}>
75.                                    <button>Edit</button>
76.                                 </Link>
77.                                 <button onClick={() => handleDeleteProduct(product)}>
78.                                    Delete
79.                                 </button>
80.                              </td>
81.                           </tr>
82.                        ))
83.                      ) : (
84.                         <tr>
85.                           <td colSpan="6">No data available</td>
86.                         </tr>
87.                      )}
88.                   </tbody>
89.                </table>
90.             </div>
91.          </div>
92.       </div>
93.    );
94. }
95.
96. export default Products;
```

PHP Back-end Code

https://github.com/lemuelrfrancisco/php-rest-api/tree/part3

Sample user:

Email: test@mail.com

Password: password

ReactJS Front-end Admin Code

https://github.com/lemuelrfrancisco/app-admin/tree/part5

ReactJS Front-end Online shop Code

https://github.com/lemuelrfrancisco/online-shop-sample

For ReactJS Projects

You need to run the `npm install` to download the dependencies and libraries used on project.