Part 5

Fetch Products using Axios

Modify the **Products.js**

```
import axios from "axios";
import { useEffect, useState } from "react";
import "./Products.css";

function Products() {
  const [products, setProducts] = useState([]);
  useEffect(() => {
    axios.get("/products").then((response) => {
      console.log(response.data);
      setProducts(response.data);
    });
  }, []);
  return (
    <div className="Products">
      <h1>Products</h1>
      <div className="content">
        <div>
          <button>Add Product</button>
        </div>
        <div>
          <table className="products-table" cellSpacing="0" cellPadding="0">
            <thead>
              <tr>
                <td>ID</td>
                <td></td>
                <td>Name</td>
                <td>Price</td>
                <td>Size</td>
                <td>Action</td>
              </tr>
            </thead>
            <tbody>
              {products !== undefined ? (
                products.map((product, i) => (
                  <tr key={i}>
                    <td>{product.id}</td>
                    <td className="product-thumbnail">
                      <img src={`http://localhost/${product.image}`} />
                    </td>
                    <td>{product.name}</td>
                    <td>{product.price}</td>
                    <td>{product.size}</td>
                    <td>
                      <button>Edit</button>
                      <button>
                        Delete
                      </button>
                    </td>
                  </tr>
                ))
              ) : (
                <tr>
                  <td colSpan="6">No data available</td>
                </tr>
              )}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  );
}

export default Products
```

Modify **Products.css**

```
.products-table {
  width: 100%;
  border: none;
}
.products-table thead tr {
  font-weight: bold;
  height: 40px;
}
.products-table tr td {
  padding-left: 10px;
}
.products-table tbody tr:nth-child(odd) {
  background-color: lightblue;
}
.product-thumbnail {
  width: 120px;
  height: 120px;
  overflow: hidden;
  padding: 0 10px;
}
.product-thumbnail img {
  display: block;
  height: 100px;
  width: 100px;
  object-fit: cover;
}
```

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

```
import { createContext, useState } from "react";

export const AuthContext = createContext({
  accessToken: "",
  authenticate: () => {},
  signOut: () => {},
  isAuthenticated: false,
});

function AuthContextProvider({ children }) {
  const [accessToken, setAccessToken] = useState();

  function authenticate(token) {
    setAccessToken(token);
  }

  function signOut() {
    setAccessToken(null);
    localStorage.removeItem("accessToken");
  }

  const value = {
    accessToken: accessToken,
    authenticate: authenticate,
    isAuthenticated: !!accessToken,
    signOut: signOut,
  };

  return <AuthContext.Provider value={value}>{children}</AuthContext.Provider>;
}

export default AuthContextProvider;
```

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

App.js

```
import "./App.css";

import { BrowserRouter, Route, Routes, Navigate } from "react-router-dom";
import Login from "./pages/Login/Login";
import Register from "./pages/Register/Register";
import Main from "./pages/Main/Main";
import Dashboard from "./pages/Main/Dashboard/Dasboard";
import Products from "./pages/Main/Products/Products";

import { useContext, useEffect, useState } from "react";
import AuthContextProvider, { AuthContext } from "./Contexts/AuthContext";

function AuthNavigation() {
  return (
    <Routes>
      <Route exact path="/" element={<Login />} />
      <Route exact path="/register" element={<Register />} />
      <Route path="*" element={<Navigate replace to="/" />} />
    </Routes>
  );
}

function AuthenticatedNavigation() {
  return (
    <Routes>
      <Route path="/main" element={<Main />}>
        <Route exact path="dashboard" element={<Dashboard />} />
        <Route exact path="products" element={<Products />} />
      </Route>
      <Route path="*" element={<Navigate replace to="/main/dashboard" />} />
    </Routes>
  );
}

function Navigation() {
  const authCtx = useContext(AuthContext);
  const accessToken = localStorage.getItem("accessToken");

  useEffect(() => {
    if (
      !authCtx.isAuthenticated &&
      (accessToken !== undefined || accessToken !== "" || accessToken !== null)
    ) {
      authCtx.authenticate(accessToken);
    }
  }, []);

  return (
    <BrowserRouter>
      {authCtx.isAuthenticated ? (
        <AuthenticatedNavigation />
      ) : (
        <AuthNavigation />
      )}
    </BrowserRouter>
  );
}

function App() {
  const handleLogout = () => {
    localStorage.removeItem("accessToken");
  };

  return (
    <div className="App">
      <AuthContextProvider>
        <Navigation />
      </AuthContextProvider>
    </div>
  );
}

export default App;
```

Login.js

```
import { useForm } from "react-hook-form";
import "./Login.css";
import axios from "axios";
import { useNavigate } from "react-router-dom";
import { useContext } from "react";
import { AuthContext } from "../../Contexts/AuthContext";

function Login() {
  const navigate = useNavigate();
  const authCtx = useContext(AuthContext);
  const {
    register,
    handleSubmit,
    formState: { errors },
  } = useForm({
    defaultValues: {
      email: "",
      password: "",
    },
  });
  const onSubmit = async (data) => {
    await axios({
      method: "post",
      url: "user/login",
      data,
      headers: { "Access-Control-Allow-Origin": "*" },
    })
      .then((res) => {
        localStorage.setItem("accessToken", res.data.access_token);
        authCtx.authenticate(res.data.access_token);
        navigate("/main/dashboard");
      })
      .catch((e) => {
        alert(e.response.data.message);
      });
  };
  return (
    <div className="Login">
      <div className="main-container">
        <h3>Login</h3>
        <form onSubmit={handleSubmit(onSubmit)}>
          <div className="form-container">
            <div>
              <div className="form-group">
                <label>E-mail:</label>
                <input type="text" {...register("email", { required: true })} />
              </div>
              {errors.email && (
                <span className="errors">This field is required</span>
              )}
            </div>
            <div>
              <div className="form-group">
                <label>Password:</label>
                <input
                  type="password"
                  {...register("password", { required: true })}
                />
              </div>
              {errors.password && (
                <span className="errors">This field is required</span>
              )}
            </div>
            <div className="submit-container">
              <button type="submit">Login</button>
            </div>
            <div className="register-container">
              <a href="/register">
                <small>Register</small>
              </a>
            </div>
          </div>
        </form>
      </div>
    </div>
  );
}

export default Login;
```

Modify the **Main.js**

```
import { useEffect } from "react";
import { Link, Outlet, useNavigate } from "react-router-dom";
import "./Main.css";
import { useContext } from "react";
import { AuthContext } from "../../Contexts/AuthContext";

function Main() {
  const authCtx = useContext(AuthContext);

  const navigate = useNavigate();
  const handleLogout = () => {
    authCtx.signOut();
    navigate("/");
  };

  useEffect(() => {
    if (
      authCtx.accessToken === undefined ||
      authCtx.accessToken === "" ||
      authCtx.accessToken === null
    ) {
      handleLogout();
    }
  }, [authCtx.accessToken]);
  return (
    <div className="Main">
      <div className="container">
        <div className="navigation">
          <ul>
            <li>
              <Link to="/main/dashboard">Dashboard</Link>
            </li>
            <li>
              <Link to="/main/products">Products</Link>

            </li>
            <li className="logout">
              <a onClick={handleLogout}>Logout</a>
            </li>
          </ul>
        </div>
        <div className="outlet">
          <Outlet />
        </div>
      </div>
    </div>
  );
}

export default Main;
```

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

```
import { createContext, useState } from "react";

export const ProductContext = createContext({
  products: [],
  updateProducts: () => {},
  addProduct: () => {},
  deleteProduct: () => {},
});

function ProductContextProvider({ children }) {
  const [products, setProducts] = useState([]);

  function updateProducts(products) {
    setProducts(products);
  }

  function addProduct(product) {
    let currentProducts = products;
    currentProducts = [...products, product];
    setProducts(currentProducts);
  }

  function deleteProduct(product) {
    //structedClone is javascript function to clone a object
    const newSet = structuredClone(products);
    const index = products.findIndex((data) => data === product);
    //splice is used to remove an element from array
    newSet.splice(index, 1);
    setProducts(newSet);
  }

  const value = {
    products: products,
    updateProducts: updateProducts,
    addProduct: addProduct,
    deleteProduct: deleteProduct,
  };

  return (
    <ProductContext.Provider value={value}>{children}</ProductContext.Provider>
  );
}

export default ProductContextProvider;
```

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

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

```
<Route
     exact
     path="products"
     element={
      <ProductContextProvider>
        <Products />
      </ProductContextProvider>
     }
/>
```

Modify **Products.js**

```
import axios from "axios";
import { useEffect, useState } from "react";
import "./Products.css";
import { useContext } from "react";
import { ProductContext } from "../../../Contexts/ProductContext";

function Products() {
  const [products, setProducts] = useState([]);
  const productCtx = useContext(ProductContext);

  useEffect(() => {
    axios.get("/products").then((response) => {
      productCtx.updateProducts(response.data);
    });
  }, []);

  const handleDeleteProduct = (data) => {
    productCtx.deleteProduct(data);
  };

  useEffect(() => {
    setProducts(productCtx.products);
  }, [productCtx.products]);
  return (
    <div className="Products">
      <h1>Products</h1>
      <div className="content">
        <div>
          <button>Add Product</button>
        </div>
        <div>
          <table className="products-table" cellSpacing="0" cellPadding="0">
            <thead>
              <tr>
                <td>ID</td>
                <td></td>
                <td>Name</td>
                <td>Price</td>
                <td>Size</td>
                <td>Action</td>
              </tr>
            </thead>
            <tbody>
              {products !== undefined ? (
                products.map((product, i) => (
                  <tr key={i}>
                    <td>{product.id}</td>
                    <td className="product-thumbnail">
                      <img src={`http://localhost/${product.image}`} />
                    </td>
                    <td>{product.name}</td>
                    <td>{product.price}</td>
                    <td>{product.size}</td>
                    <td>
                      <button>Edit</button>
                      <button onClick={() => handleDeleteProduct(product)}>
                        Delete
                      </button>
                    </td>
                  </tr>
                ))
              ) : (
                <tr>
                  <td colSpan="6">No data available</td>
                </tr>
              )}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  );
}

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

```
import "./ProductForm.css";
import { useNavigate, useParams } from "react-router-dom";
import { useEffect, useState } from "react";
import { useForm } from "react-hook-form";
import axios from "axios";
import { useContext } from "react";
import { AuthContext } from "../../../../Contexts/AuthContext";

function ProductForm() {
  const authCtx = useContext(AuthContext);

  const [selectedFile, setSelectedFile] = useState(undefined);
  const [preview, setPreview] = useState(undefined);

  const navigate = useNavigate();

  let { id } = useParams();
  const {
    register,
    handleSubmit,
    formState: { errors },
    setValue,
  } = useForm({
    defaultValues: {
      name: "",
      description: "",
      size: "",
      price: "",
      file: undefined,
    },
  });

  useEffect(() => {
    if (id !== undefined) {
      axios.get(`/products/${id}`).then((res) => {
        setValue("name", res.data.name);
        setValue("description", res.data.description);
        setValue("size", res.data.size);
        setValue("price", res.data.price);
        setValue("is_available", res.data.is_available ? 1 : 0);
        setPreview(`http://localhost/php-rest-api-git/${res.data.image}`);
      });
    }
  }, []);

  const handleFileSelect = (event) => {
    const file = event.target.files[0];
    setSelectedFile(file);

    const reader = new FileReader();
    const url = reader.readAsDataURL(file);

    reader.onloadend = () => {
      setPreview(reader.result);
    };
  };

  const onSubmit = async (data) => {
    if (id === undefined) {
      //create new product
      const formData = new FormData();
      formData.append("name", data.name);
      formData.append("description", data.description);
      formData.append("size", data.size);
      formData.append("price", data.price);
      formData.append("file", selectedFile);
      formData.append("is_available", data.is_available);

      axios
        .post("/products", formData, {
          headers: {
            "Content-Type": "multipart/form-data",
            Authorization: `Bearer ${authCtx.accessToken}`,
          },
        })
        .then((res) => {
          if (res) {
            alert("Product successfully saved.");
            navigate("/main/products");
          }
        })
        .catch((error) => {
          alert(error.response.data.message);
        });
    } else {
      //update existing product
      axios
        .patch(`/products/${id}`, data, {
          headers: {
            Authorization: `Bearer ${authCtx.accessToken}`,
          },
        })
        .then((res) => {
          if (res) {
            alert("Product successfully updated.");
            navigate("/main/products");
          }
        })
        .catch((error) => {
                 alert(error.response.data.message);
               });
           }
         };
         return (
           <div className="ProductForm">
             <h1>Product Form</h1>
             <div className="content">
               <form onSubmit={handleSubmit(onSubmit)}>
                 <div className="form-container">
                   <div className="form-group">
                     <label>Name:</label>
                     <input type="text" {...register("name", { required: true })} />
                   </div>
                   {errors.name && (
                     <span className="errors">This field is required</span>
                   )}
```

```
                  <div className="form-group">
                    <label>Description:</label>
                    <textarea
                      rows={10}
                      {...register("description", { required: true })}
                    />
                  </div>
                  {errors.description && (
                    <span className="errors">This field is required</span>
                  )}
                  <div className="form-group">
                    <label>Size:</label>
                    <input
                      type="number"
                      {...register("size", {
                        required: true,
                        valueAsNumber: true,
                        pattern: {
                          value: /^[0-9]+$/,
                          message: "Please enter a number",
                        },
                      })}
                    />
                  </div>
                  {errors.size && errors.size.type === "required" && (
                    <span className="errors">This field is required</span>
                  )}
                  {errors.size && errors.size.type === "valueAsNumber" && (
                    <span className="errors">This field accepts numbers only</span>
                  )}
                  <div className="form-group">
                    <label>Price:</label>
                    <input
                      type="number"
                      {...register("price", {
                        required: true,
                        valueAsNumber: true,
                        pattern: {
                          value: /^[0-9]+$/,
                        },
                      })}
                    />
                  </div>
                  {errors.price && errors.price.type === "required" && (
                    <span className="errors">This field is required</span>
                  )}
                  {errors.price && errors.price.type === "valueAsNumber" && (
                    <span className="errors">This field accepts numbers only</span>
                  )}
                  <div className="form-group">
                    <label>Availability:</label>
                    <select {...register("is_available")}>
                      <option value="0">Unavailable</option>
                      <option value="1">Available</option>
                    </select>
                  </div>
                  {!id ? (
                    <input
                      type="file"
                      {...register("file")}
                      onChange={handleFileSelect}
```

```
       />
      ) : null}
      {(selectedFile || preview) && (
        <img className="product-preview" src={preview} />
      )}

      <div className="submit-container">
        <button type="submit">Submit</button>
      </div>
     </div>
    </form>
   </div>
  </div>
 );
}
export default ProductForm;
```

ProductForm.css

```
.ProductForm .form-group label {
  width: 100px;
  margin-right: 8px;
  text-align: left;
}

.ProductForm .form-container {
  border: 0;
  border-radius: 0;
  max-width: 50%;
  align-items: flex-start;
}

.ProductForm .form-container .form-group input,
.ProductForm .form-container .form-group textarea {
  width: 500px;
}

.ProductForm .product-preview {
  width: 200px;
}
```

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

```
import "./App.css";

import { BrowserRouter, Route, Routes, Navigate } from "react-router-dom";
import Login from "./pages/Login/Login";
import Register from "./pages/Register/Register";
import Main from "./pages/Main/Main";
import Dashboard from "./pages/Main/Dashboard/Dasboard";
import Products from "./pages/Main/Products/Products";

import { useContext, useEffect, useState } from "react";
import AuthContextProvider, { AuthContext } from "./Contexts/AuthContext";
import ProductContextProvider from "./Contexts/ProductContext";
import ProductForm from "./pages/Main/Products/ProductForm/ProductForm";

function AuthNavigation() {
  return (
    <Routes>
      <Route exact path="/" element={<Login />} />
      <Route exact path="/register" element={<Register />} />
      {/* <Route path="*" element={<Navigate replace to="/" />} /> */}
    </Routes>
  );
}

function AuthenticatedNavigation() {
  return (
    <Routes>
      <Route path="/main" element={<Main />}>
        <Route exact path="dashboard" element={<Dashboard />} />
        <Route
          path="products"
          element={
            <ProductContextProvider>
              <Products />
            </ProductContextProvider>
          }
        ></Route>
        <Route
          path="products/form/:id?"
          element={
            <ProductContextProvider>
              <ProductForm />
            </ProductContextProvider>
          }
        ></Route>
      </Route>
      <Route path="/" element={<Navigate replace to="/main/dashboard" />} />
    </Routes>
  );
}

function Navigation() {
  const authCtx = useContext(AuthContext);
  const accessToken = localStorage.getItem("accessToken");

  useEffect(() => {
    if (
      !authCtx.isAuthenticated &&
      (accessToken !== undefined || accessToken !== "" || accessToken !== null)
    ) {
      authCtx.authenticate(accessToken);
    }
  }, []);
  return (
    <BrowserRouter>
      {authCtx.isAuthenticated ? (
        <AuthenticatedNavigation />
      ) : (
        <AuthNavigation />
      )}
    </BrowserRouter>
  );
}

function App() {
  const handleLogout = () => {
    localStorage.removeItem("accessToken");
  };

  return (
    <div className="App">
      <AuthContextProvider>
        <Navigation />
      </AuthContextProvider>
    </div>
  );
}

export default App;
```

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

```
import axios from "axios";
import { useEffect, useState } from "react";
import "./Products.css";
import { useContext } from "react";
import { ProductContext } from "../../../Contexts/ProductContext";
import { Link } from "react-router-dom";
import { AuthContext } from "../../../Contexts/AuthContext";

function Products() {
  const [products, setProducts] = useState([]);
  const productCtx = useContext(ProductContext);
  const authCtx = useContext(AuthContext);

  useEffect(() => {
    axios.get("/products").then((response) => {
      productCtx.updateProducts(response.data);
    });
  }, []);

  const handleDeleteProduct = (data) => {
    axios
      .delete(`/products/${data.id}`, {
        headers: {
          Authorization: `Bearer ${authCtx.accessToken}`,
        },
      })
      .then((res) => {
        alert(res.data.message);
        productCtx.deleteProduct(data);
      })
      .catch((error) => {
        alert(error.response.data.message);
      });
  };

  useEffect(() => {
    setProducts(productCtx.products);
  }, [productCtx.products]);
  return (
    <div className="Products">
      <h1>Products</h1>
      <div className="content">
        <div>
          <Link to="/main/products/form">
            <button>Add Product</button>
          </Link>
        </div>
        <div>
          <table className="products-table" cellSpacing="0" cellPadding="0">
            <thead>
              <tr>
                <td>ID</td>
                <td></td>
                <td>Name</td>
                <td>Price</td>
                <td>Size</td>
                <td>Action</td>
              </tr>
            </thead>
            <tbody>
              {products !== undefined ? (
                products.map((product, i) => (
                  <tr key={i}>
                    <td>{product.id}</td>
                    <td className="product-thumbnail">
                      <img
                        src={`http://localhost/php-rest-api-git/${product.image}`}
                      />
                    </td>
                    <td>{product.name}</td>
                    <td>{product.price}</td>
                    <td>{product.size}</td>
                    <td>
                      <Link to={`/main/products/form/${product.id}`}>
                        <button>Edit</button>
                      </Link>
                      <button onClick={() => handleDeleteProduct(product)}>
                        Delete
                      </button>
                    </td>
                  </tr>
                ))
              ) : (
                <tr>
                  <td colSpan="6">No data available</td>
                </tr>
              )}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  );
}

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.