Crud React : (Title , Price) :

CREATE:---

```
import { useState } from "react";
import { useNavigate } from "react-router-dom";

function ProductCreate() {
  const navigate = useNavigate();
  const [title, setTitle] = useState("");
  const [price, setPrice] = useState("");
  const onTitleChange = (e) => {
   setTitle(e.target.value);     };
  const onPriceChange = (e) => {
   setPrice(e.target.value);    };
  const onFormSubmit = (e) => {
   e.preventDefault(); };
   navigate(“/products”)
  return ( _<>  <h1>Nouveau Produit</h1>
    <form onSubmit={onFormSubmit}>
      <input value={title} onChange={onTitleChange} />
<br />
      <input value={price} onChange={onPriceChange}
type="number" /> <br />
      <input type="submit" value="Créer" />  </form></>
)  ;}
export default ProductCreate;
```

UPDATE:---

```
function ProductEdit(prod,onSave){
  const [editProd ,      setEditProd]=useState({...prod});
  const InputChange=(e){
setEditProd({...editProd,
[e.target.name]:e.target.value});},
const HandleSave=()=>{
  onSave(editProd), };
}
  return ( <>
    <h1>Edite Produit</h1>
     <form>
        <input name=”name” , value=”{editProd.name}”
onChange={ InputChange} />
<button> onclick={HandleSave}</button>
      </form>
)  ;}
export default  ProductEdit ;
```

READ & DELETE:----------------------------------

```
function PoductsPage(){
  const [products, setProducts] = useState([]);
  useEffect(() => {
   fetch(" http://localhost:5000/products")
   .then((res) => res.json())
 .then((data) => {setProducts(data);});},[]);

const onDelete = (e) => {
    fetch("http://localhost:5000/products", {
     method: "DELETE",  })
     .then((res) => res.json()) .then((data) => {
      console.log(data); });

return ( <>  <h1>Products Page</h1>
     <table ><thead><tr>
         <th>#</th> <th>Titre</th>
         <th>Prix</th> <th></th></tr> </thead>
      <tbody>   {products.map((p) => (
          <tr key={p.id}>
           <td>{p.id}</td>
           <td>
   <Link to={`/products/${p.id}`}>{p.title}</Link>
           </td>  <td>{p.price}</td><td>
             <button onClick={onDelete}>x</button>
           </td></tr>))}</tbody>
     </table> </>   ;}
export default ProductPage ;
```

AUTH USING REACT-REDUX :

npm i –save react-redux #reduxjs/toolkit

1-Store.js

```
Import {configureStore} from “@reduxjs/tookit”
Export const store =configureStore({
        reducer:{ auth:authReducer}
})
```

2-AuthActions.js :

```
export const LOGIN_SUCCESS = 'LOGIN_SUCCESS';
export const LOGOUT = 'LOGOUT';
export const loginSuccess = () => {
  return {type: LOGIN_SUCCESS,};};
export const logout = () => {
  return {type: LOGOUT,};
};
```

3- authReducer.js:

```
import { LOGIN_SUCCESS, LOGOUT } from
'../actions/authActions';
const initialState = {isAuthenticated: false,};
export const authReducer = (state = initialState, action)
=> { switch (action.type) {
     case LOGIN_SUCCESS: return {  ...state,
     isAuthenticated: true,  };
    case LOGOUT:  return { ...state,
      isAuthenticated: false, };   default:  return state; } };
```

4-Index.js :

```
import {  configureStore } from “@reduxjs/tookit ;
import authReducer from './authReducer';

export const rootReducer =  configureStore ({
  reducer:{ auth:authReducer}
});
```

5-App.js :

```
import React from 'react';
import { useSelector, useDispatch } from 'react-redux';
import { loginSuccess, logout } from
'./redux/actions/authActions';

export function App() {
 const isAuthenticated = useSelector(state =>
state.auth.isAuthenticated);
 const dispatch = useDispatch();

 const handleLogin = () => {
   dispatch(loginSuccess()); };

 const handleLogout = () => {
   dispatch(logout());  };
 return (
   <div> {isAuthenticated ? (
     <button
onClick={handleLogout}>Déconnexion</button>  ) : (
 <button onClick={handleLogin}>Connexion</button> )}
   </div> ); }
```

React Router : App.js :

```
  <BrowserRouter>
    <div className="App">
     <main>  <Routes>
<Route path="/products"element={<ProductsPage   />}
/>
<Route
path="/products/create"element={<ProductCreate
/>} />
       </Routes></main> </div>
   </BrowserRouter>
```

Index.js :

```
const root =
ReactDOM.createRoot(document.getElementById('roo
t'));
root.render(<React.StrictMode>
<App/></React.StrictMode>);
```

CRUD REACT-REDUX :

1-Reducer.js:

```
const initialState = {
   users:[
      { id: 1, name: "Fl", email: "ai@gmail.com" },
      { id: 2, name: "Ha",email: "b@gmail.com" } ]};
export const reducer = (state=initialState, action) => {
   switch(action.type) {  case "Add_user":
         return {...state, users:[...state.users, action.payload]}
        case "Update_user":
         const user =
state.users.find((u)=>u.id===parseInt(action.payload.id))
    if (user) {    user.name =action.payload.name
                    user.email = action.payload.email} return
state
      case "Delete_user":   return {...state, users:
[...state.users.filter((u)=>u.id!==parseInt(action.payload))]}
default:    return state } }
```

2-Actions.js:

```
export const addUserAction = (user) => {
   return {type:"Add_user", payload:user}}
export const updateUserAction = (newuser) => {
   return {type:"Update_user", payload:newuser}}
export const deleteUserAction = (id) => {
   return {type:"Delete_user", payload:id} }
```

3-AddUser:

```
 import { addUserAction } from_"../Config/actions"
 exportfunction AddUser() {
   const count = useSelector(data=>data.users.length);
   const [name, setName] = useState("");
   const [email, setEmail] = useState("");
   const dispatch = useDispatch();
   const navigate = useNavigate()
   const handleClick = () => {
      dispatch(addUserAction({
         id:count+1, name:name,email:email }))  navigate('/') }
   return ( <form> <label>Name</label>
         <input type="text" value={name}
onChange={(e)=>setName(e.target.value)} />
         <label>Email</label>
         <input type="email" value={email}
onChange={(e)=>setEmail(e.target.value)} />
         <button onClick={handleClick}>Enregistrer</button>
</form> )}
```

4-UpUser:

```
export function UpdateUser() {
   const {id} = useParams()    const user =
useSelector(data=>data.users.find((u)=>u.id===parseInt(id))
);
   const [name, setName] = useState(user.name);
   const [email, setEmail] = useState(user.email);
   const dispatch = useDispatch();
   const navigate = useNavigate()
   const handleClick = () => {
      dispatch(updateUserAction({
       id:id,  name:name,email:email })) navigate('/')}
   return (<form> <label>Name</label>
         <input type="text" value={name}
onChange={(e)=>setName(e.target.value)} />
         <label>Email</label>
         <input type="email" value={email}
onChange={(e)=>setEmail(e.target.value)} />
         <button
onClick={handleClick}>Enregistrer</button></form>)}
```

5-ListUser.js:

```
import {deleteUserAction} from '../Config/actions'
export function UserList() {
   const users = useSelector((data)=>data.users);
   const dispatch = useDispatch(); const handleDelete = (id)
=> { dispatch(deleteUserAction(id)) }
   return (<div> <p> <Link to="/add-user"> <button>Add
user</button> </Link>
      </p>  <table><thead>  <tr>
             <td>ID</td>     <td>Name</td> <td>Email</td>
             <td>Action</td>  </tr>  </thead><tbody>
           {users.map((user, index)=> {return (
               <tr key={index}>  <td>{user.id}</td>
<td>{user.name}</td><td>{user.email}</td>  <td>
     <Link to={`/update-user/${user.id}`}>
<button>Edit</button></Link> <button onClick={() =>
handleDelete(user.id)}>Delete</button>
          </td> </tr>  )   }) } </tbody>  </table> </div> ) }
```