Num.js

import React from "react";

import { useSelector, useDispatch } from "react-redux";

import { useState } from "react";

export default function Num() {

  const Tvoitures = useSelector((*data*) => *data*.cars);

  const dispatch = useDispatch();

  const [voitureId, setVoitureId] = useState("");

  const [voitureMarque, setVoitureMarque] = useState("");

  const [voicarbiron, setVoicarbiron] = useState("Essence");

  const [voitureColor, setVoitureColor] = useState("#000000");

  return (

    <center>

      <input

*type*="text"

*placeholder*="id"

*value*={voitureId}

*onChange*={(*e*) => setVoitureId(*e*.target.value)}

      />

      <br />

      <input

*type*="text"

*placeholder*="Marque"

*value*={voitureMarque}

*onChange*={(*e*) => setVoitureMarque(*e*.target.value)}

      />

      <br />

      <label>

        Essence

        <input

*type*="radio"

*name*="carbiron"

*value*="Essence"

*checked*={voicarbiron === "Essence"}

*onChange*={() => setVoicarbiron("Essence")}

        />

        Gasoil

        <input

*type*="radio"

*name*="carbiron"

*value*="Gasoil"

*checked*={voicarbiron === "Gasoil"}

*onChange*={() => setVoicarbiron("Gasoil")}

        />

      </label>

      <br />

      Car Color:

      <input

*type*="color"

*value*={voitureColor}

*onChange*={(*e*) => setVoitureColor(*e*.target.value)}

      />

      <br />

      <br />

      <button

*className*="btn btn-success"

*onClick*={() => {

          dispatch({

            type: "insert",

            payload: {

              id: voitureId,

              marque: voitureMarque,

              carbiron: voicarbiron,

              color: voitureColor,

            },

          });

          setVoitureId("");

          setVoitureMarque("");

          setVoicarbiron("Essence");

          setVoitureColor("#000000");

        }}

      >

        Ajouter

      </button>

      {"  "}

      <button

*className*="btn btn-danger"

*onClick*={() => {

          dispatch({

            type: "UPDATE",

            payload: {

              id: voitureId,

              marque: voitureMarque,

              carbiron: voicarbiron,

              color: voitureColor,

            },

          });

          setVoitureId("");

          setVoitureMarque("");

          setVoicarbiron("Essence"); // Assuming you want to reset to "Essence"

          setVoitureColor("#000000");

        }}

      >

        Modifier

      </button>

      <table *border*="0" *width*="80%">

        <tr>

          <th>ID</th>

          <th>Mark</th>

          <th>Carbiron</th>

          <th>Color</th>

          <th>action</th>

        </tr>

        {Tvoitures.map((*voitu*) => (

          <tr *key*={*voitu*.id}>

            <td>{*voitu*.id}</td>

            <td>{*voitu*.marque}</td>

            <td>{*voitu*.carbiron}</td>

            <td>{*voitu*.color}</td>

            <td>

              <button

*onClick*={() =>

                  dispatch({ type: "supprimer", payload: *voitu*.id })

                }

              >

                Supprimer

              </button>

            </td>

            <td></td>

          </tr>

        ))}

      </table>

    </center>

  );

}

Reducer.js

const initState = {

  cars: [

    { id: 1, marque: "BMW", carbiron: "Gasoil", color: "Red" },

    { id: 2, marque: "GOLF 8", carbiron: "Gasoil", color: "Blue" },

    { id: 3, marque: "AUDIT", carbiron: "Essence", color: "green" },

    { id: 4, marque: "ALPHA ROMIO", carbiron: "Essence", color: "black" },

  ],

};

const reducer = (*state* = initState, *action*) => {

  switch (*action*.type) {

    case "insert":

      return { cars: [...*state*.cars, *action*.payload] };

    case "supprimer":

      return {

        cars: *state*.cars.filter((*v*) => *v*.id !== *action*.payload),

      };

    case "UPDATE":

      return {

        cars: *state*.cars.map((*voitu*) => {

          if (*voitu*.id === *action*.payload.id) {

            return {

              ...*voitu*,

              marque: *action*.payload.marque,

              carbiron: *action*.payload.carbiron,

              color: *action*.payload.color,

            };

          }

          return *voitu*;

        }),

      };

    default:

      return *state*;

  }

};

export default reducer;