import React, { useState } from "react";

import { flightData } from "./airlines.js";

// Parent component - App

function App() {

  const [searchTerm, setSearchTerm] = useState("");

  function onSearchFormChange(event) {

    setSearchTerm(event.target.value);

  }

  return (

    <>

      <h1>Parent component CS385-Lab 3 Airline Search</h1>

      <p>Your current search term is [{searchTerm}]</p>

      <form>

        <h3>Type your search here: </h3>

        <input onChange={onSearchFormChange} type="text" />

      </form>

      <hr />

      <ResultsComponent

        searchTermFromParent={searchTerm}

        airlineArrayFromParent={flightData}

      />

    </>

  );

}

function ResultsComponent(props) {

  function airlineFilterFunction(searchTerm) {

    return function (airlineObject) {

      let flight = airlineObject.flight.toLowerCase();

      let dept = airlineObject.dept.toLowerCase();

      let dest = airlineObject.dest.toLowerCase();

      return (

        searchTerm !== "" &&

        (flight.includes(searchTerm.toLowerCase()) ||

          dept.includes(searchTerm.toLowerCase()) ||

          dest.includes(searchTerm.toLowerCase()))

      );

    };

  }

  let numberResults = props.airlineArrayFromParent.filter(

    airlineFilterFunction(props.searchTermFromParent)

  ).length;

  return (

    <>

      <h1>Child Component: Search Results</h1>

      <h2>There are {numberResults} search results </h2>

      {props.airlineArrayFromParent

        .filter(airlineFilterFunction(props.searchTermFromParent))

        .map((a, index) => (

          <p key={index}>

            <b>{a.flight}</b>, <i>{a.dest}</i> Streams: {a.dept}{" "}

          </p>

        ))}

    </>

  );

} // end of child component for results.

export default App;

import React, { useState } from "react";

import { flightData } from "./airlines.js";

// Parent component - App

function App() {

  const [searchTerm, setSearchTerm] = useState("");

  function onSearchFormChange(event) {

    setSearchTerm(event.target.value);

  }

  return (

    <>

      <h1>Parent component CS385-Lab 3 Airline Search</h1>

      <p>Your current search term is [{searchTerm}]</p>

      <form>

        <h3>Type your search here: </h3>

        <input onChange={onSearchFormChange} type="text" />

      </form>

      <hr />

      <ResultsComponent

        searchTermFromParent={searchTerm}

        airlineArrayFromParent={flightData}

      />

    </>

  );

}

function ResultsComponent(props) {

  function airlineFilterFunction(searchTerm) {

    return function (airlineObject) {

      let flight = airlineObject.flight.toLowerCase();

      let dept = airlineObject.dept.toLowerCase();

      let dest = airlineObject.dest.toLowerCase();

      return (

        searchTerm !== "" &&

        (flight.includes(searchTerm.toLowerCase()) ||

          dept.includes(searchTerm.toLowerCase()) ||

          dest.includes(searchTerm.toLowerCase()))

      );

    };

  }

  let numberResults = props.airlineArrayFromParent.filter(

    airlineFilterFunction(props.searchTermFromParent)

  ).length;

  return (

    <>

      <h1>Child Component: Search Results</h1>

      <h2>There are {numberResults} search results </h2>

      {numberResults === 0 && <p>No flight information available</p>}

      {numberResults === 1 && <p>One flight available</p>}

      {numberResults >= 2 && numberResults <= 20 && <p>Several flights available</p>}

      {numberResults > 20 && <p>A large number of search results - please consider

narrowing your search</p>}

      <button>Clear the result</button>

      {props.airlineArrayFromParent

        .filter(airlineFilterFunction(props.searchTermFromParent))

        .map((a, index) => (

          <p key={index}>

            <b>{a.flight}</b>, <i>{a.dest}</i> Streams: {a.dept}{" "}

          </p>

        ))}

    </>

  );

} // end of child component for results.

export default App;

import React, { useState } from "react";

import { flightData } from "./airlines.js";

// Parent component - App

function App() {

  const [searchTerm, setSearchTerm] = useState("");

  function onSearchFormChange(event) {

    setSearchTerm(event.target.value);

  }

  function clearResults() {

    setSearchTerm("")

  }

  return (

    <>

      <h1>Parent component CS385-Lab 3 Airline Search</h1>

      <p>Your current search term is [{searchTerm}]</p>

      <form>

        <h3>Type your search here: </h3>

        <input onChange={onSearchFormChange} type="text" value={searchTerm}/>

      </form>

      <hr />

      <ResultsComponent

        searchTermFromParent={searchTerm}

        airlineArrayFromParent={flightData}

        clearResultsFromParent = {clearResults}

      />

    </>

  );

}

function ResultsComponent(props) {

  function airlineFilterFunction(searchTerm) {

    return function (airlineObject) {

      let flight = airlineObject.flight.toLowerCase();

      let dept = airlineObject.dept.toLowerCase();

      let dest = airlineObject.dest.toLowerCase();

      return (

        searchTerm !== "" &&

        (flight.includes(searchTerm.toLowerCase()) ||

          dept.includes(searchTerm.toLowerCase()) ||

          dest.includes(searchTerm.toLowerCase()))

      );

    };

  }

  let numberResults = props.airlineArrayFromParent.filter(

    airlineFilterFunction(props.searchTermFromParent)

  ).length;

  return (

    <>

      <h1>Child Component: Search Results</h1>

      <h2>There are {numberResults} search results </h2>

      <button onClick={()=>props.clearResultsFromParent()}>Clear the result</button>

      {numberResults === 0 && <p>No flight information available</p>}

      {numberResults === 1 && <p>One flight available</p>}

      {numberResults >= 2 && numberResults <= 20 && <p>Several flights available</p>}

      {numberResults > 20 && <p>A large number of search results - please consider

narrowing your search</p>}

          <table border = "5">

            <thead>

              <tr>

                <th>Flight No.</th>

                <th>Departure</th>

                <th>Destination</th>

                <th>Status</th>

              </tr>

            </thead>

            <tbody>

              {props.airlineArrayFromParent

                .filter(airlineFilterFunction(props.searchTermFromParent))

                .map((a, index) => (

                  <tr key={index}>

                    <td>{a.flight}</td>

                    <td>{a.dept}</td>

                    <td>{a.dest}</td>

                    <td>{a.status}</td>

                  </tr>

              ))}

            </tbody>

          </table>

    </>

  );

} // end of child component for results.

export default App;