// utils/throttle.js

export function throttle(func, delay) {

let lastCall = 0;

return (...args) => {

const now = Date.now();

if (now - lastCall >= delay) {

func(...args);

lastCall = now;

}

};

}

import React, { useState, useEffect, useCallback } from "react";

import { throttle } from "./utils/throttle";

const ThrottledSearch = () => {

const [query, setQuery] = useState("");

// Create throttled version of the function

const throttledSearch = useCallback(

throttle((value) => {

console.log("Fetching API with query (Throttle):", value);

// fetch(`https://api.example.com/search?q=${value}`);

}, 1000),

[]

);

useEffect(() => {

if (query) throttledSearch(query);

}, [query, throttledSearch]);

return (

<div>

<h2>Throttled Search</h2>

<input

type="text"

placeholder="Type to search..."

onChange={(e) => setQuery(e.target.value)}

/>

</div>

);

};

export default ThrottledSearch;