**Problem Statement**

Read carefully and implement the solution

**React Challenge – "CourseCatalog SPA"**

**1 Scenario**

Your academy's marketing team wants a public preview of its course catalog. They need:

* a homepage that lists courses from a demo API
* the ability to mark favourites
* per‑course details on its own route
* basic theme toggle and pagination
* good perceived speed

They'll replace the API and styling later; you just deliver a solid technical spike.

**2 Feature ↔ Concept Matrix**

| Required Concept | Where to demonstrate it | | --- | --- | | **Vite + JSX & Babel** | Scaffold with **npm create vite@latest course‑catalog -- --template react**. | | **Components, props, VDOM** | **<CourseCard>**, **<Pagination>**, **<FavouriteBadge>** etc. | | **useState** | Track filter text, "dark mode", current page. | | **Axios + APIs** | **GET https://dummyjson.com/products** (or any open JSON source). | | **useEffect + Lifecycles** | Fetch courses on mount & when page changes. | | **Context API** | **ThemeContext** to toggle light/dark across the tree. | | **React Router** | **/** (list) • **/course/:id** (details). | | **useRef + Pagination** | **searchInputRef.current.focus()** after page change; maintain **currentPage** with refs to avoid stale closures if you like. | | **useReducer (advanced state)** | Handle complex favourite‑list actions: **{type:'add', id}** / **{type:'remove', id}**. | | **Redux (+ middleware)** | Optional: migrate the favourites reducer into Redux with **@reduxjs/toolkit**; use **redux‑thunk** to simulate async write to localStorage. | | **Performance optimisation** | *Lazy load* detail page (**React.lazy** + **Suspense**), wrap **CourseCard** in **React.memo**, and **useCallback** handlers to prevent re‑renders. |

**3 Functional Requirements**

1. **Course List Page (/)**
   * Fetch courses (**limit=10&skip=(page‑1)\*10**).
   * Render cards (title, price, thumbnail).
   * Paginate with **<Prev/Next>** (client‑side calculation).
   * Filter by title (case‑insensitive) via a search input.
2. **Course Detail Page (/course/:id)**
   * Lazy‑loaded route component.
   * Fetch single course data; show specs & "Add to Favourites" button.
3. **Favourites Sidebar**
   * Global list via **useReducer** or Redux.
   * Persist to **localStorage**; hydrate on load.
   * Show badge count in header.
4. **Theme Toggle**
   * Button switches CSS class (**light** / **dark**) stored in Context.
   * Remember preference in **localStorage**.
5. **Performance Touches**
   * **React.memo** on **CourseCard**.
   * Code‑split detail page.
   * Use **useRef** for stable scroll‑to‑top on page change.