Awesome! I’ll create a **personal React roadmap checklist** and a **project list** tailored for your level (3 years of experience). Here's a preview of what you’ll get:

**✅ Your Personalized React Developer Checklist (PDF includes this):**

**🔹 Core Concepts**

* JSX, props, and state
* Conditional rendering
* List rendering with keys
* Lifting state up

**🔹 Essential Hooks Mastery**

* useState, useEffect (with cleanup)
* useContext + useReducer
* useMemo, useCallback
* useRef, forwardRef
* Custom hooks

**🔹 Component Architecture**

* Controlled vs uncontrolled inputs
* Compound component pattern
* Reusable and scalable components

**🔹 Performance & Optimization**

* Lazy loading components
* React.memo, useMemo
* Avoiding prop drilling

**🔹 State Management Options**

* Context + useReducer
* Redux Toolkit basics
* Zustand / Jotai familiarity

**🔹 Routing & APIs**

* React Router v6
* Protected Routes
* API abstraction and loading states

**🔹 Testing (Basic to Intermediate)**

* Unit testing with Jest
* Component testing with React Testing Library

**🔹 Bonus (Advanced/Nice-to-Have)**

* React Query / TanStack
* TypeScript with React
* Error boundaries

**🛠️ Projects to Practice These Concepts:**

1. **Task Manager** (React + CRUD + Context)
2. **Theme Switcher App** (Dark/Light mode with useContext)
3. **E-commerce Cart** (Redux or Zustand, Routing, Lazy loading)
4. **Weather Dashboard** (API fetch, Error handling, useEffect, loading states)
5. **Blog App** (Routing, Protected Routes, Role-based auth)
6. **Movie Search App** (Debouncing, Memoization, API, useRef)
7. **Portfolio Website** (React + Tailwind + Animations)
8. **React Quiz App** (useReducer + Custom Hooks)
9. **Todo + Filters + Drag Drop** (useReducer + performance optimization)
10. **React with TypeScript mini app** (for typing practice)

Design a scalable React application with dynamic routing and code-splitting that demonstrates advanced performance optimization techniques. Your implementation should: - Create a multi-page application with at least 3 different routes - Implement lazy loading for each route component using React.lazy() - Configure a centralized Redux store accessible across all components - Handle loading states with Suspense and provide fallback UI - Ensure efficient chunk-based code splitting with meaningful error boundaries Focus on creating a modular architecture that showcases efficient component loading and state management strategies.