Here’s a structured 30-day React.js learning challenge designed to take you from beginner to intermediate level. Each day focuses on building your skills progressively, with coding exercises, small projects, and essential concepts.

**Week 1: React Basics**

**Day 1: Introduction to React**

* **Objective**: Understand what React is and why it’s popular.
* **Tasks**:
  + Install Node.js and npm.
  + Install React using create-react-app.
  + Explore the project structure.
  + Learn JSX syntax.

**Day 2: Components & Props**

* **Objective**: Learn how to create components.
* **Tasks**:
  + Create functional components.
  + Understand the difference between functional and class components.
  + Pass data using props.
  + Build a simple component that accepts props (e.g., user profile card).

**Day 3: State & Hooks (useState)**

* **Objective**: Learn how to manage state in functional components.
* **Tasks**:
  + Learn about the useState hook.
  + Create a simple counter app with useState.

**Day 4: Handling Events**

* **Objective**: Learn how to handle user events.
* **Tasks**:
  + Handle button clicks, form submissions, etc.
  + Modify your counter app to increment and decrement values.

**Day 5: Conditional Rendering**

* **Objective**: Understand conditional rendering in React.
* **Tasks**:
  + Use if, else, ternary operators, and logical &&.
  + Create a simple login/logout component.

**Day 6: Lists & Keys**

* **Objective**: Learn how to render lists and use keys.
* **Tasks**:
  + Render a list of items dynamically.
  + Map over an array and assign unique keys.

**Day 7: Forms & Controlled Components**

* **Objective**: Learn how to work with forms in React.
* **Tasks**:
  + Create a form with inputs (text, select, etc.).
  + Use controlled components to handle form data.

**Week 2: React Core Concepts**

**Day 8: useEffect Hook**

* **Objective**: Learn the basics of side effects in React.
* **Tasks**:
  + Use useEffect for API calls, data fetching, and DOM updates.
  + Build a component that fetches data from a public API (e.g., GitHub users).

**Day 9: React Router Basics**

* **Objective**: Learn how to navigate between different pages.
* **Tasks**:
  + Install and use react-router-dom.
  + Create simple routes with Link and Switch.

**Day 10: Styling in React**

* **Objective**: Learn how to style components in React.
* **Tasks**:
  + Use CSS modules and inline styles.
  + Explore styled-components.

**Day 11: Component Lifecycle**

* **Objective**: Understand component lifecycle methods.
* **Tasks**:
  + Explore component mounting, updating, and unmounting using useEffect.

**Day 12: Context API**

* **Objective**: Learn how to manage global state with Context.
* **Tasks**:
  + Create a theme or user authentication context.
  + Use useContext to access global state in different components.

**Day 13: Handling APIs with Axios**

* **Objective**: Learn how to make API requests.
* **Tasks**:
  + Install Axios.
  + Fetch data from an external API and display it in a component.

**Day 14: Build a Simple Project**

* **Objective**: Build a small project using everything learned in the first two weeks.
* **Tasks**:
  + Create a to-do app or a weather forecast app with API integration.

**Week 3: Intermediate React Concepts**

**Day 15: React Developer Tools**

* **Objective**: Learn how to use React DevTools.
* **Tasks**:
  + Install the React Developer Tools extension.
  + Explore component hierarchies and inspect state and props.

**Day 16: Higher-Order Components (HOCs)**

* **Objective**: Understand what HOCs are.
* **Tasks**:
  + Create a higher-order component for logging prop changes.

**Day 17: useReducer Hook**

* **Objective**: Learn state management with useReducer.
* **Tasks**:
  + Refactor your useState counter app to use useReducer.

**Day 18: React Fragments & Portals**

* **Objective**: Understand fragments and portals.
* **Tasks**:
  + Learn how to avoid unnecessary DOM nodes with <React.Fragment>.
  + Explore portals to render children outside the parent DOM node.

**Day 19: Error Boundaries**

* **Objective**: Handle errors gracefully in React.
* **Tasks**:
  + Create an error boundary component.
  + Catch JavaScript errors anywhere in your component tree.

**Day 20: React Performance Optimization**

* **Objective**: Learn basic optimization techniques.
* **Tasks**:
  + Use React.memo to prevent unnecessary re-renders.
  + Explore useCallback and useMemo.

**Day 21: Lazy Loading Components**

* **Objective**: Learn how to load components only when needed.
* **Tasks**:
  + Implement code-splitting with React.lazy and Suspense.

**Week 4: Advanced Topics & Project Building**

**Day 22: Prop Drilling & Context API Revisited**

* **Objective**: Solve prop drilling problems with the Context API.
* **Tasks**:
  + Refactor an app with deeply nested components to use Context API.

**Day 23: Custom Hooks**

* **Objective**: Learn how to create reusable logic.
* **Tasks**:
  + Create a custom hook for fetching API data.
  + Use this custom hook in multiple components.

**Day 24: TypeScript in React (Optional)**

* **Objective**: Add TypeScript to your React project.
* **Tasks**:
  + Convert a small component or app to use TypeScript.

**Day 25: Testing with Jest & React Testing Library**

* **Objective**: Learn how to write tests for React components.
* **Tasks**:
  + Set up Jest and React Testing Library.
  + Write unit tests for a few components.

**Day 26: Advanced Patterns (Render Props)**

* **Objective**: Learn advanced patterns in React.
* **Tasks**:
  + Implement a render prop pattern for sharing logic between components.

**Day 27: State Management with Redux (Optional)**

* **Objective**: Learn global state management with Redux.
* **Tasks**:
  + Install Redux and set up a basic store.
  + Dispatch actions and manage state across components.

**Day 28: Build a Full Project**

* **Objective**: Combine everything into a final project.
* **Tasks**:
  + Build a full-fledged project such as a task management app or an e-commerce product page with cart functionality.

**Day 29: Deploying React Apps**

* **Objective**: Learn how to deploy your app.
* **Tasks**:
  + Deploy your project to platforms like Vercel or Netlify.

**Day 30: Reflection & Next Steps**

* **Objective**: Review your progress and plan for future learning.
* **Tasks**:
  + Identify areas you need to revisit.
  + Plan to explore advanced topics like React Native, GraphQL, or Next.js.

This plan ensures consistent growth and practice over the 30 days. Let me know if you'd like to modify or add anything specific to your interests!