**1. Understand the Basics of React.js**

* **Core Concepts**: Start with React's basic building blocks:
  + Components (Functional & Class components).
  + JSX (JavaScript XML) syntax.
  + Props and state management.
  + Event handling in React.
* **Resources**:
  + [Official React Docs](https://reactjs.org/docs/getting-started.html)
  + FreeCodeCamp's React tutorials.

**2. Practice With Small Projects**

* Build simple applications like:
  + A to-do list.
  + A counter app.
  + A weather app using an open API.
* This will help reinforce the concepts.

**3. Learn Modern React Features**

* **Hooks**: useState, useEffect, and other built-in hooks.
* **React Router**: For managing navigation and routing.
* **Context API**: For state management without external libraries.

**4. Understand How React Integrates With WordPress**

* **Why React with WordPress?**
  + React is used to build dynamic and interactive features in WordPress (e.g., Gutenberg editor).
* **REST API**:
  + Learn how to fetch data from WordPress using the REST API.
  + Example: Fetch posts or custom post types from a WordPress site.
* **WP-React Integration**:
  + Embed React apps in WordPress using shortcodes or custom themes/plugins.
  + Use the wp\_enqueue\_script function to include React in WordPress.

**5. Advanced Topics for Web Development**

* **State Management**:
  + Learn about Redux or Zustand for complex state management.
* **Styling in React**:
  + CSS modules, Styled-components, or Tailwind CSS.
* **APIs and Asynchronous Code**:
  + Work with APIs using fetch or Axios.

**6. Practice WordPress-Specific Use Cases**

* **Custom Gutenberg Blocks**:
  + Learn how to create custom blocks using @wordpress/scripts and React.
* **Headless WordPress**:
  + Learn how to use WordPress as a backend and React as the frontend.

**7. Explore Tools and Libraries**

* **Build Tools**:
  + Learn how to set up a React project using Vite, CRA (Create React App), or Next.js.
* **Version Control**:
  + Use Git to manage your projects.
* **Testing**:
  + Learn unit testing with Jest or React Testing Library.

**8. Build Real Projects**

* Examples:
  + A WordPress plugin with a React-based dashboard.
  + A front-end React app consuming WordPress REST API.
  + A custom theme with React for interactivity.

**9. Resources and Learning Platforms**

* **Free Resources**:
  + [React Docs](https://react.dev/)
  + [MDN JavaScript Resources](https://developer.mozilla.org/)
* **Paid Courses**:
  + Udemy: "React - The Complete Guide" by Maximilian Schwarzmüller.
  + Codecademy: React.js track.
  + Frontend Masters.

**Daily Practice Plan (2–3 Hours a Day):**

1. **First Week**:
   * 1 hour: React basics (components, JSX, state, and props).
   * 1 hour: Build small projects.
2. **Second Week**:
   * 1 hour: Hooks (useState, useEffect).
   * 1 hour: WordPress REST API basics.
3. **Third Week and Beyond**:
   * 1 hour: Advanced React (Context API, Routing).
   * 1 hour: WordPress + React projects.