ReactJS Full Stack Roadmap

SECTION I

JavaScript Fundamentals through CLI Apps (10-15 Days)

☑ Topics to Cover:	
 Basics: Variables, Data Types, Loops, Conditional Statements. Functions: Writing reusable code, scope, and return values. Arrays and Objects. JSON: Parsing and iterating through JSON data. 	
■ Hands-On Projects:	
 Sales Report generator: Generate a report with the data given in JSON formoder Class Report: Generator a class report with Student performance Marketing Report: Generate a marketing report with summarized campaign data 	
★ Key Advice for Mastery	
Focus on programmatic thinking. Using JSON makes your logic closer to real-world do handling—think like a developer early on.	ıta
SECTION II	
HTML & CSS Basics (5-8 Days)	
 ☐ HTML: Basic structure, forms, lists, images, and links. ☐ CSS: Margin, padding, box model and selectors. ☐ Multi Page site structure and linking pages. 	
Hands-On Projects:	
☐ Build multi-page portfolios	
☐ Build a multi page portfolio to showcase your work.	
☐ Websites of famous personalities: Like A.P.J. Abdul Kalam, Sachin Tendulkar.	



Avoid diving too deep into design. Stick to creating clean, structured websites with proper linking and styling.

SECTION III

Vanilla JavaScrip	ot + API	Calls	(15-20)	Days)
-------------------	----------	-------	---------	-------

	Topics to Cover:
	 DOM Manipulation: Select, modify, create, and delete elements dynamically. Event handling and debugging with Chrome DevTools. Fetch API: GET and POST requests.
	Hands-On Projects:
	 Employee Management System: CRUD functionality with local or API data. Player Manager: Fetch player data from a sports API and manage it (filter, sort). Movie Database: Build a front-end to fetch and display movies with filters.
❖	- Key Advice for Mastery
	aster Chrome DevTools for debugging—it's a superpower for developers. Combine API lls and DOM manipulation for dynamic apps.
SI	ECTION IV
Ge	etting Ready for React (ES6 + Functional Programming) (7-10 Days)
	Topics to Cover:
	 ES6: let/const, arrow functions, spread/rest operators, destructuring, and template literals.
	☐ Functional Programming: map, reduce and filter.
	Hands-On Projects:
	☐ Total of a Cart: Calculate the total using reduce.

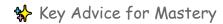


Practice functional programming with real-world examples. Build confidence in transforming and filtering data.

SECTION V

Introduction to Read	ct (4 W	eeks)
----------------------	---------	-------

Ø	Topics to Cover:
	 □ JSX: Writing HTML-like syntax in JavaScript. □ React Router: Navigation and multi-page apps. □ State Management: useState for managing dynamic data. □ Fetch API: Integrate API calls with React.
	Hands-On Projects:
	 To-Do App: Build a multi-page application for task management. Zomato Clone: Basic restaurant listing with product cart, filters, and navigation. Simple E-Commerce App: Features include add-to-cart functionality, filtering, mapping, and product search.
❖	Key Advice for Mastery
	e React for practical, real-world projects. Start small and incrementally scale nplexity.
SE	ECTION VI
Ge	t Started with Backend (1-1.5 Months)
Ø	Topics to Cover:
	 □ REST APIs: Build APIs with Express.js. □ Database: MongoDB or MySQL basics. □ CRUD Operations: Endpoints for GET, POST, PUT, DELETE.
	Hands-On Projects:
	$\hfill\square$ Restaurant Management API: Manage menus, orders, and customers efficiently.



Focus on clean and simple REST APIs. Start with one database—don't complicate things early.

SECTION VII

Backend Integration (2-3 Weeks)

Topics to Cover: Integrate React frontend with backend APIs. Fetch and display API data.
Hands-On Projects: Meetup App: Includes event search, event details page, and filtering functionality.
 Key Advice for Mastery
ster the frontend-backend connection. Building full-stack features boosts your fidence as a developer.
CTION VIII dux Toolkit (2 Weeks)
Topics to Cover:
 State Management using Redux Toolkit. Actions, reducers, and stores.
Hands-On Projects:
 School Management App: A comprehensive app to manage student data, track attendance, and record grades for streamlined academic management. Finance Management App: An intuitive app to manage your finances, track expenses, monitor budgets, and plan savings efficiently.

★ Key Advice for Mastery

Redux improves app scalability. Learn it through simple projects before adding complexity.

SECTION IX

React Advanced Topics (2 Weeks)

Topics to Cover:	
☐ Hooks: useReducer, useContext, useMemo, and custom hooks.	
■ Hands-On Projects:	
 React Project Upgrades Enhance existing React projects by incorporating custom hooks and useReducer f better state management and code reusability. 	or
★ Key Advice for Mastery	
Custom hooks simplify React code. Learn when and how to create them effectively.	
SECTION X	
Advanced CSS (2 Weeks)	
Topics to Cover:	
☐ Flexbox and Grid layout.	
☐ Responsive design and UI components.	
■ Hands-On Projects:	
 Blog Layout: Design a responsive blog layout using grids and flexbox for modern w design. 	/eb
 UI Component Practice: Focus on building small, reusable UI components like card modals, and forms. 	ls,
★ Key Advice for Mastery	

Advanced CSS is about layout precision. Learn to build clean, responsive components.

SECTION XI

JS Concepts for Interviews (2 Weeks)
Topics to Cover:
☐ Core Concepts: call, bind, apply, closures, promises, hoisting, event loop.
■ Hands-On Projects:
 Practical Task: Record live coding videos to explain each concept and demonstrate problem-solving techniques in real-time.
★ Key Advice for Mastery
Explain JavaScript concepts clearly—this is crucial for interviews and will deepen your knowledge.
SECTION XII
Full-Stack Plus (Advanced Features) 2-3 Weeks
Topics to Cover:
☐ WebSockets: Real-time apps using Socket.IO.☐ Authentication: JWT and OAuth (Google/GitHub).
■ Hands-On Projects:
Chat App: Build a chat application with real-time messaging and user authentication for secure communication.
★ Key Advice for Mastery
Learn to handle real-world challenges like real-time updates and secure logins.
SECTION XIII
Frontend Plus (2 Weeks)
Topics to Cover:
☐ TypeScript: Types, interfaces, and applying TypeScript to projects.

	☐ Testing: Basics of unit testing with Jest.
	☐ Performance: Debugging and optimization techniques.
	☐ Build Tools: Webpack and build processes.
<u> </u>	Hands-On Projects:
	☐ Enhancement: Apply TypeScript for type safety and integrate unit tests into existing projects to ensure reliability and maintainability.
☆	Key Advice for Mastery
• •	eScript and testing give you an edge in interviews. Use these to write reliable, tainable code.
Con	clusion (8-10 Months)
Ø (Outcome:
	A strong portfolio with diverse projects: CLI apps, frontend, backend, integrations, and advanced full-stack apps.
	Expertise in real-world concepts: APIs, state management, authentication, andTypeScript.
繋	Key Advice for Mastery

Stay consistent. Each section builds on the last—don't skip steps. Focus on projects to learn by doing, and explain concepts clearly to master them.