Admissions & Support:
support@mockattempt.com
Phone / WhatsApp: +91 98765 43210
Course Code: REAC • **Updated:** 01 Nov 2026
https://mockattempt.com/courses/reactjs-training

REAC • FRONTEND & REACT ECOSYSTEM **BEGINNER TO JOB-READY REACT.JS ENGINEER**

100% LIVE INDUSTRY EXPERT COHORT + INTERNSHIP LETTER

Reactjs Training

Master modern React.js from Virtual DOM Fiber reconciliation, JSX, Props/State & Hooks (useState, useEffect, useRef, forwardRef) to Portals, React Router, Code Splitting, Context API, Redux Toolkit & Vitest — with runnable JSX components, 10×50 MCQ Mock Tests per topic, and Spoken Voice + Gemini AI Interviews at ₹3,999.

Duration: 16–18 Weeks • Live + Hands-On Projects (212 Hrs) • **Modules:** 16 • **Live Topics:** 53 • **Labs & Tasks:** 106 • **Mocks / AI Vivas:** 386 / 386 • **Projects:** 16 Module + 1 Live Client Prod

Fixed Non-Negotiable Tuition Fee: Base Fee **₹3,999** + 18% GST (₹720) = **Total Incl. GST: ₹4,719**

Live Batch Tracks: Weekday Live Online Batch (Mon – Fri) (Monday to Friday (5 Live Interactive Classes / Week) • 1.5 Hours / Day (60 Mins Live Architecture & Coding + 30 Mins Hands-On Lab) • 18–20 Weeks Fast-Track) | Weekend Intensive Live Online Batch (Sat – Sun) (Saturday & Sunday (Designed for Working Professionals & College Students) • 3.0 Hours / Day (2 Hrs Deep-Dive Live Coding + 1 Hr Guided Project Build) • 20–22 Weeks Weekend Track)

Core Technologies & Engineering Stack: React 19 • JSX & Fiber Virtual DOM • React Hooks • React Router • Tailwind CSS • Context & Redux Toolkit • React Testing Library

1. PROGRAM MASTERY OUTCOMES & ELIGIBILITY

End-to-End Engineering Outcomes:

1. Understand React Fiber reconciliation, Virtual DOM diffing, keys, pure functional components, and immutable state updates
2. Master React Hooks (useState, useEffect cleanup, useRef, forwardRef, useContext, custom hooks) without stale-closure or infinite-loop bugs
3. Architect scalable SPAs with React Router, React.lazy Suspense code splitting, Error Boundaries, Portals, Tailwind CSS, and Redux Toolkit
4. Clear 350 Timed Mock Exams (17,500 React.js MCQs) and 350 Spoken Voice + Gemini AI Technical Interviews
5. Feature-sliced component architecture with custom Hooks, compound components, and reusable accessible UI primitives.
6. Clean separation between server cache state (REST/async fetching, optimistic updates) and local/global UI state.
7. Protected routing, role-based guards, schema-validated forms, and resilient Error Boundaries with skeleton states.
8. Render profiling via React DevTools, route-level lazy code splitting (`Suspense`), and list virtualization.

Eligibility & Prerequisites: Basic familiarity with HTML, CSS, and ES6 JavaScript (arrow functions, map/filter, destructuring) • Any laptop/PC running Node.js 20+, Vite, Chrome React DevTools, and VS Code • No prior Reactjs industry experience required — Module 1 starts from environment setup, core mental models, and plain-English fundamentals. • Basic computer & internet familiarity on a Windows, macOS, or Linux laptop (8 GB+ RAM recommended for smooth live coding labs). • Ideal for B.Tech / BCA / MCA / B.Sc IT students, non-CS beginners, career switchers, and working developers upgrading to 2026 production standards. • Commitment to attend live Weekday or Weekend classes, complete daily hands-on lab checklists, and practice topic-wise Mock Tests & AI Interviews.

2. DETAILED MODULE & TOPIC-WISE SYLLABUS (16 MODULES • 53 LIVE TOPICS)

MODULE 1 OF 16 • 12 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 1: React.js Architecture, Fiber Virtual DOM, Vite Tooling & JSX Internals

Understand declarative UIs, React Fiber Virtual DOM reconciliation, Vite ES-module HMR setup, `<StrictMode>`, and how JSX compiles to `_jsx()` element descriptors.

Objectives: ✓ Explain Virtual DOM diffing & React Fiber Render vs Commit phases • ✓ Bootstrap fast modern React 19 apps with Vite & <StrictMode> • ✓ Write XSS-safe JSX expressions using className, htmlFor, and camelCase props

1.1 React Introduction — Declarative UIs (`UI = f(state)`) & Component-Based Architecture [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Introduction — Declarative UIs (`UI = f(state)`) & Component-Based Architecture in real Reactjs systems — specifically: Imperative DOM mutat...

1.2 React Features — Virtual DOM Diffing, React Fiber & Unidirectional Data Flow [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Features — Virtual DOM Diffing, React Fiber & Unidirectional Data Flow in real Reactjs systems — specifically: How React compares previous a...

1.3 Modern React Setup — Vite, ES Modules, HMR & `<StrictMode>` Double-Invocation Checks [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Modern React Setup — Vite, ES Modules, HMR & `<StrictMode>` Double-Invocation Checks in real Reactjs systems — specifically: Setting up React 19 w...

1.4 React JSX Deep Dive — Compilation (`_jsx`), Expressions `{}` & XSS Auto-Escaping [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React JSX Deep Dive — Compilation (`_jsx`), Expressions `{}` & XSS Auto-Escaping in real Reactjs systems — specifically: Why JSX is syntactic suga...

Module 1 Hands-On Capstone Project (100 Marks): Vite + React 19 Developer Profile & Course Catalog Shell — Bootstrap a clean Vite + React 19 application under `<StrictMode>` rendering a modular Academy Header, Hero Stats Banner, and JSX Course Highlight Cards with computed discount badges.

MODULE 2 OF 16 • 13 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 2: Components, Immutable Props, Local State & Lifting State Up

Design pure PascalCase functional components, pass read-only `props` & `children`, manage reactive component state, and coordinate siblings by lifting state up.

Objectives: ✓ Always name React components in PascalCase so JSX distinguishes them from HTML tags • ✓ Treat props as strictly read-only immutable inputs and leverage `children` composition • ✓ Coordinate sibling components cleanly by lifting shared state to their closest common parent

2.1 React Functional vs Class Components — Purity, PascalCase & Composition [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Functional vs Class Components — Purity, PascalCase & Composition in real Reactjs systems — specifically: Pure functional components vs lega...

2.2 React Props — Destructuring Defaults, Prop Forwarding & Read-Only Immutability [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Props — Destructuring Defaults, Prop Forwarding & Read-Only Immutability in real Reactjs systems — specifically: Passing primitives, objects...

2.3 React State — Snapshot Semantics, Immutable Array/Object Updates & Batching [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React State — Snapshot Semantics, Immutable Array/Object Updates & Batching in real Reactjs systems — specifically: Why local `let` variables don’...

2.4 State vs Props, Single Source of Truth & Lifting State Up [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply State vs Props, Single Source of Truth & Lifting State Up in real Reactjs systems — specifically: Component-owned mutable state vs parent-passed r...

Module 2 Hands-On Capstone Project (100 Marks): Interactive Seat Selector, Pricing Summary & Coupon Synchronization App — Build a multi-component Workshop Seat & Batch Selector where `SeatGrid`, `BatchPicker`, and `CheckoutSummary` stay synchronized via lifted state in `EnrollmentPage`, deriving total payable ₹...

MODULE 3 OF 16 • 12 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 3: Synthetic Events, Conditional Rendering, Dynamic Lists & Reconciliation Keys

Handle cross-browser SyntheticEvents cleanly, master JSX conditional patterns without the `0 &&` trap, and render dynamic lists with stable reconciliation `key`s.

Objectives: ✓ Pass event handler function references (`onClick={handleSave}`) without invoking them during render • ✓ Avoid the classic `{count && <Badge />}` zero-rendering bug • ✓ Choose stable entity IDs over array indexes for React list `key`s

3.1 React Synthetic Events — `onClick`, Handler References, `preventDefault` & Bubbling [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Synthetic Events — `onClick`, Handler References, `preventDefault` & Bubbling in real Reactjs systems — specifically: React root-level event...

3.2 React Conditional Rendering — Ternaries, Early `if` Returns & the `0 && <JSX/>` Trap [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Conditional Rendering — Ternaries, Early `if` Returns & the `0 && <JSX/>` Trap in real Reactjs systems — specifically: Early guard `if (!dat...

3.3 Rendering Lists in React — `.map()`, `.filter()` Chains & Empty-State UX [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Rendering Lists in React — `.map()`, `.filter()` Chains & Empty-State UX in real Reactjs systems — specifically: Transforming data arrays into JSX...

3.4 React `key` Prop Internals — Reconciliation Identity, Index-Key Bugs & State Reset Trick [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React `key` Prop Internals — Reconciliation Identity, Index-Key Bugs & State Reset Trick in real Reactjs systems — specifically: How Fiber matches...

Module 3 Hands-On Capstone Project (100 Marks): Interactive Topic Question Palette & Filterable Assessment Explorer — Build an interactive Mock Exam Question Palette & Topic Explorer supporting filter tabs (`All` / `Answered` / `Flagged`), empty-state illustrations, reorderable/deletable items with stable `...

MODULE 4 OF 16 • 11 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 4: React Forms — Controlled vs Uncontrolled Inputs, Validation & Contracts

Build accessible React 19 forms using Controlled state (`value` + `onChange`), Uncontrolled `FormData` / refs, multi-field dynamic handlers, and TypeScript/PropTypes contracts.

Objectives: ✓ Prevent the "switching uncontrolled to controlled input" React warning • ✓ Handle text, select, checkbox (`e.target.checked`), and radio inputs cleanly • ✓ Validate component prop contracts and form fields with accessible error states

4.1 React Controlled Forms — `value`, `onChange`, Checkboxes (`checked`) & Multi-Input State [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Controlled Forms — `value`, `onChange`, Checkboxes (`checked`) & Multi-Input State in real Reactjs systems — specifically: Single-state obje...

4.2 Controlled vs Uncontrolled Components — `defaultValue`, `FormData`, File Inputs & Trade-offs [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Controlled vs Uncontrolled Components — `defaultValue`, `FormData`, File Inputs & Trade-offs in real Reactjs systems — specifically: When to prefe...

4.3 Component Contracts — TypeScript Props Interfaces, `PropTypes` & Runtime API Validation [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Component Contracts — TypeScript Props Interfaces, `PropTypes` & Runtime API Validation in real Reactjs systems — specifically: Documenting & enfo...

Module 4 Hands-On Capstone Project (100 Marks): Multi-Step Student Enrollment & Skill Self-Assessment Form Wizard — Create a 3-step React Admission Wizard collecting personal info, preferred track (`select`/radio), multi-checkbox skill tags, and live validation errors with Back/Next state preservation and...

MODULE 5 OF 16 • 15 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 5: Core React Hooks — `useState`, `useEffect` Synchronization, Cleanup & Lifecycles

Master the Rules of Hooks, lazy `useState(() => init())` initialization, `useEffect` dependency arrays, `AbortController` fetch cleanup, and mapping class lifecycles to Hooks.

Objectives: ✓ Obey the 2 Rules of Hooks (top-level only, React functions/custom hooks only) • ✓ Prevent infinite `useEffect` loops, stale closures, and race-condition state updates • ✓ Extract reusable stateful logic into `useDebounce`, `useLocalStorage`, and `useFetch` custom hooks

5.1 React Hooks Architecture — Rules of Hooks, Fiber Linked List & Custom Hooks (`use*`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Hooks Architecture — Rules of Hooks, Fiber Linked List & Custom Hooks (`use*`) in real Reactjs systems — specifically: Why React tracks hook...

5.2 React `useState` Deep Dive — Lazy Initialization `() => fn()`, Functional Updaters & Stale State [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React `useState` Deep Dive — Lazy Initialization `() => fn()`, Functional Updaters & Stale State in real Reactjs systems — specifically: Passing `...

5.3 React `useEffect` — Dependency Arrays, Cleanup Functions, Race Conditions & When NOT to Use Effects [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React `useEffect` — Dependency Arrays, Cleanup Functions, Race Conditions & When NOT to Use Effects in real Reactjs systems — specifically: No dep...

5.4 React Component Lifecycle — Mount, Update & Unmount (Class Methods vs Modern Hooks Mapping) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Component Lifecycle — Mount, Update & Unmount (Class Methods vs Modern Hooks Mapping) in real Reactjs systems — specifically: Mounting (`con...

Module 5 Hands-On Capstone Project (100 Marks): Live Countdown Exam Timer & Auto-Saving Workspace with Custom Hooks — Engineer a timed Assessment Workspace powered by custom hooks (`useCountdownTimer`, `useLocalStorageState`, and `useWindowVisibility`) that counts down accurately without interval drift or s...

MODULE 6 OF 16 • 12 LIVE + LAB HOURS • 4 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 6: React Fragments, `useRef`, `forwardRef` & Escape-Hatch Portals

Group sibling JSX cleanly with `<Fragment>`, manage mutable non-rendering values & DOM focus via `useRef`, forward refs to design-system inputs, and escape `overflow: hidden` via `createPortal`.

Objectives: ✓ Keep CSS Grid/Flexbox and table markup clean using <> and <Fragment key={id}> • ✓ Access DOM focus/scroll imperatively via useRef without triggering re-renders • ✓ Render accessible modals and tooltips into document.body using createPortal

6.1 React Fragments (`<>...</>` vs `<Fragment key={id}>`) — Zero Extra DOM Wrapper Nodes [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Fragments (`<>...</>` vs `<Fragment key={id}>`) — Zero Extra DOM Wrapper Nodes in real Reactjs systems — specifically: Returning multiple si...

6.2 React `useRef` — Imperative DOM Access, Mutable Instance Values & Avoiding Re-Renders [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React `useRef` — Imperative DOM Access, Mutable Instance Values & Avoiding Re-Renders in real Reactjs systems — specifically: `const ref = useRef(...

6.3 Forwarding Refs in React (`forwardRef`, React 19 `ref` Prop & `useImperativeHandle`) [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Forwarding Refs in React (`forwardRef`, React 19 `ref` Prop & `useImperativeHandle`) in real Reactjs systems — specifically: Exposing a child desi...

6.4 React Portals (`createPortal`) — Escaping `overflow: hidden` & React Event Bubbling [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Portals (`createPortal`) — Escaping `overflow: hidden` & React Event Bubbling in real Reactjs systems — specifically: Rendering modals, draw...

Module 6 Hands-On Capstone Project (100 Marks): Accessible Command-Palette Modal (`Cmd+K`) with Auto-Focus Ref & `createPortal` — Build a keyboard-triggered (`Ctrl+K` / `Cmd+K`) Search Command Palette Modal that renders into `#modal-root` via `createPortal`, auto-focuses the custom `<SearchInput>` using `useRef` + ref...

MODULE 7 OF 16 • 11 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 7: React Styling Systems — CSS Modules, Tailwind CSS in React & UI Animations

Scope component styles collision-free with CSS Modules (`*.module.css`), build rapid responsive UIs with Tailwind CSS utility composition, and animate list/modal transitions smoothly.

Objectives: ✓ Prevent global CSS class-name collisions using Vite CSS Modules • ✓ Compose clean conditional Tailwind utility classes without broken dynamic string concatenation • ✓ Animate mounting/unmounting and interactive state transitions at 60fps

7.1 Styling React Components — CSS Modules (`*.module.css`), `style` Objects & Scoped Classes [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Styling React Components — CSS Modules (`*.module.css`), `style` Objects & Scoped Classes in real Reactjs systems — specifically: Global styleshee...

7.2 Tailwind CSS in React — Variant Maps, Conditional Utilities & Design System Primitives [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Tailwind CSS in React — Variant Maps, Conditional Utilities & Design System Primitives in real Reactjs systems — specifically: Configuring Tailwin...

7.3 React UI Animations — CSS Transitions, `requestAnimationFrame`, View Transitions & Motion [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React UI Animations — CSS Transitions, `requestAnimationFrame`, View Transitions & Motion in real Reactjs systems — specifically: Animating state-...

Module 7 Hands-On Capstone Project (100 Marks): Tailwind + React Dark-Mode Dashboard & Animated Notification Toast Stack — Build a responsive Student Analytics Dashboard in React + Tailwind CSS featuring persistent Dark/Light theme switching, variant-driven `<Button>` components, and an animated Toast Notificati...

MODULE 8 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 8: Client-Side Routing (`React Router`), Code Splitting (`React.lazy`) & Error Boundaries

Build multi-view SPAs with React Router (`BrowserRouter`, `<Link>`, `useParams`, Protected Routes), slash initial bundle size via `React.lazy` + `<Suspense>`, and isolate crashes with Error Boundaries.

Objectives: ✓ Navigate without full-page reloads using <Link>, <NavLink>, <Outlet>, and useNavigate() • ✓ Split heavy admin/chart routes into on-demand async chunks with React.lazy() & <Suspense> • ✓ Catch render-phase crashes gracefully using Error Boundary fallback UIs

8.1 React Router — `BrowserRouter`, Nested `<Outlet>`, `<Link>` vs `<a>`, `useParams` & Auth Guards [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Router — `BrowserRouter`, Nested `<Outlet>`, `<Link>` vs `<a>`, `useParams` & Auth Guards in real Reactjs systems — specifically: HTML5 Hist...

8.2 React Code Splitting — Dynamic `import()`, `React.lazy()` & `<Suspense fallback>` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Code Splitting — Dynamic `import()`, `React.lazy()` & `<Suspense fallback>` in real Reactjs systems — specifically: Breaking monolithic JS b...

8.3 React Error Boundaries — `getDerivedStateFromError`, `componentDidCatch` & Crash Isolation [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Error Boundaries — `getDerivedStateFromError`, `componentDidCatch` & Crash Isolation in real Reactjs systems — specifically: Preventing a si...

Module 8 Hands-On Capstone Project (100 Marks): Role-Protected Multi-Route Learning Portal SPA with Lazy Chunks & Crash Recovery — Architect a multi-page React Router SPA (`/`, `/courses`, `/courses/:slug`, protected `/student/dashboard`, `*` 404) using nested `<Outlet />` layouts, route-level `React.lazy` + `<Suspense>...

MODULE 9 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 9: Global State Architecture — Context API, Flux Unidirectional Flow & Redux Toolkit

Eliminate deep prop-drilling with React Context (`createContext`/`useContext`), understand Flux action->reducer unidirectional flow, and manage complex global state with Redux Toolkit slices.

Objectives: ✓ Cure prop-drilling cleanly for auth, theme, and locale state via Context + custom provider hooks • ✓ Avoid Context re-render storms by splitting high-frequency state or memoizing provider values • ✓ Build predictable global state slices with Redux Toolkit (`createSlice`, `useSelector`, `useDispatch`)

9.1 React Context API (`createContext`, `useContext`) — Solving Prop Drilling & Re-Render Splits [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Context API (`createContext`, `useContext`) — Solving Prop Drilling & Re-Render Splits in real Reactjs systems — specifically: Creating type...

9.2 Flux Architecture & `useReducer` — Actions, Pure Reducers & Predictable State Machines [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Flux Architecture & `useReducer` — Actions, Pure Reducers & Predictable State Machines in real Reactjs systems — specifically: Why bidirectional M...

9.3 React Redux & Redux Toolkit (RTK) — `configureStore`, `createSlice`, Immer & `useSelector` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply React Redux & Redux Toolkit (RTK) — `configureStore`, `createSlice`, Immer & `useSelector` in real Reactjs systems — specifically: Modern Redux To...

Module 9 Hands-On Capstone Project (100 Marks): Global Cart, Auth Session & Exam Progress Store with Redux Toolkit & Context — Build an Academy Course Cart & Active Exam Session Store using React Context for Theme/Auth and Redux Toolkit (`configureStore` + `createSlice`) for multi-course cart items, coupon discounts...

MODULE 10 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Module 10: Advanced React Patterns — Higher-Order Components, `React.memo`/`useMemo` & Testing

Understand Higher-Order Components & Render Props vs Hooks, eliminate expensive re-renders with `React.memo`/`useMemo`/`useCallback`, and test user flows with Vitest + React Testing Library.

Objectives: ✓ Refactor legacy Higher-Order Components (HOCs) and understand wrapper composition • ✓ Profile first, then apply React.memo, useMemo, and useCallback where referential stability matters • ✓ Write resilient user-centric component tests using getByRole and userEvent

10.1 Higher-Order Components (HOCs) — `withAuth(Component)`, Prop Forwarding & Hooks Comparison [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Higher-Order Components (HOCs) — `withAuth(Component)`, Prop Forwarding & Hooks Comparison in real Reactjs systems — specifically: Functions takin...

10.2 `React.memo`, `PureComponent`, `useMemo` & `useCallback` — Referential Equality & Profiling [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply `React.memo`, `PureComponent`, `useMemo` & `useCallback` — Referential Equality & Profiling in real Reactjs systems — specifically: Why children r...

10.3 Testing React Components — Vitest, React Testing Library (`getByRole`, `findByText`) & `userEvent` [50 mins • Assignment (100M) • Hands-On Lab (50M) • 10 Mock Sets • 10 AI Vivas]

Purpose: To understand and apply Testing React Components — Vitest, React Testing Library (`getByRole`, `findByText`) & `userEvent` in real Reactjs systems — specifically: Testing...

Module 10 Hands-On Capstone Project (100 Marks): Virtualized 1,000-Row Leaderboard & Full React Testing Library Suite — Build a high-FPS Student Leaderboard filtering & sorting 1,000+ rows using memoized row components (`React.memo`), `useMemo` derived rankings, `useCallback` stable handlers, and a comprehens...

MODULE 11 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Server-State Caching with TanStack Query & Optimistic UI Updates

Production-grade Reactjs Training engineering module covering Server-State Caching with TanStack Query & Optimistic UI Updates with hands-on labs and capstone build.

Objectives: ✓ Declarative Queries, Cache Keys, StaleTime & Background Refetching — Purpose: Eliminate manual loading/error useState boilerplate across every API-driven screen. • ✓ Optimistic Mutations with Automatic Error Rollback — Purpose: Update task statuses and likes in 0ms while syncing safely with backend APIs. • ✓ Infinite Cursor Lists, Prefetch-on-Hover & Dependent Queries — Purpose: Make multi-page React dashboards feel instantaneous during navigation.

11.1 Declarative Queries, Cache Keys, StaleTime & Background Refetching [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Declarative Queries, Cache Keys, StaleTime & Background Refetching in real Reactjs systems — specifically: Eliminate manual loading/error useState...

11.2 Optimistic Mutations with Automatic Error Rollback [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Optimistic Mutations with Automatic Error Rollback in real Reactjs systems — specifically: Update task statuses and likes in 0ms while syncing saf...

11.3 Infinite Cursor Lists, Prefetch-on-Hover & Dependent Queries [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Infinite Cursor Lists, Prefetch-on-Hover & Dependent Queries in real Reactjs systems — specifically: Make multi-page React dashboards feel instant...

Module 11 Hands-On Capstone Project (100 Marks): Real-Time SaaS CRM Board with Instant Optimistic Mutations — Design, implement, test, and document 'Real-Time SaaS CRM Board with Instant Optimistic Mutations' applying every production concept from Module 11 (Server-State Caching with TanStack Query...

MODULE 12 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

TypeScript for React: Generic Components, Discriminated Props & Forms

Production-grade Reactjs Training engineering module covering TypeScript for React: Generic Components, Discriminated Props & Forms with hands-on labs and capstone build.

Objectives: ✓ Typing Props, Children, DOM Events & Generic Table/Select Components — Purpose: Catch missing props and wrong callback arguments in VS Code before saving a file. • ✓ Schema-Driven Forms with React Hook Form & Zod Validation — Purpose: Validate 30-field dynamic onboarding forms with zero unnecessary component re-renders. • ✓ Compound Components, Portals & Accessible Headless Dialogs — Purpose: Build reusable enterprise UI primitives without prop-drilling nightmares.

12.1 Typing Props, Children, DOM Events & Generic Table/Select Components [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Typing Props, Children, DOM Events & Generic Table/Select Components in real Reactjs systems — specifically: Catch missing props and wrong callbac...

12.2 Schema-Driven Forms with React Hook Form & Zod Validation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Schema-Driven Forms with React Hook Form & Zod Validation in real Reactjs systems — specifically: Validate 30-field dynamic onboarding forms with...

12.3 Compound Components, Portals & Accessible Headless Dialogs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Compound Components, Portals & Accessible Headless Dialogs in real Reactjs systems — specifically: Build reusable enterprise UI primitives without...

Module 12 Hands-On Capstone Project (100 Marks): Type-Safe Enterprise DataGrid & Multi-Step Validated Wizard — Design, implement, test, and document 'Type-Safe Enterprise DataGrid & Multi-Step Validated Wizard' applying every production concept from Module 12 (TypeScript for React: Generic Components...

MODULE 13 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

React Render Profiling, Virtualization, Suspense & Error Boundaries

Production-grade Reactjs Training engineering module covering React Render Profiling, Virtualization, Suspense & Error Boundaries with hands-on labs and capstone build.

Objectives: ✓ React Profiler Flamegraphs, Referential Stability & Render Isolation — Purpose: Pinpoint and fix cascade re-renders slowing down complex interactive screens. • ✓ Windowed List Virtualization & Route-Based Lazy Code Splitting — Purpose: Keep initial JS bundles compact and scroll 50,000 rows smoothly on budget laptops. • ✓ Granular Error Boundaries & Graceful Widget Recovery — Purpose: Isolate third-party chart or network crashes to a single card with a Retry button.

13.1 React Profiler Flamegraphs, Referential Stability & Render Isolation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply React Profiler Flamegraphs, Referential Stability & Render Isolation in real Reactjs systems — specifically: Pinpoint and fix cascade re-renders s...

13.2 Windowed List Virtualization & Route-Based Lazy Code Splitting [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Windowed List Virtualization & Route-Based Lazy Code Splitting in real Reactjs systems — specifically: Keep initial JS bundles compact and scroll...

13.3 Granular Error Boundaries & Graceful Widget Recovery [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Granular Error Boundaries & Graceful Widget Recovery in real Reactjs systems — specifically: Isolate third-party chart or network crashes to a sin...

Module 13 Hands-On Capstone Project (100 Marks): 60fps Financial Analytics Workspace with 50,000-Row Virtual Table — Design, implement, test, and document '60fps Financial Analytics Workspace with 50,000-Row Virtual Table' applying every production concept from Module 13 (React Render Profiling, Virtualiza...

MODULE 14 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Testing React Apps (Vitest, Testing Library, Playwright) & Prod Deploy

Production-grade Reactjs Training engineering module covering Testing React Apps (Vitest, Testing Library, Playwright) & Prod Deploy with hands-on labs and capstone build.

Objectives: ✓ Component Behavior Testing with Vitest & React Testing Library — Purpose: Test real clicks, keyboard typing, and ARIA roles just like an actual user. • ✓ Mocking REST APIs Cleanly with MSW (Mock Service Worker) — Purpose: Verify loading skeletons, 500 errors, and empty states deterministically in CI. • ✓ Env Config, SPA Nginx/CDN Rewrite Rules & Sentry Source Maps — Purpose: Deploy crash-monitored React SPAs where deep links refresh cleanly without 404s.

14.1 Component Behavior Testing with Vitest & React Testing Library [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Component Behavior Testing with Vitest & React Testing Library in real Reactjs systems — specifically: Test real clicks, keyboard typing, and ARIA...

14.2 Mocking REST APIs Cleanly with MSW (Mock Service Worker) [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Mocking REST APIs Cleanly with MSW (Mock Service Worker) in real Reactjs systems — specifically: Verify loading skeletons, 500 errors, and empty s...

14.3 Env Config, SPA Nginx/CDN Rewrite Rules & Sentry Source Maps [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Env Config, SPA Nginx/CDN Rewrite Rules & Sentry Source Maps in real Reactjs systems — specifically: Deploy crash-monitored React SPAs where deep...

Module 14 Hands-On Capstone Project (100 Marks): CI-Verified React Production Build with E2E Checkout Tests — Design, implement, test, and document 'CI-Verified React Production Build with E2E Checkout Tests' applying every production concept from Module 14 (Testing React Apps (Vitest, Testing Libra...

MODULE 15 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

Server-State Caching, Optimistic Mutations & Accessible React Forms

Self-explanatory list-and-bullet production mastery module covering Server-State Caching, Optimistic Mutations & Accessible React Forms in Reactjs Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ Declarative Data Fetching, Stale-While-Revalidate Caching & Race-Condition Cancellation — Purpose: Eliminate duplicate API spinners, stale tab data, and out-of-order fetch overwrites in React. • ✓ Optimistic UI Updates, Automatic Rollback on Failure & Toast Feedback — Purpose: Make enterprise dashboards feel instantaneous while keeping server state 100% consistent. • ✓ Schema-Validated Complex Forms, Accessible Error Focus & Dirty-State Guards — Purpose: Handle dynamic field arrays, async validation, and unsaved-changes protection cleanly. • ✓ Build, verify, and document 'Optimistic Kanban & Validated Multi-Step Form Workspace in React' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

15.1 Declarative Data Fetching, Stale-While-Revalidate Caching & Race-Condition Cancellation [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Declarative Data Fetching, Stale-While-Revalidate Caching & Race-Condition Cancellation in real Reactjs systems — specifically: Eliminate duplicat...

15.2 Optimistic UI Updates, Automatic Rollback on Failure & Toast Feedback [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Optimistic UI Updates, Automatic Rollback on Failure & Toast Feedback in real Reactjs systems — specifically: Make enterprise dashboards feel inst...

15.3 Schema-Validated Complex Forms, Accessible Error Focus & Dirty-State Guards [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Schema-Validated Complex Forms, Accessible Error Focus & Dirty-State Guards in real Reactjs systems — specifically: Handle dynamic field arrays, a...

Module 15 Hands-On Capstone Project (100 Marks): Optimistic Kanban & Validated Multi-Step Form Workspace in React — Architect, implement, test, and document 'Optimistic Kanban & Validated Multi-Step Form Workspace in React' applying every bullet-wise engineering concept from Module 15 (Server-State Cachin...

MODULE 16 OF 16 • 14 LIVE + LAB HOURS • 3 TOPICS • 1 CAPSTONE PROJECT (100 MARKS)

React Testing Library, Bundle Profiling & Frontend Interview Architecture

Self-explanatory list-and-bullet production mastery module covering React Testing Library, Bundle Profiling & Frontend Interview Architecture in Reactjs Training, paired with hands-on labs, capstone repository, Topic Mock Tests, and Spoken Gemini AI Viva.

Objectives: ✓ Component & User-Flow Testing with Vitest, React Testing Library & Mocked APIs — Purpose: Test real user clicks, keyboard input, and async states instead of brittle internal implementation details. • ✓ React DevTools Profiler, Virtualized Lists & Route-Level Bundle Splitting — Purpose: Render 10,000-row tables smoothly at 60fps and shrink initial JavaScript download size. • ✓ React Machine-Coding Challenges, Hook Pitfalls & Frontend System Design — Purpose: Avoid stale closures and infinite useEffect loops while acing live React coding interviews. • ✓ Build, verify, and document 'Vitest + React Testing Library Suite & Flamegraph Render Optimizer' with clean Git commits and edge-case protection. • ✓ Clear 100% of Topic Timed Mock Tests and Spoken Voice + Google Gemini AI Technical Interviews.

16.1 Component & User-Flow Testing with Vitest, React Testing Library & Mocked APIs [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply Component & User-Flow Testing with Vitest, React Testing Library & Mocked APIs in real Reactjs systems — specifically: Test real user clicks, keyb...

16.2 React DevTools Profiler, Virtualized Lists & Route-Level Bundle Splitting [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply React DevTools Profiler, Virtualized Lists & Route-Level Bundle Splitting in real Reactjs systems — specifically: Render 10,000-row tables smoothl...

16.3 React Machine-Coding Challenges, Hook Pitfalls & Frontend System Design [65 mins • Assignment (100M) • Hands-On Lab (50M) • 2 Mock Sets • 2 AI Vivas]

Purpose: To understand and apply React Machine-Coding Challenges, Hook Pitfalls & Frontend System Design in real Reactjs systems — specifically: Avoid stale closures and infinite...

Module 16 Hands-On Capstone Project (100 Marks): Vitest + React Testing Library Suite & Flamegraph Render Optimizer — Architect, implement, test, and document 'Vitest + React Testing Library Suite & Flamegraph Render Optimizer' applying every bullet-wise engineering concept from Module 16 (React Testing Lib...

3. LIVE CLIENT PRODUCTION PROJECT, GLOBAL CERTIFICATIONS & CAREER SCOPE

End-of-Course Live Client Production Project + Official Internship Confirmation Letter

Project: PulseBoard Real-Time SaaS Operations & Team Workflow Web Application • **Confirmed Role:** React.js Frontend Engineering Intern

Domain & Production Infra: B2B Productivity & Operations SaaS Client (React + Custom Hooks + Code-Split Vite Bundle + Cloud HTTPS Hosting)

- Built modular React component hierarchies with custom hooks, Context/Redux state management, and protected routing.
- Integrated live REST/WebSocket endpoints with optimistic UI updates, skeleton states, and error boundaries.
- Eliminated unnecessary re-renders via memoization, route-level lazy code splitting, and virtualized lists.
- Deployed tested production frontend builds with automated CI checks and live client acceptance sign-off.

Aligned Global Vendor Certifications: **META-FE** (Meta Certified Front-End & React Developer Credential) • **JSNAD** (OpenJS Node.js Application Developer (JSNAD))

Target Job Roles: React.js Frontend Developer, Frontend Product Engineer (SDE-1), UI Application Engineer, SPA Web Developer

Realistic Annual CTC Scope: **Fresher / Entry-Level (0–1 Yr):** ₹5.0 LPA – ₹9.5 LPA **Mid-Level React Product Engineer (1–3 Yrs):** ₹10.0 LPA – ₹20.0 LPA **Senior Frontend / Full-Stack Architect (3–6+ Yrs):** ₹20.0 LPA – ₹38.0+ LPA

MockAttempt Academy Pvt. Ltd. • Email: support@mockattempt.com • Phone/WhatsApp: +91 98765 43210 • Web: https://mockattempt.com/courses/reactjs-training

Try 2 Free Topic-Wise 50-MCQ Mock Tests & Spoken Gemini AI Interviews daily on every topic above before enrolling.