

KAUR'S CODING ACADEMY

React JS Frontend Development

Module-wise Teacher Syllabus

A modern React JS syllabus including React overview, JSX, components, props, state, lifecycle, forms, events, refs, router, Flux/Redux, hooks, APIs, optimization, animation, deployment and testing.

Course	Audience	Suggested Duration	Final Output
React JS Frontend Development	Grades 9+, college students and professionals	66 classes / 12 to 16 weeks	Deployed React application with routing, API integration, state management and testing basics

Teacher Note

Follow the module sequence carefully. Begin with visual examples, then build reusable components and gradually introduce routing, APIs, state management and deployment.

Learning Outcomes

- Understand React features, advantages, limitations and single page application architecture.
- Build reusable components using JSX, props, state and hooks.
- Handle forms, events, refs, routing, API calls and global state.
- Use Redux/Flux concepts, performance optimization, animations and deployment workflow.
- Complete a professional React project with GitHub and live demo.

Document Use

This PDF is made for teachers to plan classes module by module. Each module includes topic names, teaching points, class practice and revision work.

Complete Module Index

The table below gives a quick view of what will be taught in this course.

Module	Topic Name	Main Things Teacher Will Cover
1	React JS - Overview	What is React JS?, React features, Component-based UI, Virtual DOM concept, and more
2	React JS - Environment Setup	Install Node.js and npm, NPM vs NPX vs Yarn, Create React app using Vite or suitable setup, Root folder and project structure, and more
3	React JS - JSX	JSX meaning and rules, Nested elements, Attributes and className, JavaScript expressions in JSX, and more
4	React JS - Components	Functional components, Class component overview, Stateless component example, Stateful component example, and more
5	React JS - Props Overview	What are props?, Passing props from parent to child, Using props inside components, Default props, and more
6	React JS - Props Validation	Why props validation is useful, PropTypes overview, Validating string, number, boolean and array props, Required props, and more
7	React JS - State and Component API	State concept, setState in class components, useState in functional components, Force update concept, and more
8	React JS - Component Life Cycle	Mounting, updating and unmounting, Lifecycle methods overview, componentDidMount and componentWillUnmount concept, useEffect as modern lifecycle handling, and more
9	React JS - Forms	Controlled components, Uncontrolled components overview, Simple form example, Complex form example, and more
10	React JS - Events	React event system, Click and change events, Keyboard events, Focus and blur events, and more
11	React JS - Refs and Keys	Using refs, useRef hook, Accessing DOM elements carefully, Focus input using ref, and more
12	React JS - Router	Install React Router, Create route components, BrowserRouter, Routes, Route and Link, Navigation bar, and more
13	React JS - Flux Concept	Need for predictable state flow, Flux architecture overview, Actions, dispatcher, store and view, Unidirectional data flow, and more
14	React JS - Using Redux / Flux Pattern	Install Redux and React-Redux, Create folders and files, Actions, Reducers, and more
15	React JS - Hooks	useState, useEffect, useRef, useCallback, and more
16	React JS - Context API	Prop drilling problem, Create context, Provider and consumer/useContext, Global theme/user state, and more
17	React JS - Working with APIs	API workflow, Axios and fetch, GET, POST, PUT and DELETE operations, Promises and async/await, and more
18	React JS - Error Handling and Error Boundaries	try/catch with async code, Error messages in UI, Error boundary concept, Fallback UI, and more
19	React JS - Rendering Optimization	Unnecessary re-renders, React.memo, useMemo, useCallback, and more
20	React JS - Animations	CSS transitions in React, Conditional class animations, Appear animation, Enter and leave animations, and more
21	React JS - Deployment	Git and GitHub setup, Build command, Netlify/Vercel deployment, Environment variables, and more
22	React JS - Testing and Final Review	Jest introduction, Component testing idea, React Testing Library overview, Mocking basics, and more

Detailed Module-wise Teaching Plan

Teachers should first explain the concept in simple language, then demonstrate code, then allow students to practice and customize the output.

Module 1: React JS - Overview

Topics to be Covered	What Teacher Will Explain / Demonstrate
- What is React JS? - React features - Component-based UI - Virtual DOM concept - React advantages - React limitations - Single Page Application idea	- Explain why React is used in modern frontend development - Compare whole page reload with component re-rendering - Discuss when React is useful and when it may be too much
Class Practice / Student Output	Homework / Revision
- Create a simple diagram of React app flow - Identify components in a sample website - Discuss React use cases	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 2: React JS - Environment Setup

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Install Node.js and npm - NPM vs NPX vs Yarn - Create React app using Vite or suitable setup - Root folder and project structure - Add dependencies and plugins - index.html, main.jsx and App.jsx - Running the development server	- Show installation checks with node -v and npm -v - Demonstrate project creation and folder structure - Explain package.json and node_modules in simple words
Class Practice / Student Output	Homework / Revision
- Create first React project - Run local server - Edit App.jsx and see live changes	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 3: React JS - JSX

Topics to be Covered	What Teacher Will Explain / Demonstrate
- JSX meaning and rules - Nested elements - Attributes and className - JavaScript expressions in JSX - Styling in JSX - Comments in JSX - Naming conventions	- Explain JSX as JavaScript-friendly HTML-like syntax - Show difference between HTML class and JSX className - Demonstrate using variables inside JSX
Class Practice / Student Output	Homework / Revision
- Create JSX profile card - Use expressions for dynamic data - Add inline and CSS class styling	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 4: React JS - Components

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Functional components - Class component overview - Stateless component example - Stateful component example - Component file organization - Export and import components	- Explain components as reusable UI blocks - Demonstrate breaking a page into header, card and footer - Show difference between static and dynamic components
Class Practice / Student Output	Homework / Revision
- Create Header, Card and Footer components - Import components into App.jsx - Reuse one component multiple times	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 5: React JS - Props Overview

Topics to be Covered	What Teacher Will Explain / Demonstrate
- What are props? - Passing props from parent to child - Using props inside components - Default props - Props.children - State and props difference	- Explain props as input data for components - Demonstrate dynamic cards using props - Show when to use state and when to use props
Class Practice / Student Output	Homework / Revision
- Build product cards using props - Add default values - Pass children content to component	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 6: React JS - Props Validation

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Why props validation is useful - PropTypes overview - Validating string, number, boolean and array props - Required props - Shape validation for objects	- Explain how validation helps avoid wrong data - Demonstrate PropTypes usage - Show warning example when wrong prop type is passed
Class Practice / Student Output	Homework / Revision
- Add prop validation to Card component - Test with wrong prop type - Fix warning by passing correct data	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 7: React JS - State and Component API

Topics to be Covered	What Teacher Will Explain / Demonstrate
- State concept - setState in class components - useState in functional components - Force update concept - findDOMNode concept and modern alternatives - State update best practices	- Explain state as changeable component data - Demonstrate counter with useState - Discuss older class APIs and modern React approach
Class Practice / Student Output	Homework / Revision
- Create counter component - Create show/hide toggle - Update state using previous value	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 8: React JS - Component Life Cycle

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Mounting, updating and unmounting - Lifecycle methods overview - componentDidMount and componentWillUnmount concept - useEffect as modern lifecycle handling - Dependency array - Cleanup function	- Explain component life cycle with real-life birth-update-remove example - Demonstrate useEffect after render - Show cleanup with timer/listener example
Class Practice / Student Output	Homework / Revision
- Build timer component - Fetch data on mount - Add cleanup function	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 9: React JS - Forms

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Controlled components - Uncontrolled components overview - Simple form example - Complex form example - Input state handling - Validation and error messages - React Hook Form introduction	- Explain data flow from input to state - Demonstrate onChange and onSubmit - Show validation using state and regex
Class Practice / Student Output	Homework / Revision
- Build registration form - Add validation messages - Submit and display entered data	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 10: React JS - Events

Topics to be Covered	What Teacher Will Explain / Demonstrate
- React event system - Click and change events - Keyboard events - Focus and blur events - Passing arguments to handlers - Child-to-parent event communication	- Explain event handlers in React - Show onClick and onChange examples - Demonstrate child component button sending data to parent
Class Practice / Student Output	Homework / Revision
- Create button actions - Build search input event - Trigger parent function from child component	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 11: React JS - Refs and Keys

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Using refs - useRef hook - Accessing DOM elements carefully - Focus input using ref - List rendering keys - Why keys should be stable	- Explain refs as direct access when needed - Demonstrate input focus with ref - Show list rendering with proper keys
Class Practice / Student Output	Homework / Revision
- Create focus input button - Render list of items with keys - Fix missing key warning	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 12: React JS - Router

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Install React Router - Create route components - BrowserRouter, Routes, Route and Link - Navigation bar - Route parameters - Nested routes - Default and not-found routes	- Explain SPA routing without page reload - Demonstrate creating Home, About and Contact pages - Show dynamic product details route
Class Practice / Student Output	Homework / Revision
- Build multi-page React app - Add navigation menu - Create dynamic detail page	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 13: React JS - Flux Concept

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Need for predictable state flow - Flux architecture overview - Actions, dispatcher, store and view - Unidirectional data flow - Flux pros and use cases	- Explain state flow using a simple diagram - Compare local component state with app-level flow - Discuss why larger apps need organized state
Class Practice / Student Output	Homework / Revision
- Draw Flux flow for a shopping cart - Identify action, store and view in example - Prepare for Redux module	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 14: React JS - Using Redux / Flux Pattern

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Install Redux and React-Redux - Create folders and files - Actions - Reducers - Store - Root component provider - Connecting other components - Thunk middleware overview	- Explain Redux flow step by step - Demonstrate action -> reducer -> store -> UI update - Show API call idea with thunk
Class Practice / Student Output	Homework / Revision
- Build cart counter using Redux - Create actions and reducer - Connect component to store	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 15: React JS - Hooks

Topics to be Covered	What Teacher Will Explain / Demonstrate
- useState - useEffect - useRef - useCallback - useMemo - useContext - Custom hooks	- Explain why hooks are used in functional components - Demonstrate each hook with a small example - Show how custom hooks reuse logic
Class Practice / Student Output	Homework / Revision
- Build API data hook - Use useMemo/useCallback for optimization example - Create context-based theme toggle	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 16: React JS - Context API

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Prop drilling problem - Create context - Provider and consumer/useContext - Global theme/user state - Context vs Redux	- Explain context through user login/theme example - Demonstrate provider wrapping routes - Discuss when Context is enough and when Redux is better
Class Practice / Student Output	Homework / Revision
- Build theme context - Toggle light/dark mode - Use context in multiple components	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 17: React JS - Working with APIs

Topics to be Covered	What Teacher Will Explain / Demonstrate
- API workflow - Axios and fetch - GET, POST, PUT and DELETE operations - Promises and async/await - Loading states - Error messages - Displaying API response data	- Explain frontend-backend communication - Demonstrate API call with axios/fetch - Show loading and error state handling
Class Practice / Student Output	Homework / Revision
- Build search app using API - Display data in cards/table - Handle error and empty states	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 18: React JS - Error Handling and Error Boundaries

Topics to be Covered	What Teacher Will Explain / Demonstrate
- try/catch with async code - Error messages in UI - Error boundary concept - Fallback UI - Debugging React errors	- Explain different types of errors - Demonstrate handling failed API request - Show simple error boundary fallback
Class Practice / Student Output	Homework / Revision
- Add error UI to API app - Create error boundary component - Test failed request state	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 19: React JS - Rendering Optimization

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Unnecessary re-renders - React.memo - useMemo - useCallback - Code splitting - Lazy loading - React Suspense fallback	- Explain performance in simple terms - Demonstrate memoizing a component - Show lazy-loaded route with fallback
Class Practice / Student Output	Homework / Revision
- Optimize a component - Add lazy loading to one route - Observe render behavior with console logs	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 20: React JS - Animations

Topics to be Covered	What Teacher Will Explain / Demonstrate
- CSS transitions in React - Conditional class animations - Appear animation - Enter and leave animations - React transition libraries overview - Framer Motion overview	- Explain animation as state-driven UI change - Demonstrate adding/removing classes - Show enter/exit animation for list item
Class Practice / Student Output	Homework / Revision
- Animate cards or modal - Add page transition idea - Create simple list enter/leave animation	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 21: React JS - Deployment

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Git and GitHub setup - Build command - Netlify/Vercel deployment - Environment variables - Common deployment errors - Live link sharing	- Demonstrate pushing React project to GitHub - Show build and deployment steps - Explain how to fix base path and environment issues
Class Practice / Student Output	Homework / Revision
- Deploy React project - Share live link - Update README with screenshots	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Module 22: React JS - Testing and Final Review

Topics to be Covered	What Teacher Will Explain / Demonstrate
- Jest introduction - Component testing idea - React Testing Library overview - Mocking basics - Testing async code overview - Course recap and Q&A;	- Explain why testing matters - Demonstrate a simple component test - Review complete course through capstone checklist
Class Practice / Student Output	Homework / Revision
- Write one simple test - Prepare final React capstone - Present project and answer review questions	- Revise the class notes and complete a small improvement in the class project. - Write down two doubts or two new things learned in the module.

Final Review and Certification Checklist

- Student can explain the main concepts in simple words.
- Student has completed the required practice tasks and mini projects.
- Student can run the project, fix common errors and present the work confidently.
- For advanced courses, student should submit GitHub repository, screenshots and deployment link where possible.