React JS Complete Guide - Step-by-Step Concepts Index

1. Introduction to React JS

- What is React JS?

- Features of React JS

- Benefits of using React

- Comparison with other frameworks (Angular, Vue.js)

2. Installation and Setup

- Different ways to install React

- Using CDN: Add React to an existing HTML website

- Using create-react-app

- Using Vite for React

- Installing React manually via npm

- Creating Your First React Project

- Setting up the development environment

- Creating a new React project

- Project structure overview

- Running the project and understanding the initial files

- "Hello World" in React

3. React DOM

- What is React DOM?

- Rendering HTML elements using ReactDOM.render()

4. SPA (Single Page Application) vs Traditional Web Applications

- Definition and differences

- Advantages of SPA

5. JSX

- What is JSX?

- Why use JSX?

- Rules of JSX (single parent element, must be closed tags, etc.)

- Convert HTML to JSX

6. Components in React

- What is a component?

- Types of Components

- Functional Components

- Class Components

- Difference between Functional and Class Components

7. Class Lifecycle Methods

- Mounting: constructor(), render(), componentDidMount()

- Updating: shouldComponentUpdate(), componentDidUpdate()

- Unmounting: componentWillUnmount()

8. Import and Export Components

- Default export vs Named export

- Importing components in React

9. Working with Multiple Components

- Component Nesting and Composition

- Conditional Rendering

- Using ternary operators

- Using if and if..else conditions

- Using switch-case

10. State in React

- What is State and its Importance?

- Initializing State

- Updating State

- Updating Arrays in State

- Updating Objects in State

- Setstate in Class Components

11. Event Handling

- Commonly used events: onClick, onChange, onSubmit, onMouseEnter, onKeyDown, etc.

- Creating Event Handlers

- UseState Hook Introduction

12. Practical Projects

- Project 1: Accept input value from user and display it in an <h1> tag

- Project 2: Calculator app using state

- Project 3: Hide/Unhide div on button click

13. Props and PropTypes

- What are Props?

- Passing Props to Components

- PropTypes for type-checking

14. Rendering Lists

- Using Map() Function

- Keys and Their Importance

15. Refs in React

- What are Refs?

- Uses of Refs

- Using useRef Hook

16. React Hooks

- useState

- useEffect (with practical examples)

- useContext

- useReducer

- useCallback

- useMemo

- Custom Hooks (Creating your own hooks)

17. Forms and Form Handling

- Creating Forms in React

- Form Submission and Validation

- Handling input fields

- React Hook Form (introduction and usage)

18. CRUD Operations

- CRUD using Arrays/Maps (no database)

- Render records in a table

- Add, Edit, and Delete operations

- CRUD using Local Storage

- Store, retrieve, update, and delete data using local storage

19. React Styling

- Basic CSS in React

- Inline CSS

- Using external CSS files

20. Material UI, Tailwind, and Bootstrap

- React Bootstrap (Buttons, grid, layout, components)

- Tailwind in React (installation and usage)

- Material UI

- Installation

- Advantages of Material UI

- Material UI Components (Buttons, grids, stack, typography, icons)

- Layout Components (Appbar, Drawer, etc.)

- Form Components (Textfields, checkboxes, validation)

- Dialogs and Modals

- Customizing Material UI Theme

21. React Router

- Installation of React Router

- BrowserRouter, Router, Route, Link, NavLink

- Route Parameters (useParams)

- Nested Routes

- Programmatic Navigation (using history)

- Lazy Loading for Performance Optimization

- Authentication and Authorization

- Redirect Unauthorized Users to Login Page

22. Animations in React

- CSS Transitions

- Using libraries like React Transition Group

23. Error Handling

- Handling errors in React

- Creating a 404 Page Not Found

24. API Integration

- Using API or FakeJsonAPI

- API Testing using Postman

25. Firebase Integration

- Installing Firebase SDK

- NoSQL Database in Firebase

- Creating Collections and Documents

- Firebase Authentication

- Firebase Storage

26. Redux

- What is Redux?

- Why use Redux?

- Redux Principles

- Core Concepts: Actions, Reducers, Store, Middleware

- React-Redux

- Setting up a Reducer and Store

- Dispatching Actions

- Combining Multiple Reducers

- Adding Middleware

- Redux DevTools

- Redux Thunk Middleware

27. Practical Projects

- CRUD Operations using Redux

- Create, Read, Update, and Delete operations

- Implementing Redux actions and reducers

- Task Manager Application

- Pending tasks, completed tasks

- Select all, deselect all functionality

28. Projects and Use Cases

- Project 1: CRUD operations using API or FakeJsonAPI

- Project 2: CRUD using Local Storage

- Project 3: Task Manager Application

- Project 4: Firebase Authentication (Registration, Login, Profile Updation)

- Project 5: Like/Dislike post application using Firebase storage

29. Advanced Topics

- State Management Problems

- Handling Complex State with Redux

- Middleware in Redux (Redux Thunk)

- Performance Optimization Techniques

- Lazy Loading Components

- Code Splitting

30. Conclusion

- Recap of Key Concepts

- What's Next?

- Resources and Learning Materials