REACTJS

Introduction to React.js

- Understanding the basics of React.js
- Setting up a React development environment (Node.js, npm/yarn)
- Creating React components using functional and class components
- JSX syntax and its importance in React development
- Rendering elements and components in React
- State and Props in React components

React Component Lifecycle and State Management

- Overview of React component lifecycle methods
- Using state and setState() method to manage component state
- Handling user input with controlled and uncontrolled components
- Conditional rendering in React using if statements and ternary operators
- Lists and keys in React: rendering lists of elements dynamically
- Introduction to React Hooks: useState, useEffect, useContext

Advanced React Concepts and Styling

- Deeper dive into React Hooks: useRef, useMemo, useCallback
- Context API for state management and avoiding prop drilling
- Error boundaries in React for better error handling
- Using Higher Order Components (HOCs) for code reuse
- CSS-in-JS: Styling React components using libraries like styled-components
- Responsive design techniques with CSS media queries

Routing and Forms in React

- Setting up client-side routing in React with React Router
- Creating and configuring routes with React Router
- Handling forms and form submissions in React
- Form validation using built-in and custom validation methods
- Handling file uploads in React applications
- Advanced form handling techniques: form libraries like Formik or React Hook Form

Managing Application State with Redux

- Introduction to Redux and its principles: Store, Actions, Reducers
- Setting up Redux in a React application using react-redux

- Creating Redux actions and action creators
- Writing Redux reducers to manage application state
- Connecting React components to Redux store using connect()
- Best practices for structuring Redux code and organizing application state