**React Topic**

**Section1: Getting Started**

* History of React Js
* What is React ?
* React JS V/S Venilla JavaScript
* How to create React App (create-react-app/vite)

**Section3: React Essentitals**

* Component, Rules for Component creation
* JSX
* Built in Component V/S Custom Component
* Output dynamic data in jsx code
* Props, Children Props
* Named & default export
* Events
* Import images
* How react checks if UI updates are needed
* Rules for Hooks (How to use Hooks in Component etc)
* useState()
* Dynamic list (map function)

**Section4&5: React Essential Deep Dive**

* Fragment
* Public V/S assets folder
* How to update state (best way to do this)
* Two-way binding
* Update object state immutably (video:80)
* Lifting state up (create state in parent component)

**Section6: Styling React Component**

* Styling with Vanilla css
* Scoping Styles with css module
* Css in JS (Inline CSS)
* Styled Component (library)
* Styling with Tailwind css
* Static and Dynamic styling

**Section7: Debugging React Apps**

* Finding logical errors via the browser
* Enabling react strict mode
* Using the React Devtools for application analysis and manipulation

**Section8: Refs & Portals**

* useRef
* useState V/S useRef
* Portal (createPortal)
* Modal-dialog box (forwardRef, useImperativeHandle)
* Dialog tag (handleReset)
* setInterval and clearInterval method
* What is Prop Drilling

**Section9: Revision of all concept which we learn till sec-8**

**Section10: React Context API & useReducer**

* Problem in shared state (prop drilling)
* Solution of prop drilling
* Context Api, createContext(), useReducer(), useContext()
* CartContext.Provider, CartContext.Consumer

**Section11: Handling side effect & working with useEffect**

* Side Effect, useEffect(), \*useEffect() clean up function
* Local storage in browser
* Json.parse(), Json.stringfy()
* useCallback()

**Section12: Practice Project | Quiz-App**

**Section13: Optimizing Techniques of React App**

* memo(), useMemo()
* cleaver Component Composition techniques
* useCallback() deep dive
* useMemo() V/S useCallback()
* Working of DOM
* Closer look at “keys”
* State scheduling & batching

**Section14: Class Based Component**

* Render(), state creation, props passing
* Life cycle method
* componentDidMount()
* componentDidUpdate()
* componentWillUnmount()

**Section15: Sending Http Request**

**Section19: Diving Into Redux**

* What is Redux
* Local state V/S Cross component state V/S App-wide state
* Redux V/S useContext
* Working of Redux, How it work
* Communication between Reducer function and Component
* Way of creating redux store

1. Using createStore
2. Using Redux Toolkit (createSlice())

* redux and react-redux package
* useStore(), useSelector(), useDispatch() hooks
* configureStore()(use for creating store in reactToolkit)
* why ReduxToolkit istead of createStore()

**Section20: Advanced Redux**

* Firebase(dummy backend)
* Why we don’t write side effect and asynchronous code inside reducer function.
* Where should we put side effect code (useEffect and function)video:312
* Action creators

**Section21: SPA(Single Page Application)**

* What is a Routing
* Multi page routing
* Building SPA, Installing react-router-dom
* Routing (RouterProvider, Link to, NavLink, useNavigate, <Outlet/>)
* Coustom Error page
* Dynamic path segmants or Path parameter
* Absolute and Relative path
* Back button **<Link to='..'>Back</Link>**
* Data Fetching and Submission
* useNavigation(), useParam() hook
* Error handling in loader function
* useLoaderData(), useRouteLoadData() hook
* useSubmit() hook