1)Why we learn ReactJs

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. What is Reactjs
2. Prerequisite
3. Key-feature of reactJs:
4. Software

\*\*\*Quiz-Assignment\*\*\*

1. Components type: classBased

* Extracting Components [Default Export + Import]
* Stateful vs Stateless Component
* Class-based Components.
* Class-based vs Functional Component
* Callback function in setState()
* Rendering Components Dynamically [Object + List]
* Extracting Components [Default Export + Import]
* Understanding Props
* Styling Components -External Styles
* Working with React DevTools
* Component Lifecycle
* DOM vs Virtual DOM
* Creation Lifecycle
* Updation Lifecycle
* shouldComponentUpdate() for Optimization
* Pure Components

1. Components type: Functional
2. Connecting To Backend

* HTTP requests in React
* Installing and Understanding Axios
* Fetching data from Server
* Handling Network Errors

1. Routing:

* Setting up React-Router
* Links
* Routes
* Passing Route Parameters
* Redirecting Requests
* Conditional Redirects
* Handling 404

1. Forms/Props Types

* Controlled and Uncontrolled Components
* Sending data to Server
* Deleting data on Server
* Code Structure
* Prop Types

1. Final Project on ReactJs
2. Redux:

* Complexity of Managing State
* What is Redux? Why Redux for React?
* Understanding Redux Flow
* Setting up Reducer and Store
* Actions and Subscriptions
* Connecting Redux to React
* Updating State Immutably
* Handling Multiple Reducers

\*how to use react snippet -

rce : it will generate class component

rconst : it will generate constructor with state

rfce : it will create function component

rafce: it will generate arrow function component