

Topics	Objectives
Introduction, Template Literals, Parameter Handling, Helpers, Functions and Enhanced Object Properties,	1. Score with const and let, 2. String interpolation 3. Default Params, Rest Operator, Spread Operator 4. forEach, map, filter, find, every, sort 5. Arrow functions, Objects, Rest and spread operator
Destructuring, Modules, Promises, Map / Set Classes	1. Property Shorthand, Computed Property Names, Method Properties 2. Array Matching, Object Matching, Deep Matching, Default values, Parameter Context Matching 3. Modules 4. Promises 5. Map/Set 6. Classes
Introduction to node, environment setup, modules, npm	What is Node? Steps to setup development environment to develop an application What is module in node? What is NPM? What is a Package? Usage of package. Installing Modules using NPM Loaders
Webpack and Babel	What is babel? Environment Setup Project setup using Babel Overview of Babel Plugins, Babel Polyfill, Babel CLI Why webpack? Webpack Config. Webpack Modules Requiring Libraries Project setup using Webpack

PRACTICE REQ FOR
REST OPERATOR

WORKING WITH PROMISE AND
MAP/SET

WEBPACK ??

Introduction to React, Virtual DOM, JSX	Whats is SPA, Role of React in MVC, create react app using npm and npx, react setup, Why we need to have react in web application, Advantages of using React over plain Javascript, JSX, React Elements, Working of Virtual DOM, React Renderer , React Reconciliation, React Keys, Explain folder structure, Styling methods, render method. Objectives: Types of JSX, React keys. Simple react problem, React elements.
React Components, (Functional and Class), rendering arrays, map, Lists, Keys, Component Identification	What is Component, Advantages of writing code as components, Functional Components , Class Components, Parent / Child components, Types of Components, Conditional Rendering, Identifying components from design, Using map to render list of components. Import Components from different packages. Objectives: Component rendering, Functional component and Class components. Structring UI into components. Rendering the components with list, map, Conditional rendering, Importing components from different place
React Props & State, Introduction to Prop types	what is a props, How props are used to communicate between components, What is the state, How to initialize and manage the state, How to change the state value using setState, Changing parent state from child, Prop types, Objectives : Understanding props, state, updating and managing state, Prop types, default prop types, Communicating parent child through props.

KEY VALUE
SETUP

CHILD-PARENT-
CHANGING VALUES
OF PARENT FROM
CHILD USING
setState -
UNDERSTANDING+P
RACTICE

Event Handling in React, Style handling (different modes)	<p>simple onclick html vs react and basic diff like cant use false have to use prevent default, SyntheticEvent intro with example max all type of events, event handlers, use this(bind vs arrow functions) with example, passing argument to event handlers, multiple event handlers,</p> <p>Ways to Style React Components: Inline CSS Normal CSS Styled Components CSS Modules Sass & SCSS Less</p>	to prevent default use <code>event.preventDefault()</code>
Life Cycle Methods & Conditional Rendering	<p>component life cycle intro , types of methods, Mounting ,Updating , Unmounting, Error Handling methods clasification with example for each.</p> <p>Conditional rendering intro and purpose with example, Inline If with Logical && Operator(ternary) and if,</p>	
Forms, Validation, Refs	Form intro react and conditional comp, basic form validation with different form fields, formik intro with example, yup validation and custom validation, reinitialize formik purpose of ref, when to use ref, create and access ref , Adding a Ref to a DOM Element, Adding a Ref to a Class Component and Function Components	
React Routing	<p>- What is routing and why we need routing ?</p> <p>- Basic Routing using links</p> <p>- Nested Routing</p>	

React Context API	What is Context When to use context API Traditional way of passing arguments What is Provider and Context Where we can use Context API What is Consumer What is Dynamic Context How to consume multiple context
Introduction to Redux	What is Redux Creating Redux Integrating Redux with React Redux Structure Building Redux Provider and Connect Redux Stores Redux Reducers Redux State Async Actions
Recap of Promises, Fetch API, Axios	-What is Promise -async, await and then in promise - Performing a GET Request, GET Request with Parameters, POST Request with axios
HOC, render props, Authentication, Cookies, JWT	1. Introduction for all Topic 2. Functional programming and higher order functions 3. Higher order component in React 4. The render props pattern 5. A simple render prop component 6. Authentication and Goals for authentication system 7. Get and Set Cookies in react 8. Understand JWTs 9. Structure of a JWT

Local Storage, Session Storage, ESLint, i18n, Lazy Loading, Error Boundaries	<ol style="list-style-type: none"> 1) What is a local store and when it can be used. 2) Set and get a value from local store and use it. 3) What is a session storage and when it can be used. 4) Set and get a value from session storage and use it. 5) What is ESLint 6) How to improve code style using ESLint. 7) How internationalization done using i18n in react. 8) What is Error Boundaries 9) Where to place Error Boundaries. 10) What is lazy loading 11) fallback, suspense and lazy in lazy loading
Introduction to Hooks, React Charts library	<ol style="list-style-type: none"> 1. Introduction about react hook 2. State hook and Stateless components 3. Declaring multiple state variables 4. Effect hook 5. Other hook 6. Important React chart libraries
Introduction to Testing, JEST	<p>Introduction to testing</p> <ol style="list-style-type: none"> 1) Definition of testing 2) Debugging vs. Testing <p>Jest</p> <ol style="list-style-type: none"> 1) Installation and testing (Javascript,react) 2) Using matchers - common matchers(expect,tobe,toEqual,not.toBe), truthiness (toBeNull,toBeUndefined,toBeDefined,toBeTruthy,toBeFalsy), Numbers,Strings, Arrays and iterables,Exceptions 3) Test asynchronous code (callback,promises,resolves/rejects,async/await) 4) Scoping (beforeAll,afterAll,beforeEach,afterEach) 5) Use mock functions

