**REACT CONTENTS**

* **Introduction To React**
* **Features of React**
* **DOM, Shadow DOM & Virtual DOM**
* **Cons of React**
* **Installation of React**
* **React using npm**
* **Introduction to React Component**
* **How to use multi components in React Application**
* **Class Component**
* **Difference between Function and Class component**
* **Data Bindings**
* **Simple, Nested Bindings**
* **Fetching Data from API and Bind**
* **How to Use State functionality and Event Binding in Functional**

**Component**

* **How to use Two Way Data Binding**
* **State, FakeStoreAPI**
* **How to Use State functionality in and Event Binding Class**

**Component**

* **Dynamic Style Change, Theme Change.**
* **Forms and Validations**
* **Using React (Lengthy Process)**
* **Using 3rd Party API/Library i.e Formik (Easy & Industrial Standard)**
* **Formik Validations. (React Forms)**
* **Data Binding & Validations using YUP Library.**
* **Formik Components.**
* **Validation State.**
* **Component Life-Cycle Hooks.**
* **For Class Component.**
* **For Functional Component.**
* **Predefined React Hook**

**\*useState()**

**\*useEffects()**

**\*useContext()**

**\*useCookies()**

**\*useReducer()**

* **Custom Hook**
* **Working with APIs (Get/Post/Patch/Delete)**
* **React Routing**
* **Intro to Routing**
* **Basic Routing Components**
* **Complete Login & Register Example using Routing**
* **Route Parameters**
* **Nested Routes**
* **Route Hooks**
* **Typescript Concepts (33-38 ½)**
* **Using typescript in React Application**
* **Material UI for React Project**
* **Redux**
* **Route Hooks**
* **Deployment (Local Server, Firebase)**
* **React Native (Generally not for web developers, it is for mobile app developers)**
* **Creating React App using Web Pack.**

**THNINGS NEEDS TO BE PRACTISE**

* **How to work with APIS, GET/POST/PATCH/PUT**

**Using Forms (Using Formik) (Refer Video No 27)**

* **Use Login and Register**
* **Perform data binding and validations. (Done)**
* **How to create Authenticate Route.** **(React\_JS\_Notes Page No: 375)**
* **Redux Example.**
* **How to show our projects on git-hub repository.(similar to git-hub desktop, we need to put the *build file* inside clone repository.**
* **Web Pack Example.**

**FAQs in REACT**

* **What is the pros of developing screen using typescript with React applications?**

**Ans. Using typescript in react application helps to develop robust, scalable & maintainable applications. It helps to reduce runtime errors because of we can handle the data type in in frontend as per the expected structure. In case of typescript errors can be noticed at the time of compilation.**

* **How to solve the error while downloading react app with typescript?**
* **npx create-react-app react-typescript-app --template typescript**

**(got error)**

* **npx create-react-app react-typescript-app --template typescript --legacy-peer-deps  
  (tsconfig.json did not get created)**
* **Manually created tsconfig.json file  
  inside this file added  
  *{***

***"compilerOptions": {***

***"target": "es5",***

***"lib": ["dom", "dom.iterable", "esnext"],***

***"allowJs": true,***

***"skipLibCheck": true,***

***"esModuleInterop": true,***

***"allowSyntheticDefaultImports": true,***

***"strict": true,***

***"forceConsistentCasingInFileNames": true,***

***"noFallthroughCasesInSwitch": true,***

***"module": "esnext",***

***"moduleResolution": "node",***

***"resolveJsonModule": true,***

***"isolatedModules": true,***

***"noEmit": true,***

***"jsx": "react-jsx"***

***},***

***"include": ["src"]***

***}***

* **Then   
  rm -rf node\_modules package-lock.json**
* **Then  
  npm install**
* **Then   
  npm install --save-dev @types/react @types/react-dom  
    
  Then at last worked fine.**
* **How copy and paste ignoring some file/folder?**

**robocopy.exe "C:\Users\bibhu\OneDrive\Desktop\About\_me\_some\_Imp\_things\13\_React JS@18\React\_Notes" "C:\Users\bibhu\OneDrive\Desktop\About\_me\_some\_Imp\_things\13\_React JS@18\React\_Notes\Notes\32 Nested Routes" /S /XD node\_modules .git Notes React-Web-App Some\_Technical\_Things\_On\_React Developed\_Components\_Inside\_Components\_folder /XF "C:\Users\bibhu\OneDrive\Desktop\About\_me\_some\_Imp\_things\13\_React JS\React\_Notes\.gitignore" React\_JS\_Notes.pdf**

* **How to change the default port number in React?**

**Create one .env file inside root directory.**

**Inside .env file  
PORT: 3001**

**Then use the command *npm start***

**When to go for React and when to Go for Angular in Java fullstack development?**

**Ans. Use React for flexible applications, custom/dynamic UIs, Smaller bundle size, better performance, small teams, Next JS. for Better SEO and SSR, product/ startup environments it is beneficial.**

**Use Angular for fully structured framework where everything included(Routing, DI, forms, testing), large enterprise apps, When frontend and backend teams are separate at that time Angular plays best role.**

**What is the use of Next.js?**

**Ans. It is used when the frontend apps needs better SEO optimization, SSR, Performance optimization.**

**\*\*Questions On REACT\*\***

**1. What is the use of Gats by. (1)**

**2. What are the challenges in modern web development and what is**

**the solution.**

**3. What are the issues with Js and DOM and what is the solution.**

**4. What is the difference between Angular & React. (2)**

**5. What is the difference between library and framework.**

**6. What is React and its Features.**

**7. What is DOM. (3)**

**8. Define the architecture & flow of browser engine.**

**9. What is the use of Rendering engine, Js interpreter, Data**

**persistence & Network.**

**10. What is Shadow DOM & Virtual DOM.**

**11. Why React used Virtual DOM.**

**12. What is Single source of truth & Reconciliation.**

**13. What are the cons of React. (4)**

**14. In how many ways we can create a React application.**

**15. What is package manager.**

**16. What is the command to install package.json file.**

**17. What are the libraries required to use React.**

**18. What is the name of JS compiler used in React.**

**19. What is the meaning of --save at the time of installing of any tool.**

**(5)**

**20. What are the module systems are available.**

**21. What is the command to create a new React App.**

**22. What is the command to run React app and what is the default**

**port number for react application.**

**23. What is a components and what it contains. (6)**

**24. How the functional and class component returns the markup.**

**25. In which file we need to import boostrap.**

**26. What are the JSX rules are there in React.**

**27. How to use Arraow functions in component. (7)**

**28. When to use normal function and when to use arrow function.**

**29. What is the benfites of using arrow function.**

**30. What is a class and what are its behaviors. (8)**

**31. What are the members of a class and define each one.**

**32. Does it mandatory to call a base class constructor.**

**33. React Class components extends from which library.**

**34. What is React and what is Component and What is a namespace.**

**35. Which is better class component or functional component.**

**36. What are the drawbacks of using class component.**

**37. What are the differences between functional and class component.(9)**

**38. What are the state management techniques are there.**

**39. Which component you will design it as a class.**

**40. What is data binding.**

**41. What are the methods that jQuery uses for data binding.**

**42. What is the data binding expression that is used for React.**

**43. What are the differences between Attribute and property.**

**44. Can we bind the data for attribute.**

**45. What are the methods are there for binding a collection.**

**46. How to bind multiple data in a list(ordered/unordered). (10)**

**47. Does React supports two-way bindings.**

**48. How to manage two-way bindings in React.**

**49. Which object is used to store the data.**

**50. Class & Function which has the default state.**

**51. How to manage state in both class and functional approach.**

**52. Which method acts as a constructor in React.**

**53. Javascript objects are managed by which object. (11)**

**54. What are the JS borwser events are there.**

**55. Is React directly use browser events. if not then it uses which event.**

**56. In React Synthetic events support how many arguments.**

**57. What is the rule for binding a value into a form element.**

**58. What is the dfiierence between useState() and useEffects(). (12)**

**59. What are the state functionality includes.**

**60. What state functionality returns.**

**61. What is the rule for declaring the state.**

**62. What useEffects() do.**

**63. When the state is maintained in class component. (13)**

**64. What is the use of props inside of a constructor.**

**65. Which method is used to set the value in state.**

**66. If our event is trying to set the value then how the event will bind into the state.**

**67. What is the use of Component properties. (14)**

**68. How to pass the data from one component to another component in both functional and class approach.**

**69. How React uses style binding and class binding.**

**70. What is AS-IS, TO-BE & GAP. (16)**

**71. What are the popular 3rd party for Forms in react.**

**72. What is initial values in Formik**

**73. What is onSubmit in Formik.**

**74. What is the use of formik.handleChange.**

**75. In Formik how the values will be accessed.**

**76. How YUP simplified the forms data and validation. (17)**

**78. What is validation schena in YUP.**

**79. Why we are using Formik Forms. (18)**

**80. What are the components provided by Formik.**

**81. What is validation state.**

**82. Formik & Yup provides how many types of validation state.**

**83. What is Form State validation and Input state validation.**

**84. What is isValid, dirty and touched.**

**85. All the Engines are managed by which library. (19)**

**86. What are the phases of component life cycle in react.**

**87. What are the methods are available in each phases and explain all.**

**88. In React component lifecycle hooks mounting and unmounting is**

**there and they are performing their actions but what the developers**

**do for that. Explain one Scenario.**

**89. What is difference between constructor and componentWillMount()**

**(20)**

**90. What is Change Detection.**

**91. Who manages change detection in react.**

**92. How it identifies the changes.**

**93. In which phase of component life cycle "Change Detection" occurs.**

**94. What is the role and responsibility of React Hooks.**

**95. What is WebSocket.**

**96. What is the rules for configuring the hooks.**

**97. What is useState and useEffect hook.**

**98. What is useContext Hook.**

**99. Give a Scenario of useContext()**

**100. What is Context memory and what is the purpose of it. (21)**

**101. Context uses which mechanism.**

**102. What is Provider and Injector.**

**103. What is Singleton pattern.**

**104. Which method is used to create context memory.**

**105. What are the steps to use context memory.**

**106. When the Context ends. (22)**

**107. What is cookie and what are its types.**

**108. What is the life span of In-Memory Cookie and Persistent Cookie.**

**109. Why cookies are used in web application.**

**110. How a developer knows the status of cookies on browser.**

**111. What is the provider name for Cookie.**

**112. What is the responsibilities of useCookies.**

**113. What is the syntax for configuring a cookie, create a cookie, cookie**

**availability and removing a cookie.**

**114. What are the cookie options are there.**

**115. If the expires is not defined then it is known as which cookie.**

**116. Why Redux. (23)**

**117. What is Reducer.**

**118. What is the benefit of Reducer over context.**

**119. How many components are in useReducer.**

**120. How many components are in Reducer.**

**121. How state cycle happens in React.**

**122. What Reducer predicts.**

**123. Dispatcher follows which pattern.**

**124. What dispatcher does.**

**124. Reducer manages the state container in which level.**

**125. What is the difference between return and break.**

**126. What is the basic rules for creating a custom hook. (24)**

**127. What are the role and responsibility of a React developer. (25)**

**128. What is Distributing computing Architecture.**

**129. What are the distributed computing technology are there.**

**130. API communication is happened in how many ways.**

**131. What Middleware does. (26)**

**132. When CORS issue raises.**

**133. Which object manages Ajax requests in Javascript based applications.**

**(27)**

**134. What is fetch() method.**

**135. What are the 3rd party APIs are there to communicating with APIs in**

**React.**

**136. What is Request Forgery.**

**137. What is Axios. (28)**

**138. Which details are contained by Axios response object.**

**139. What SEO friendly URL allows.**

**140. Which library is used for React Routing.**

**141. What is BrowserRoute.**

**138. What is Browser Router. (29)**

**139. Which browser object of JS can handle navigation.**

**140. How we can add a new page into current browsing history.**

**141. Who creates Route table.**

**142. What are the Route properties.**

**143. How to define route parameters and search params.**

**144. What is Link and what is the difference between anchor tag and Link.**

**145. What is Outlet.**

**146. How to define a default Route.**

**147. How to define a route path that is not available in route table.**

**148. How to navigate to any specific path dynamically. (30)**

**149. What is the use of route parameters. (31)**

**150. Which hook is used to access the route parameters.**

**151. Can Nested route have Route parameters. (32)**

**152. Can we have multiple Outlets in a component.**

**153. <Route path='plogin' element={<PersonalLogin />} />**

**<Route path='/plogin' element={<PersonalLogin />} />**

**Which will remove the existing and which will add to the existing path.**

**154.**