PHASE 4 TOPICS

**Front End Basic TOPICS**

1. HTML => Understand the various tags in HTML
2. CSS => to style the web pages.
3. Bootstrap => a css library used for styling
4. Javascript
   1. Dialog boxes
      1. Alert
      2. Prompt
      3. Comfirm
   2. Variable : vat , let and const
   3. Operators
   4. == and ===
   5. Conditional constructs : if-else and switch-case
   6. Loops : for , while and do-while
   7. Arrays :
   8. Array methods
   9. String methods
   10. Functions
       1. Named
       2. Function expressions
       3. IIIFE
       4. Arrow
   11. ES6 Syntax
       1. Spread
       2. Rest
       3. Destructuring syntax
       4. Default arguments in functions
       5. Immutable and Mutable state of javascript properties
       6. Class
       7. Promise and AJAX
       8. Fetch for REST API

**REACT TOPICS**

1. Download VSCode
2. Download NodeJS
3. Using create-react-app, create a react project
4. Components in React : Reusable JSX functions or class that consists of View and BL
   1. Functional Based
   2. Class Based
5. Props : They are read only and passed as parameters to the components
   1. Class Based : this.props
   2. Functional : props (as they are passed as parameters)
6. State : They maintain the state of component within the class or function. They are like instance variables local to the component
   1. Class : this.state and define in the constructor   
      To modify the state value use setState()
   2. Functional : useState  
      To modify use the setter
7. Conditional Rendering :
   1. If
   2. Short circuit operator &&
8. Lifecylce of the component
   1. Class Based
      1. ComponentDidMount: called once at the start of the lifecycle
      2. shouldComponentUpdate : called anytime the props or state value changes and only if it returns true will the UI be updated
      3. componentDidUpdate : if shouldComponentUpdate returns true then this method is called and component is updated
      4. componentWillUnMount : Called once the lifecycle when component is removed from the DOM hierarchy
   2. Functional Based:
      1. useEffect(()=>{}) : With only 1 parameter is same as all lifecycle methods in class based and will be called again and again
      2. useEffect(()=>{}) : With 2 parameters and 2nd parameter as empty array is same as componentDidMount and will be called only once
      3. useEffect(()=>{}) : With 2 parameter 2 and 2nd parameter with a prop or state name in array is same as componentdidmouint and for the changes in array variable is same as shouldComponentUpdate and componentDidUpdate
      4. useEffect(()=>{}) => If first parameter returns a function then that is componenetWillUnmount
9. Form Handling:
   1. React controlled form components by associating the value attribute with state of the component and handle onChange event for updating the state
   2. React uncontrolled form components by not associating the value attribute with state of the component and not to handle onChange event for updating the state