REACT JS – L1: Trend.Nxt Hands-on Assignments

Estimated Efforts: 3 PDs

For detailed ToC and other Details: https://wipro365.sharepoint.com/sites/ku-practice-4043/CCX-Competency/TREND.NxT/Pages/React-JS-L1.aspx

Author: sridhar.savitha@wipro.com

Date: 25th May 2018

TOC:

Topic No	Topic Name	Sub Topics	Min No of Assignments to be Done
1	React JS - Overview	What is React JS, Features of React JS, Advantages and Limitations of React JS	0
2	React JS - Environment	Setting up the React JS Environment	1
3	React JS - JSX	Usage of JSX in React JS	3
4	React JS - Components	Developing Stateless and Stateful Components	1
5	React JS - State	Introduction to React JS State	
6	React JS - Props	Introduction to React JS Props	1
7	React JS - Props Validation	Validating the props	1
8	React JS - Component API	Introduction to React Component API and the methods of the component API	1
9	React JS - Component Life Cycle	Introduction to the Component Lifecycle methods	1
10	React JS - Forms	Introduction to the usage of forms in React JS	1
11	React JS - Events	Event Handling in React JS	
12	React JS Router	Working with React JS Router	1
13	React JS Refs	Working with Refs	0

Total Min No of Assignments to be Done	11

Topic 2: React JS - Environment

Assignment 1:

Create a React Program that would render just one greeting message - "**Happy Learning-React**"

Topic 3: React JS - JSX

Assignment 1:

Create a React Program that would render a Heading and also a Table with 5 employee details. The employee details to include are Employee ID, Employee Name and Employee Email ID

Assignment 2:

Update the solution of Assignment-1 to style your table.
Styling recommendations: Table should have a dashed border of green color and the Table heading columns should have a background color as cyan.

Assignment 3:

Create a React Program that would render a Heading and also a Table which shows the multiplication tables of number 5.

(Hint: a. 5 X 1 = 5, 5 X 2 = 10 .. Do not hard code the values. Generate them by performing mathematical operations
        b. Use JS expressions for performing mathematical operations
)

Topic 4: React JS – Components & Topics 5: React JS - State

Assignment 1:

Create a React Program based on the below requirements:

a. The main component is the "Main" component that stores the state information about 5 students like studentId, studentName and studentMarks.
b. Another component called "Header" renders a heading as "Student Details"
c. Another component called "Details" displays all the student details provided by the "Main" component in the form of a table.
d. Finally the main component needs to take care of rendering both the Header and the Details.
e. Use ECMAScript => syntax wherever needed.
f. You can apply appropriate styling to Heading and Details table as well

Topic 6: React JS – Props

Assignment 1:

Create a React program based on the below requirements:
a. The main program passes two immutable variables to the "App" component which are 'companyName' and 'companyLocation' and these values gets rendered there in the App component.
b. If the main program does not pass these values, then the default values to use for these are: 'companyName' = 'WIPRO' and 'companyLocation' = 'BANGALORE'.

Topic 7: React JS - Props Validation

Assignment 1:

Create a React program based on the below requirements:
a. The main program passes the below immutable variables to the "App" component which are Name, Preferred cities and Age.
b. We need to also apply the below validations to the above passed values:
    i.   Name : Should be string type and is mandatory
    ii. Preferred cities: Should be an array type
    iii. Age: Should be a number and should be greater than or equal to 18 and less than or equal to 60.
c. In case the values do not get passed, then we should use the default values as: Name: Steve, Preferred cities: [Bangalore, Chennai] and Age: 18.
d. While testing and executing this app, try to pass values which do not cater to the required validation rules and see how you get warnings in the console (capture these console outputs and submit during assignment submission)

Topic 8: React JS - Component API

Assignment 1:

Create a React program based on the below requirements:
    a. The web page should show a button 'Click to generate Multiplication tables of 5'. When we click on the button, it should start rendering the multiplication tables of 5. Every click should generate one row of the multiplication table.
    b. So initially the Component in its state maintains an empty array and whenever the click happens, the array is updated with one line of the multiplication table which in turn gets rendered.

Topic 9: React JS - Component Life Cycle

Assignment 1:

Create a React program based on the below requirements:
    a. The React program displays/renders a Div and a paragraph within it which displays a nice thought for the day.
    b. When we click on the "Div" , it should get removed (unmounted).
    c. Implement all the Component Life Cycle methods like 'ComponentWillMount' , 'ComponentDidMount' and 'ComponentWillUnmount' etc to view the order in which these life cycle methods get called.

Topic 10: React JS – Forms & Topic 11: React JS - Events

Create a React program based on the below requirements:
    a. The React program renders a form so that the user can enter his/her individual subject marks for the subjects: Physics, Chemistry, Biology and Mathematics.
    b. Upon entering of all the details and on click of the button "Find Average", the average of those marks should be displayed as an alert.

Topic 12: React JS Router

Create a React program based on the below requirements:

    a. The React program should support "Movie Details Management".
    b. Movie details will hold details like MovieID, MovieName, LeadActor, LeadActree,YearOfRelease,Language,Collections
    c. The program should support the below routes/tasks/operations
        a. / → should display all existing movie details.
        b. /MovieID → should display details only about a particular movie
        c. /addMovie -> should provide a form to enter a new movie details and add it to the existing collection

d. We should also be able to delete an existing movie detail based on its MovieID

Note: Wherever applicable, capture the screen shots of all the outputs and use it during your assignment submission.