ReactJS Assignments - Detailed Output

# 1. myfirstreact

Objectives

Define SPA and its benefits

Define React and identify its working

Identify the differences between SPA and MPA

Explain Pros & Cons of Single-Page Application

Explain about React

Define virtual DOM

Explain Features of React

In this hands-on lab, you will learn how to:

Set up a react environment

Use create-react-app

Prerequisites

The following is required to complete this hands-on lab:

Node.js

NPM

Visual Studio Code

Notes

Estimated time to complete this lab: 30 minutes.

Create a new React Application with the name “myfirstreact”, Run the application to print “welcome to the first session of React” as heading of that page.

To create a new React app, Install Nodejs and Npm from the following link:

Install Create-react-app by running the following command in the command prompt:

To create a React Application with the name of “myfirstreact”, type the following command:

Once the App is created, navigate into the folder of myfirstreact by typing the following command:

Open the folder of myfirstreact in Visual Studio Code

Open the App.js file in Src Folder of myfirstreact

Remove the current content of “App.js”

Replace it with the following:

Run the following command to execute the React application:

Open a new browser window and type “localhost:3000” in the address bar

# 2. StudentApp

Objectives

Explain React components

Identify the differences between components and JavaScript functions

Identify the types of components

Explain class component

Explain function component

Define component constructor

Define render() function

In this hands-on lab, you will learn how to:

Create a class component

Create multiple components

Render a component

Prerequisites

The following is required to complete this hands-on lab:

Node.js

NPM

Visual Studio Code

Notes

Estimated time to complete this lab: 30 minutes.

Create a react app for Student Management Portal named StudentApp and create a component named Home which will display the Message “Welcome to the Home page of Student Management Portal”. Create another component named About and display the Message “Welcome to the About page of the Student Management Portal”. Create a third component named Contact and display the Message “Welcome to the Contact page of the Student Management Portal”. Call all the three components.

Create a React project named “StudentApp” type the following command in terminal of Visual studio:

Create a new folder under Src folder with the name “Components”. Add a new file named “Home.js”

Type the following code in Home.js

Under Src folder add another file named “About.js”

Repeat the same steps for Creating “About” and “Contact” component by adding a new file as ”About.js”, “Contact.js” under “Src” folder and edit the code as mentioned for “Home” Component.

Edit the App.js to invoke the Home, About and Contact component as follows:

In command Prompt, navigate into StudentApp and execute the code by typing the following command:

Open browser and type “localhost:3000” in the address bar:

# 3. scorecalculatorapp

Objectives

Explain React components

Identify the differences between components and JavaScript functions

Identify the types of components

Explain class component

Explain function component

Define component constructor

Define render() function

In this hands-on lab, you will learn how to:

Create a function component

Apply style to components

Render a component

Prerequisites

The following is required to complete this hands-on lab:

Node.js

NPM

Visual Studio Code

Notes

Estimated time to complete this lab: 30 minutes.

Create a react app for Student Management Portal named scorecalculatorapp and create a function component named “CalculateScore” which will accept Name, School, Total and goal in order to calculate the average score of a student and display the same.

Create a React project named “scorecalculatorapp” type the following command in terminal of Visual studio:

Create a new folder under Src folder with the name “Components”. Add a new file named “CalculateScore.js”

Type the following code in CalculateScore.js

Create a Folder named Stylesheets and add a file named “mystyle.css” in order to add some styles to the components:

Edit the App.js to invoke the CalculateScore functional component as follows:

In command Prompt, navigate into scorecalculatorapp and execute the code by typing the following command:

Open browser and type “localhost:3000” in the address bar:

# 4. blogapp

Objectives

Explain the need and Benefits of component life cycle

Identify various life cycle hook methods

List the sequence of steps in rendering a component

In this hands-on lab, you will learn how to:

Implement componentDidMount() hook

Implementing componentDidCatch() life cycle hook.

Prerequisites

The following is required to complete this hands-on lab:

Node.js

NPM

Visual Studio Code

Notes

Estimated time to complete this lab: 60 minutes.

Create a new react application using create-react-app tool with the name as “blogapp”

Open the application using VS Code

Create a new file named as Post.js in src folder with following properties

Figure 2: Post class

Create a new class based component named as Posts inside Posts.js file

Figure 3: Posts Component

Initialize the component with a list of Post in state of the component using the constructor

Create a new method in component with the name as loadPosts() which will be responsible for using Fetch API and assign it to the component state created earlier. To get the posts use the url ()

Figure 4: loadPosts() method

Implement the componentDidMount() hook to make calls to loadPosts() which will fetch the posts

Figure 5: componentDidMount() hook

Implement the render() which will display the title and post of posts in html page using heading and paragraphs respectively.

Figure 6: render() method

Define a componentDidCatch() method which will be responsible for displaying any error happing in the component as alert messages.

Figure 7: componentDidCatch() hook

Add the Posts component to App component.

Build and Run the application using npm start command.

# 5. cohortdashboard

Objectives

Understanding the need for styling react component

Working with CSS Module and inline styles

In this hands-on lab, you will learn how to:

Style a react component

Define styles using the CSS Module

Apply styles to components using className and style properties

Prerequisites

The following is required to complete this hands-on lab:

Node.js

NPM

Visual Studio Code

Notes

Estimated time to complete this lab: 30 minutes.

My Academy team at Cognizant want to create a dashboard containing the details of ongoing and completed cohorts. A react application is created which displays the detail of the cohorts using react component. You are assigned the task of styling these react components.

Download and build the attached react application.

Unzip the react application in a folder

Open command prompt and switch to the react application folder

Restore the node packages using the following commands

Figure 1: Restore packages

Open the application using VS Code

Create a new CSS Module in a file called “CohortDetails.module.css”

Define a css class with the name as “box” with following properties

Width = 300px;

Display = inline block;

Overall 10px margin

Top and bottom padding as 10px

Left and right padding as 20px

1 px border in black color

A border radius of 10px

Define a css style for html <dt> element using tag selector. Set the font weight to 500.

Open the cohort details component and import the CSS Module

Apply the box class to the container div

Define the style for <h3> element to use “green” color font when cohort status is “ongoing” and “blue” color in all other scenarios.

Final result should look similar to the below image

Figure 2: Final Result