R0333 - Project Web App Development with a Javascript Framework

React.js: Workshop 1 - Basics

Exercise 1 - Setup the environment

Open the Node.js command prompt. Run the npx command to set up local server environment for Node. This might take a while.

```
$ npx create-react-app workshop1
$ cd workshop1
$ npm start
```

Now you have a local React.js development environment set up and server running at http://localhost:3000. See whats there!

Exercise 2 - Studying the files

1. Open newly created folder workshop1 with the editor of your choice.
2. Open two files: **public/index.html** and **src/App.js**. Study them a bit and try to see whats going on. Compare the code to what you see in the Browser window.
3. Try to add text in index.html. See the results in browser.
4. Change the page title and intro text from App.js. Save the file and see the output in browser. Notice that the the server updates automatically.
5. Now you can clean up the files a bit: remove file ***App.js***, ***App.css***, ***App.test.js***, ***logo.svg*** and ***registerServiceWorker.js.***

Exercise 3 - Writing your first component

1. Open the file index.js
2. Add the following code snippets there and see the output in the browser

```
import React from "react";
        import ReactDOM from "react-dom";

        const Greetings = () => {
        console.log("Hello World");

        return (
        <div>
          <p>Hello world</p>
        </div>
        );
        };

        ReactDOM.render(
        <Greetings />
        , document.getElementById("root"));
```

Exercise 4 - Writing another component with styles

1. Create another function called Quote
2. Place it after the Greetings-function in index.js
3. The function should return a div element with an id="myStyle". Come up with a nice quote for the contents.
4. Define myStyle CSS in App.css. You can create your own style, but here is a sample to start with:

```
#myStyle {
        border: 2px solid black;
        text-align: center;
        background: #f5f5f5;
```

```
color: #333;
margin: 10px;
padding: 10px;
}
```

5. Finally add the statement import "./App.css"; in the beginning of the index.js file
6. See the output
7. Add a new to index.html and render the output there.

Exercise 5 - Nested components

1. Create another function called ManyQuotes
2. Place it after the Quote-function in index.js
3. Function should return a element, which has 5 <Quote /> components inside.
4. See the output
5. Add a new to index.html and render the output there.

Exercise 6 - Passing values to components with props

1. Create another function called CustomQuote
2. Function should return a element, which returns a quote defined as props in JSX tag
3. Render the function using the tag <CustomQuote quote="Make love, not war." />
4. See the output
5. Add id and CSS styles for the elements
6. Add a new to index.html and render the output there.
7. Render 3 different quotes using CutomQuote-component
8. Add another attribute "author" to the CustomQuote -tag, which will define author the the quote.
9. Try it using the render methdod with <CustomQuote quote="Make love, not war." author="John Lennon" />

Exercise 7 - Passing JSON data to components

1. Create another function called QuoteArray
2. Function should return a element, which return a list of quotes passed in as JavaScript array
3. Output every quote between <li> tags
4. Add id and CSS styles for the elements
5. Add a new to index.html and render the output there.
6. You can use sample JSON dataset as below.

```
const quotes = [
{
quote: "Life isn’t about getting and having, it’s about giving and being.",
author: "Kevin Kruse"
},
{
quote: "Whatever the mind of man can conceive and believe, it can achieve.",
author: "Napoleon Hill"
},
{
quote: "Strive not to be a success, but rather to be of value.",
author: "Albert Einstein"
},
{
quote:
"Two roads diverged in a wood, and I—I took the one less traveled by, And that has made all the difference.",
author: "Robert Frost"
},
{
quote: "I attribute my success to this: I never gave or took any excuse.",
author: "Florence Nightingale"
}
];
```

7. Render 1st, 3rd and 7th quotes from the table
8. When everything works, output the author after every quote.

Exercise 8 - Looping through the JSON data

1. Create another function called QuoteArrayAll, you can copy the code from ex 6 as template
2. Refactor the code so that it will loop through the array and output all the content as an HTML list.
3. HINT: Use the map-function in return statement to go through the data.

   Try the map function in console first:

```
  // Define quote data
  const quotes = [
  {
  quote: "Life isn’t about getting and having, it’s about giving and being.",
  author: "Kevin Kruse"
  },
  {
  quote: "Whatever the mind of man can conceive and believe, it can achieve.",
  author: "Napoleon Hill"
  }

  ];
// Loop through the data using map function, for each item, execute and return the item.author.toUpperCase() function
quotes.map( item => (
               item.author.toUpperCase()
    )
 );
```

4. Add nice styles to the app.
