

CMPS 356 – Fall 2022

Web Applications Design and Development

Lab 02

React Basics

Objective

- Practice using React basic features to build a web app
- Use JSX to define the UI
- Create and render components
- Handle component state and events
- Use common react hooks particularly useState and useEffect

Prerequisites

- React app creation utility: <https://create-react-app.dev>
- React tutorials: <https://reactjs.org/tutorial> and <https://beta.reactjs.org/learn>
- React Developer Tools browser extension: <https://blog.openreplay.com/an-introduction-to-react-dev-tools>
- Visual Studio Code: <https://code.visualstudio.com>

1. Hello, React!

- Under your private lab repository, create a directory named Lab02-React-Basics.
- Create a folder 01-hello-react under Lab02-React-Basics. Then open this folder in Visual Studio Code.
- Add hello.html inside it and create a Hello component that returns 'Hello React!'. Render the Hello component inside the root div. Your page content will be as shown below:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello React!</title>
    <script
src="https://unpkg.com/react@18.2.0/umd/react.production.min.js"></script>
    <script src="https://unpkg.com/react-dom@18.2.0/umd/react-
dom.production.min.js"></script>
    <script src="https://unpkg.com/babel-
standalone@6.26.0/babel.min.js"></script>
    <script type="text/babel">
      const Hello = () => <h1>Hello React!</h1>;
      const rootElement = document.querySelector("#root");
      ReactDOM.render(<Hello />, rootElement);
    </script>
  </head>
  <body>
    <div id="root"></div>
  </body>
</html>
```

- Extend Hello to accept a name parameter.

- Redo the above example by creating a new React app using `npx create-react-app`.
- Explore the files and directories that are generated.
- Create `HelloReact` component that returns “Hello, React!”
- Display the `HelloReact` component in `index.js`.
- Render an ordered list of a few elements using JSX.
- Create a `User` component with first name and last name properties.
- Render two `User` instances.

First Name: John
 Last Name: Doe
 First Name: Jane
 Last Name: Doe

2. A Clock with a Difference

- Create a React application using the `create-react-app` utility.
- Design a `Clock` component that displays the current time. Use a state variable to keep track of the time.
- Update your component to keep the rendered time in sync with the current time
- Augment your component with a time difference (in hours) property and an associated state variable. This difference is used to shift the time by that number of hours., similarly to a time-zone difference.
- Add two buttons to increment and decrement, respectively, the time difference (in hours) by ½-hour steps.
- Log a message to the console with the current value every time the date or time difference are updated, respectively.

8/30/2022, 4:13:09 PM



2-hour difference.

3. Country Facts app

- Explore and run the Country Facts app provided under `base/country-facts` that was developed using plain JavaScript.
- Recreate the app using React and use at least two components.
- Use `useEffect` hook to fetch the list of continents when the application is first loaded.

4. ToDo app

- Explore and run the ToDo app provided under `base/todo-list` that was developed using plain JavaScript.
- Recreate the ToDo app using React and use at least two components.
- Use `useEffect` hook to load the list of todos from local storage when the application is first loaded.