CPSC 2600 Written Response for Assignment 2

Jay Seung Yeon Lee

100 357 736

1. *create-react-app gives us an initial file structure for our project. Briefly go over how it is structured.*

npx create-react-app app-name creates an initial application with src folder, public folder, node\_modules folder and package/package-lock.json file.

Src folder is the folder for the source codes where most of our work will be put in to. Source codes should consist of javascript (JSX) files for components and css files to style. With an exception of possibly editing index.html file from public folder, all work should be done in src folder.

When react application is initialized via create-react-app, public folder should consist index.html, icons, logos, and manifest.json file. Index.html file is crucial for starting react app since it is the only html code responsible for displaying all the components in a div called “root”

Node\_modules folder and package.json file share some similarities at a glance, but node\_modules folder contains all the dependencies packages for react app and package.json file provides information to the npm that allows identifying the project and it’s dependencies.

1. *Briefly go over the component - what is its purpose and how is it being rendered on the screen.*

App component is the main component that is rendered on the screen through index.html’s “root” div. App component is a component that will be ultimately call / render all other components before it gets rendered to root div.

1. *In Task 4, you added some interactivity to your todo-list app. For the following operation, walk through how the React app executes the operation (Hint: try breaking the process down into steps)*

First, Todo.js component replaced li, label, and button element that was initially hard coded in App.js. In order to make the component dynamic, we used useState hook to store new Todo component and refresh the page to display newly added component. map() method was used to add new components to existing array of object literals when adding new components.

For deleting components, we used useState hook to update the object literal array and filter() method to remove component with matching id from the list and used setTask (useState) to update displayed components.