**Wednesday May 1st, 2019**

0:02 — It’s a good thing I got a morning coding session in yesterday. I spend the rest of the day with the girl I’m currently dating. She has been staying with me the last several days. She’s a doll, and an amazing ball of happiness.

With that being said, I can’t even begin to explain how much harder having her constantly around has made it to get quality coding time. Now she’s sleeping and I’m exhausted, but it seems the only time I can get my pomodoro sessions in is when she’s sleeping hahaha.

In all seriousness, while I haven’t been coding as much as I would’ve liked to the last few days, I’ve only had 1 weekend long vacation in the last 14 months. Maybe taking it easy the last few days isn’t the end of the world… Especially considering I’ve still been coding for at least an hour everyday.

Anyway, this isn’t a diary. It’s a coding journal. Let’s get to the code!

0:18 — Alright I’m totally spent. I just spent some time giving myself an initial exposure to parent and children elements. I’m not sure how much I retained being this tired, however, so I’ll review by starting in the same place tomorrow.

12:14 — I’m back.

12:18 — In the context of React, component names use capital letters whereas plain JSX elements use lower case letters that typically resemble plain HTML.

12:32 — I’ve spent the last 15 or so minutes creating an assortment of parent and child elements. One file exports to another file which imports that file and exports itself to another file which imports it and so on. Here’s an incomplete sample of my code (it’s hard to show everything), but this should give you an idea of what I’ve been working on:

import React from "react"

import Footer from "./Footer"

import Header from "./Header"

import MainContent from "./MainContent"

function App() {

return (

<div>

<Header />

<MainContent />

<Footer />

</div>

)

}

export default App

import React from "react"

import ReactDOM from "react-dom"

import App from "./App"

ReactDOM.render(<App />, document.getElementById("root"))

12:34 — As you can see, I have my main.js file import and render App. App is also importing from multiple sources including Header.js, MainContent.js, and Footer.js. What’s the point in structuring things in this manner? I would presume cleaner and more maintainable code.

12:50 — It turns out that the next exercise in the course was doing what I basically just did, only from scratch.

12:53 — Well, I did it. I built a barebones React app from scratch. In the process I internalized a few concepts including:

— Remembering to nest all of my JSX elements in a single parent or container.

— The importance of importing and exporting all of the files correctly.

— Remembering to manually add the individual component tags into the App function so that they would be returned for rendering in the index.js file.

Forgetting that last step wouldn’t result in an error, but it would obviously prevent anything forgotten from being properly rendered and displayed.

13:01 — The instructor’s solution to the problem was basically the same as my solution. The only real difference was that he created a separate components folder to house all of his individual files.

13:02 — My attention is starting to wane. I’m going to take a break, and come back later in the day for another session or two. When I come back, I’ll start building a basic todo app in React.

13:21 — Guess who’s back…back… back… back again? Cameron’s back… 2 build comp-on-ents!

13:29 — Unexpected interruption. I’ll continue building the todo app later.

21:45 — I’m back. Let’s get to building that todo app.

21:52 — I’m quickly reviewing the HTML syntax for checkboxes.

22:00 — Here’s a quick look at what I just coded:

index.js:

import React from "react"

import ReactDOM from "react-dom"

import App from "./App"

ReactDOM.render(<App/>, document.getElementById("root"))

App.js:

import React from "react"

function App () {

return (

<div>

<form>

<input type="checkbox" name="task1" /> Task 1

<br />

<input type="checkbox" name="task2" /> Task 2

<br />

<input type="checkbox" name="task3" /> Task 3

<br />

<input type="checkbox" name="task4" /> Task 4

</form>

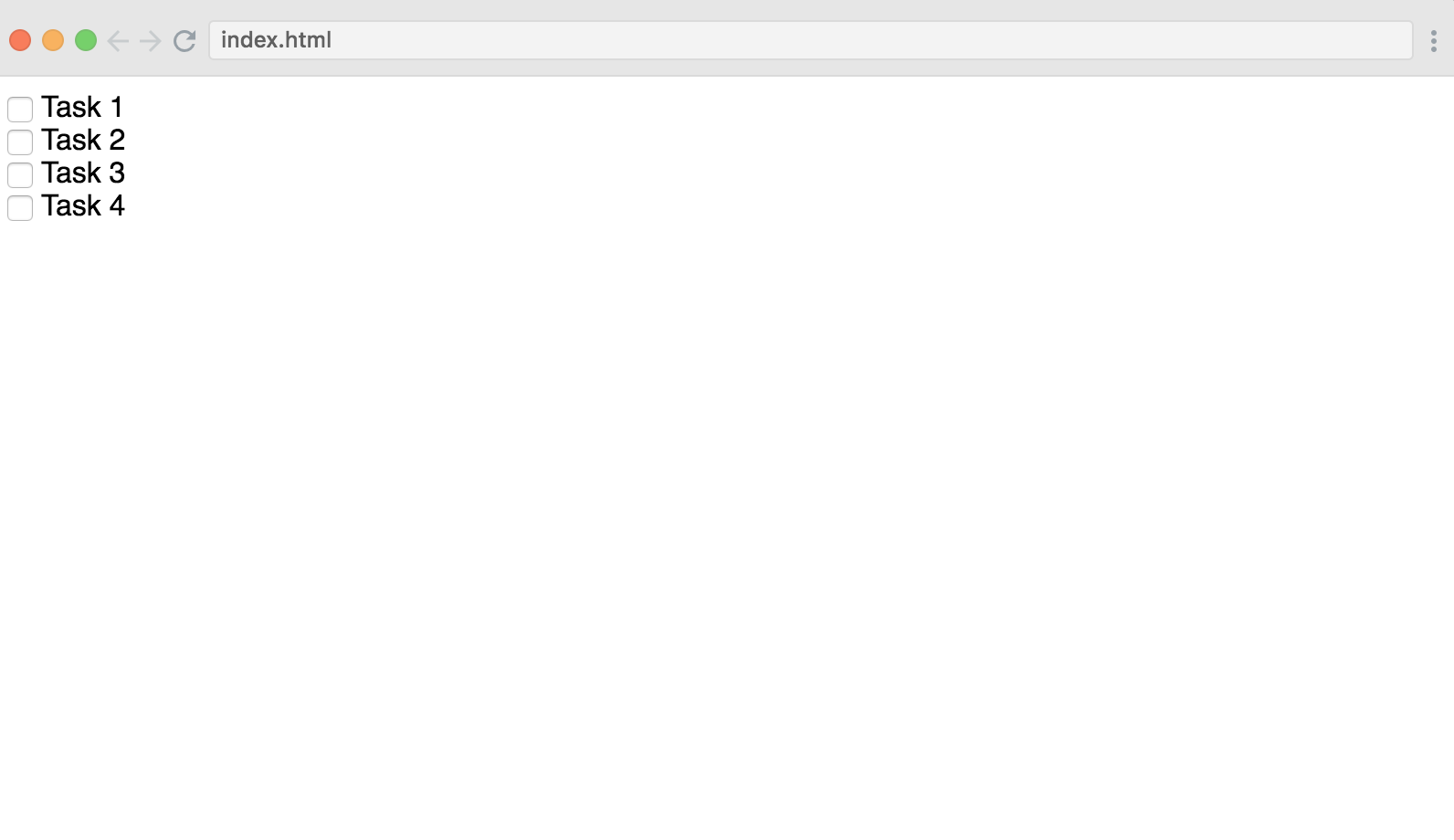
</div>

)

}

export default App

I also used an existing index.html file that was included with the course. Apart from the index.html file, I coded everything else myself from scratch. Here’s the end result.

****

22:03 — Hey, I didn’t say it was anything special ;) With that being said, I do quickly find myself becoming more comfortable with the most basic foundations in React, however.

22:05 — The instructor’s solution ended up being pretty similar to mine.

22:06 — Now I’m going to get started with learning the basics of how to style things using CSS in the context of a React application.

22:09 — Normally, in HTML we’d give an element a class by saying something like:

<h1 class = “title”>Insert your title here </h1>

One thing we have to remember when working in React, however, is use className in place of class because we’re working in JSX and not HTML.

22:18 — I’m now comfortable with the very basics of giving class names, and accessing those class names with CSS to do simple styling in React.

22:20 — The author discussed some caveats about his course such as his decision to mostly drop semi colons in JavaScript. He also mentioned that in most situations he preferred to use traditionally defined functions rather than arrow functions, but that he understands that other devs may have differing preferences.

22:22 — I’m going to call it a day here. Tomorrow I’ll start learning about how JavaScript and JSX interact in the 14th video of the course “JSX to JavaScript and Back”

**Total time spent coding today: 2 hours 4 minutes**

**Total time spent coding in May 2019: 2 hours 4 minutes.**