Part 0 & Part 1

Introduction

This is the first biweekly report for my self-guided study into the FullStackOpen under the supervision of Dr. Dianxiang Xu. Every two modules (specifically, the course equivalent of roughly two weeks), I will release a report of what I've learned, along with links to the deliverables of work completed during those two weeks.

All documentation and source code will be posted in the GitHub repository here. Deliverables can be found within the appropriately named section with links to the commit/diff of the Git repo. Some exercises refactor or change the code from previous exercises; in this event, it's best to view the git repository at the specified commit rather than the latest commit.

I will also detail all supplementary material studied that isn't in the course. Optional material and exercises will be completed, if viable to the course objectives.

This biweekly module consisted of the introduction to the course, a refresher of the basic, and introduction to using the ReactJS frontend framework. Two of the four course objectives were progressed. The self-evaluation of my accomplishment of the above course objectives can be found below.

Self Evaluation:	Mastery / Progressing / Needs Improvement
Due date:	2021-09-03
Completion:	2021-08-25
GitHub:	https://github.com/ZDGharst/FullStackOpen

Part 0 – Fundamentals of Web apps

Part 0a is course info: prerequisites, course materials, and info for those who wish to be certified in the course (I won't be doing this, as I'm focusing on myself and UMKC course credits). No knowledge of JavaScript or the other topics in this course is required. Another part of 0a talks about getting the developing tools set up. I've installed (or already had installed): Chrome, Firefox, git, VSCode, and NodeJS (npm/npx with it). I will be developing on Ubuntu 20.04 with the terminal for most tasks rather than GUI tools as this is my preferred developer environment; however, I have a laptop with Windows 10 that I may use if I'm not able to use my desktop PC for some reason.

Part 0b is where course material begins. This part has introductory material explaining web technology: HTTP requests and methods, using developer tools within the browser, and how a web server returns data requested by the user. This part also illustrates several pieces of code, however, it's not expected for the learner to understand every piece of it yet.

For me, part 0 is mostly a refresher of the basics of web technologies which I was already well-versed in. It's important to have a good grasp on these concepts, however, as knowing correct terminology and design is vital in today's programming in cooperative environments, especially with how intensive software engineering technical interviews are today.

Exercises 0.1 through 0.3 were instructions to review HTML, CSS, and web forms through Mozilla documentation with no deliverables. The charts for exercises 0.4 through 0.6 can be found in the deliverables section below.

Part 1 – Introduction to React

Part1a is the Introduction to React. It introduces us to a tool called create-react-app, which creates a new project, sets up the directory structure, initializes a git repo, configures the React project for various tasks (this is learned in part 7), creates an App component, and scaffolds other code that may be needed. Some of the things learned in this section: creating React components, using JSX to render elements to the DOM, using the JavaScript console log, and other JavaScript design patterns. One thing I do in this part as part of the curriculum is to divide the App component into separate components, and how to pass data between components using props. Exercises 1.1 and 1.2 were part of this section; I

created a new project, took existing code that is a single, large component, and refactored it into smaller, more manageable components.

The goal of *Part1b* is to describe the history of JavaScript (specifically, the official standard named ECMAScript), common uses of JavaScript, and the syntax. Most of this section covers programming paradigms and syntax for JavaScript. Exercise 1.3 instructs me to refactor the same project from 1.1 & 1.2 by changing the data from many, singleton variables into individual objects. Exercise 1.4 takes each individual object and turns it into an array of objects. Exercise 1.5 takes the array from 1.4, and creates an enclosed object with metadata. Within each exercise, I also have to make the code still work as expected with that required data structure. Finally, *Part1b* finishes with optional, supplementary material: a review of why the course avoids the "this" keyword, class syntax, and JavaScript materials.

Part1c returns to working with React. Here the class teaches how to use component helper functions, object/array destructuring, and page re-rendering. In order to re-render pages efficiently, I learned how to use a React hook to add state to a component. A stateful component will be re-rendered when its state is changed. An example that I worked through is to add event handlers to buttons, then those event handles have function references that change the state of a component. Lastly, I refactored the app by separating the component into smaller, reusable components. There were no exercises in *Part1c*.

Part1d introduces working with additional, more complex states. While I could use already learned convention to do this, it shows best practices, such as how to properly work with array and object states (such as using the object spread syntax). Old style (pre React 16.8.0) class React components are addressed, and the course explains why it's important to cover it even if no new code should be written with class components. Another large part of this section is debugging the React application: console logging, breakpoints, etc. As a reminder, the course covers hook rules, event handling, reference pointers one last time, and avoiding component nesting one last time. There are a multitude of exercises in this section, which are included in the deliverables section.

Additional notes can be found in the Part 1 README.

Deliverables

Exercise	Commit / Link	Diff	Notes
0.4	link		
0.5	link		
0.6	link		
1.1	965cd7c	diff	
1.2	a7b1d7f	diff	
1.3	9955879	diff	
1.4	bc5faa0	diff	
1.5	f96eb5b	diff	
1.6	545f04e	diff	screenshot
1.7	36b1711	diff	screenshot
1.8	ce307a2	diff	
1.9	0d18c82	diff	screenshot
1.10	This exercise instructed me to refactor the application such that the buttons and individual statistics are their own, reusable components, but I already completed this when I implemented them originally, as I felt it would be best practice.		
1.11	8a349b9	diff	screenshot
1.12	c2e3616	diff	screenshot
1.13	3407eaf	diff	screenshot
1.14	eb9448e	diff	screenshot

Supplementary Material

Learn React in 30 Minutes – To learn from a secondary perspective to make sure all important topics are covered.