



CP 325.8: Refactoring to React

Version 1.0, 08/14/23

[Click here to open in a separate window.](#)

Introduction

This document will challenge you with taking the next steps in your Capstone Project journey.

Assignment Objective

- Refactor your existing application to utilize the React framework.

Submission

This portion of the Capstone Project will not be submitted on its own; only the final project will be submitted. You are encouraged to seek feedback from instructors and peers along the way.

Instructions

React, as you have discovered, is *powerful*. This is one of the reasons that it is one of the most popular front-end frameworks in modern development.

Hopefully, the features React offers have sparked some additional ideas for your Capstone Project, because now you will be given the opportunity to convert your existing HTML, CSS, and JavaScript into a fully-featured React application. The only thing that is likely to remain exactly the same is your CSS (which will probably also change a bit), so embrace the upgrade!

This does not need to be done all in one sitting; take the time to do it right! Along the way, think about the new tools you have at your disposal for user interaction, state management, API communications, page routing, and more. What can you change? What can you upgrade? What might you want to get rid of entirely?

This is effectively the final step in your Capstone Project journey, though you still have a decent amount of time before the project is due. You may discover new tools on your own, or learn new perspectives from the remaining lessons in class, but you now have every piece of the MERN stack available to you.

Time to put your fingers to the keyboard and work toward completion!

Partial Requirements

Let's look at which Capstone Project requirements are related to this stage of development. Requirements not related to this stage of development have been omitted. For a full list of requirements, see CP 323.1 - Planning a Project, or CP 323.10 - Capstone Completion.

(20%) Project Structure, Standardization, and Convention	Weight
Project is organized into appropriate files and directories, following best practices.	2%
Project contains an appropriate level of comments.	2%
Project is pushed to GitHub, and contains a README file that documents the project, including an overall description of the project.	5%
Standard naming conventions are used throughout the project.	2%
Ensure that the program runs without errors (comment out things that do not work, and explain your blockers - you can still receive partial credit).	4%
Level of effort displayed in creativity, presentation, and user experience.	5%

(12%) Core JavaScript	Weight
Demonstrate proper usage of ES6 syntax and tools.	2%
Use functions and classes to adhere to the DRY principle.	2%
Use Promises and async/await, where appropriate.	2%
Use Axios or fetch to retrieve data from an API.	2%
Use sound programming logic throughout the application.	2%
Use appropriate exception handling.	2%

(9%) Database	Weight
Use MongoDB to create a database for your application.	5%
Apply appropriate indexes to your database collections.	2%
Create reasonable schemas for your data by following data modeling best practices.	2%

(19%) Server	Weight
Create a RESTful API using Node and Express. * For the purposes of this project, you may forgo the HATEOAS aspect of REST APIs.	7%
Include API routes for all four CRUD operations.	5%
Utilize the native MongoDB driver or Mongoose to interface with your database.	5%
Include at least one form of user authentication/authorization within the application.	2%

(35%) Front-End Development	Weight
Use React to create the application's front-end.	10%
Use CSS to style the application.	5%
Create at least four different views or pages for the application.	5%
Create some form of navigation that is included across the application's pages, utilizing React Router for page rendering.	5%
Use React Hooks or Redux for application state management.	5%
Interface directly with the server and API that you created.	5%

The following section is NOT included in the requirements for this project. Completing this section is NOT required. This section will NOT negatively impact your grade if left unfinished.

This section is intended for learners looking to go the extra mile by showcasing additional skills such as project management, and optional technologies like TypeScript.

You must complete ALL other requirements to receive credit for this section; however, this extra credit will not be included if you have already received the maximum 100% grade. The extra credit

can only offset points lost elsewhere.

(5%) Extra Credit	Weight
Adhere to Agile principles and the Scrum framework. Perform stand-up sessions (with an instructor) when possible.	1%
Successfully track your project using a software similar to Jira.	1%
Build your application primarily with TypeScript.	3%