



CP 325.9: Capstone Completion

Version 1.0, 08/14/23

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

Introduction

This document will remind you of the requirements of the course Capstone Project (CP), and give you final tasks to work towards for its completion.

Capstone Objective

- Create a full-stack web application using MongoDB, Express, React, and Node (MERN).

Timeline

As the course progressed, you were given incremental assignments that challenged you to integrate the content that you have learned into your capstone project, pushing it closer to completion over time.

At each stage of your project, you have witnessed your own growth through the application's progress. Hopefully, both you and your application have embraced change, persevered through the struggles, and done some try/catching when there was an exception to be handled.

Now that you have reached the final days of the course, there will be **eight days** of dedicated project time for final tasks. You have made it to the end of the race, now it is time to cross the finish line!

The complete application is due by 11:59pm on submission day, following the class time zone.

Submission

When your project is finally complete, submit the link to your completed assessment using the **Start Assignment** button on the assignment page in Canvas.

Your submission should include:

- A GitHub link to the repository for your completed project.

You will also be given the opportunity to present your project to guests, instructors, and/or a Talent Advocate Manager (not the class) upon completion. Lean into this opportunity to become accustomed with speaking about your work, both from a technical and non-technical perspective.

Your presentation should include:

- A demonstration of the application.
- An overview of the challenges you endured, and how you handled them.
- A short question-and-answer period.

Your instructor will schedule practice presentations throughout the development phase.

Instructions

This final assignment is a reminder to double-check everything you have done so far, and finish any remaining tasks over the next eight days. Be sure to reference the requirements outlined below to ensure your project meets the technical criteria outlined.

Your project vision and its implementation likely changed substantially from the planning phase many weeks ago. Check in again with your instructor to make sure that you are on pace for success by the end of this final project phase.

Finishing Up

Reference your original project plan and any notes you have made for yourself since then. Ensure that you have time to complete all of the tasks you outlined for yourself. Prioritize well, and if you cannot finish everything, decide what is most important.

As you finish building your application, ask yourself the following questions:

- Did you deliver a project that met all of the technical requirements?
- Given what the class has covered, did you build something reasonably complex?
- Did you add a personal touch or a creative element into your project submission?
- Did you deliver something of value to the end-user (not just a login button and an index page)?
- Did you follow the code style guidance and exercise best practices?
- Did you provide an appropriate level of comments?
- Did you try to deploy your application to a public URL as a personal stretch goal?

Lastly, congratulations! You have come incredibly far in a very short period of time. Be proud.

Requirements and Rubrics

These requirements are intended to give you as much creative freedom as possible while still showcasing the skills gained during your technical learning journey.

Requirements

Your project folder should be named as follows, for submission:

- LastName_FirstName_ProjectName_Capstone

| (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% |

| (5%) Presentation | Weight |
|--|---------------|
| Create a short overview of your application. | 1% |
| Highlight the use cases of your application. | 1% |
| Highlight the technical functionality of the application, from a high-level perspective. | 1% |
| Discuss what you have learned through the development of the application. | 1% |
| Discuss additional features that could be added to the application in the future. | 1% |

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% |