TASK

Capstone Project Part II: Develop Full Stack MERN Application

Visit our website

Introduction

WELCOME TO THE CAPSTONE PROJECT PART II!

Congratulations on making it to the second part of your final Capstone project of this Bootcamp! This final Capstone project will require that you completely showcase the skills that you have acquired as a full-stack web developer. You will be required to completely conceive, design and build a full-stack web application using the MERN stack.

PROJECT REQUIREMENTS

You can choose to design any web application that you like as long as it meets the following requirements:

- It is built using Express, React, and MongoDB (the MERN stack).
- It creates, reads, updates, and deletes (CRUD) information from MongoDB.
- It has a custom server built using Express.
- It authenticates users using JWT.
- The front-end is built using React. You can use a React framework (e.g. Create React App or Next.js) of your choice.
- The application allows for normal end-user access and admin access. An administrator should be able to monitor and make changes to users' behaviour.

Need some inspiration? Here are some examples of projects:

- An application that allows doctors to track information about patients and appointments. Here normal end-users may only be able to view appointments for a certain period of time (e.g. for a day or week), whereas an administrator can make, cancel or edit appointments and patient information.
- An application that a store owner can use to keep track of inventory. A normal end-user may be able to see when certain stock is running low and

order new stock. An administrator would be able to add an item to the inventory, set the thresholds for stock (e.g. how much of a certain item to order at a time, when should new stock be ordered etc), decide to no longer stock certain items etc.

- An application that a conference centre could use to advertise upcoming events. Normal end-users might be able to see a list of upcoming events, whereas an administrator might be able to add information about new events, cancel events, edit information about events etc.

- Your own implementation of an existing app, e.g. write your own web version of WhatsApp, Twitter or Instagram.

SUBMISSION CRITERIA

In this part of the Capstone Project, you will be required to code, test, and debug the application you designed in part 1.

You will need to submit:

1. The source code for your project. As with all your other projects, you will need to delete all 'node_modules' directories and submit compressed project folders for the front-end and back-end of your application. Ensure that your code meets the following criteria:
    a. The UI should be attractive, easy to use, and intuitive. Be sure to incorporate any suggestions that you received from your reviewer for part 1 of this Capstone Project regarding UI design.
    b. You should ensure that your application has been appropriately tested. You should include at least one **snapshot test** and appropriate **unit tests** for both the front-end and back-end of the application.
    c. A reviewer should be able to launch your app by typing 'npm start' from the command line interface.
    d. The file structure of the project should be **well organised** and easy to understand and use.
    e. Your code should be **well documented** with appropriate comments.
    f. The code should also be easy to read, adhering to **Google's style guide** about indentation, meaningful variable and component names, etc.
    g. Your code should be **modular** to make testing, debugging, code reuse and maintenance of your app easier.

2. An expanded readme.md file. In part 1 of this Capstone Project, you already created a file called readme.md. Expand this file so that it now also includes the following:
    a. An explanation of how to use the app.
    b. Clear instructions that an end user will be able to follow to install, test, and run your app on their local machine. This should include instructions for modifying any MongoDB URIs or API keys, etc., for your app.
    c. A description of the measures that you have taken to ensure the **security** of this app, including a description of how API keys have been dealt with.
    d. A description of any **third-party APIs** that you have used in your code.
    e. A description of where and how the application has been deployed. How has it been deployed? Back-end and front-end together or separate? Why?
    f. A **link (or links)** to the **deployed app**.

Instructions

- This project involves creating apps that need some modules to run. These modules are located in a folder called 'node_modules'. Please note that this folder typically contains hundreds of files which, if you're working directly from Dropbox, has the potential to slow down Dropbox sync and possibly your computer. As a result, please follow this process when creating/running such apps:
    - Create the app on your local machine (outside of Dropbox).
    - When you're ready to have a reviewer review the app, please delete the node_modules folder.
    - Compress the folder and upload it to Dropbox. Your reviewer will, in turn, decompress the folder, install the necessary modules, and run the app from their local machine.

Compulsory Task 1

Follow these steps:

- Create a full-stack web application that meets ALL the criteria listed previously for this Capstone Project.
- Deploy your app. Add the link to your deployed application to the readme.md file of your project.
- Push all the work that you have generated for this project (including the design documentation that you generated in the first part of the project) to GitHub.
- Add a text file in your DropBox with the GitHub link.

Completed the task(s)?

Ask an expert to review your work!

Review work

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.