TASK

Capstone Project - React App

Visit our website

Introduction

WELCOME TO THE REACT APP CAPSTONE PROJECT

Now that you know how to use React to create an attractive and functional web application, you are going to apply your knowledge in this task to create a web application that you can add to your developer portfolio. You will also create a Git repository to facilitate version control of your project.

Let’s start by talking about version control.

WHY DO YOU NEED A VERSION CONTROL SYSTEM?

Below are some of the benefits of using a Version Control System for your projects:

- **Collaboration:** When working on a large (or even medium-sized) project, more often than not, you will find yourself working as part of a team of developers. Therefore, you will have multiple people who need to work on the same file. Without a Version Control System in place, you will probably have to work together in a shared folder on the same set of files. It is difficult to know when someone is currently working on a file and, sooner or later, someone will overwrite someone else's changes.

  By using a Version Control System, everybody on the team is able to work on any file at any time. The Version Control System then allows you to merge your changes into a common version, so the latest version of the project is stored in a common, central place.

- **Storing versions:** It is especially important to save a version of your project after making any modifications or changes. This can become quite confusing and tedious if you do not have a Version Control System in place. A Version Control System acknowledges that there is only one project being worked on; therefore, there is only one version on the disk you are currently working on. All previous versions are neatly stored inside the Version Control System. When you need to look at a previous version, you can request it at any time.

- **Restoring previous versions:** Being able to restore older versions of a file enables you to easily fix any mistakes you might have made. Should you wish to undo any changes, you can simply restore your project to a previous version.

- **Understanding what happened:** Your Version Control System requires you to provide a short description of the changes you have made every time you decide to save a new version of the project. It also allows you to see exactly what was changed in a file's content. This helps you understand the modifications that were made in each version of the project, even if you weren't the one who made them.

- **Backup:** A Version Control System can also act as a backup. Every member of the team has a complete version of the project on their disk. This includes the project's complete history. If your central server breaks down and your backup drive fails, you can recover your project by simply using a team member's local repositories.

Read the **Git Basics additional reading** in your Dropbox folder before continuing to create your React App for this Capstone Project.

THE TASK AT HAND

For this Capstone Project, you will be tasked to create a Web Store using React. To be able to successfully do this, you will need to consolidate all the concepts you have learnt about React, JSX, and JavaScript.

Create a React app that displays products of a fictional (or real, if you have a client in mind) online store.

The app you create should meet the following criteria:

1. It should be created using Create React App.
2. It should include at least 4 different types of attractively styled components that respond to user interaction. Feel free to use React-Bootstrap or another library and/or your own custom stylesheets.
3. The user should be able to register by entering the following details:
    - First name
    - Surname
    - Username
    - Email address
    - A password

   Make sure that all input is appropriately validated (i.e. ensure the user has entered a valid email address, a strong password and any other validation checks that you think are necessary).

4. The user should be able to log in by entering a username that is stored in a state and displayed in a header component.
5. The header component should include a navigation menu and be displayed at the top of the page for all URLs.
6. The application should contain at least 3 components:
    - a landing page,
    - a store or product page that displays the items to the user,
    - and a cart page.
7. A number of components should be rendered using the `array.Map()` method. Each component rendered in this way should have a key that uniquely identifies it.
8. User interaction should modify the state of some components.
9. The application should make use of Redux to manage the state.
10. The user's purchases should be added to a "cart" state that is used to display the user's selection and the total cost, etc., on the cart page.
11. The user should be able to select a method of shipment.
12. The user should easily be able to request "help", which will inform the user about the details of the shipping options.
13. The UI should be attractive and intuitive.
14. Initialise a Git repository in your project folder to track changes as you go along.
15. Ensure that your project folder includes a file called "README.md" that explains the application and how to use it.
    - In your editor, create a new file and name it README.md. The **.md** extension stands for Markdown, which is a lightweight markup language used for formatting text and creating documentation.
    - You can quickly brush up on Markdown syntax for your **README.md file here**. Keep it simple! Provide clear instructions that an end user will be able to follow to install and run your app on their local machine. You can also read more about the **README GitHub guide here**.
16. The code reviewer should be able to launch your app by typing the following into the command line interface:
    - '`npm install`'
    - '`npm start`'
17. The file structure of the project should be well organised and in line with **these guidelines**. The code should also be easy to read, adhering to **Google's style guide** about indentation, meaningful variables, component names, etc.
18. Your code should be well-documented with appropriate comments.

Capstone Project

Follow these steps:

- Create a Web Store application using the Create React App Starter Kit.
- Ensure that the application adheres to **all** the criteria listed above for this Capstone Project.
- Once you are ready to have your code reviewed, **delete** the **node_modules** folder before submitting your project folder.
    - This folder typically contains hundreds of files that have the potential to **slow down Dropbox sync and possibly your computer** if you're working directly from your Dropbox folder. Therefore, you may prefer to work in a different folder on your local machine and move it to Dropbox when you are finished.
    - When creating a React app the node_modules folder should automatically be added to the **.gitignore** file so that it does not form part of the project's Git repository.
- Please compress (e.g. .zip) your project folder before you submit it via the relevant task folder in Dropbox.

Rate us

Share your thoughts

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

Think 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.

REFERENCE

Meta Open Source. (2023). Learn React - Quick Start. Retrieved 25 August 2023, from **Learn React - Quick Start**.