Peer-graded Assignment: Peer review: Build a calculator app

Deadline May 26, 11:59 PM CEST

It looks like this is your first peer-graded assignment. Learn more

Ready for the assignment?
You will find instructions below to submit.

Instructions My submission Discussions

By working through the lessons in this course, you've learned the necessary skills and knowledge to develop a calculator app using React.

You were provided with code snippets and your task was to use these, plus any of your own code, to complete a calculator app that can perform the four basic mathematical operations: addition, subtraction, multiplication, and division.

You will now take part in a peer review exercise in which you will submit your completed calculator app for two of your peers to review. You will also be required to review two of your peers' calculator apps.

More detailed criteria are covered in the grading criteria overview below.

Grading Criteria Overview

When you submit your assignment, other learners in the course will review and grade your work. They will be looking at the following.

Calculator functionality

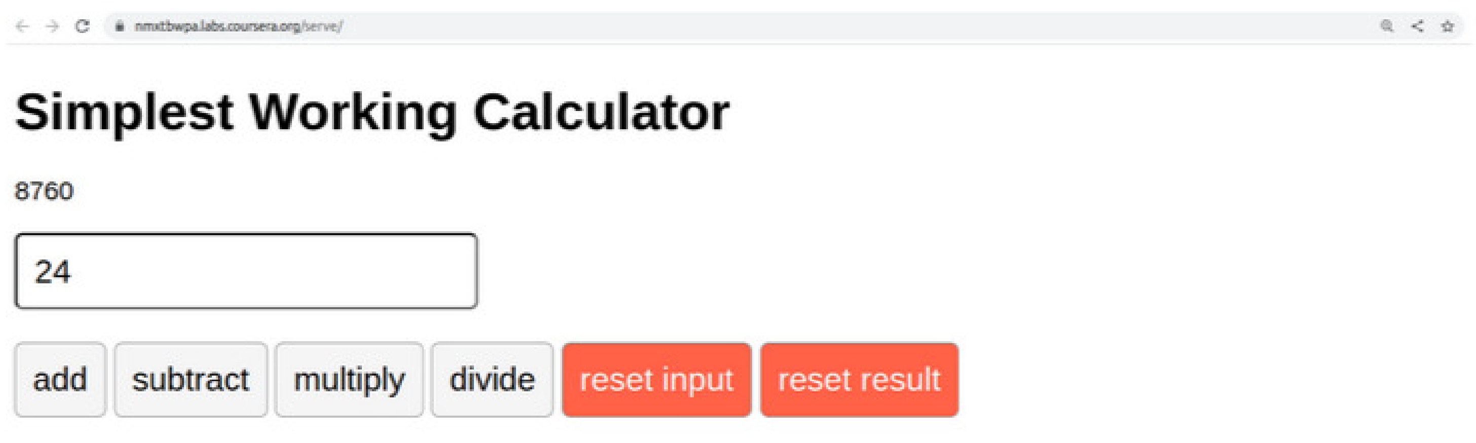
When interacting with the calculator app in the UGL or VS Code:

- Did each button have a function?
- Did each button contain the required mathematical operator?
- Did each calculation provide the correct result as its output?
- Can you suggest any improvements for the calculator app?

You'll also need to give feedback on and grade the assignments of two other learners using the same criteria.

Example Submissions

Example of the calculator app.



nmxtbwpa.labs.coursera.org/serve/

Simplest Working Calculator

8760

24

How to create and submit your assignment

Your calculator has to be a React app. You can develop this app in any lab environment that you have used earlier in this course. Alternatively, you can use VS Code on your local machine. The reading Setting up a React project in VS Code provides the steps on how to set up VS Code on your computer if you choose to do so.

If you plan on using a lab environment that you have used previously, your work will only be available during that session.

Note: Make sure that you download your files before exiting the lab environment. To work on your project again later, you can open the React app on your local machine and copy and paste the code into the template files in the lab again. Remember to download the edited versions again at the end of the session.

To submit your project you need to download your files to your local machine by right-clicking on them in the Explorer panel and selecting "Download".

You will be required to submit your React app by uploading a zipped project folder that contains your app's code. To learn more about how to zip and unzip folders visit the Mac or Windows support page.

Important note: Before uploading your solution, make sure to delete the **node_modules** folder from your project. This will save about 500Mb on the upload which will ensure that you can upload the file.

How to review

Once you have submitted your app, you are required to review two peer submissions. You can view the peers that you need to review in the "Peers to review" section. You need to download their zipped project folders, unzip them and open them in VS Code.

When you open a peer's project on your local machine in VS Code, you need to open the integrated terminal using the *View, Terminal* command from the top-level main menu.

Inside the terminal, you need to open the folder that contains the **`package.json`** file. This is because you need to install the **`node_modules`** folder using the **`npm install`** command, with your terminal pointing to the folder that contains the **`package.json`** file. You should be familiar with these steps because they are the same ones you've had to take while working on the ungraded labs in the course's live code labs.

Once the **`npm install`** command is finished, you can serve the app locally by running the npm start command, with your terminal still pointing at the folder containing your peer's app files.

Alternatively, you can copy and paste your peer's code in the **`App.js`** file in the UGL project sandbox mentioned earlier in this lesson. For this exercise, the **`App.js`** file is the only file that you need to update. This approach is probably easier than the first one for which you had to install node_modules locally.

Examples of Good Feedback

The focus of your feedback should be on the functionality of the calculator app.

Follow the prompts and look for the expected output. If you notice any errors in the functionality of any of the elements of the calculator, you will have the opportunity to provide guidance to your peers on how they might fix the error.

An example of good feedback would be:

"On the whole the calculator app performed as expected; however, there was one button function that did not perform the mathematical operation as expected. The *subtract* button gave the incorrect outcome for the instructions provided. I would suggest reviewing the code that you have written for the function of this button, you could also revisit the lesson *Dynamic events and how to handle them*."

Like Dislike Report an issue