

Assessment Three
Portfolio

Best Programming Practices (Web & Mobile Development)

BCDE211

Semester One, 2022

Due date: Friday, 10 June 2022

Time: 5:00pm

TOTAL MARKS: **50**

Student Name/ID

If a learner needs to apply for an extension, they can do so by completing the extension request form ([app505m-extension-of-time-application.pdf](#)). Extension requests must be submitted to the lecturer prior to the assessment due date.

If an assessment is handed in late without an approved extension, a penalty of 10% per day will apply, up to a maximum of 50%. If an assessment is received more than five days after the due date without an approved extension, it will not be marked. Should a learner wish to appeal any decisions, they may do so in writing to the Head of Department within ten days of receiving the decision.

This assessment is worth 50% of the total marks for this course.
To pass this course, learners must gain an average of at least 50% across all assessments, and gain at least 50% in Assessment 3.
This paper has four (4) pages including the cover sheet.

Learning Outcome Assessed in this assignment.

Learning Outcome Three

Apply knowledge of standards and tools to build complex systems.

OVERVIEW

Your task for this assessment is to create a React-based user interface for the cycling related solution of your Assessment #2 of this course.

More specific tasks are listed below.

1 Implementing a user interface for your solution, i.e., the model, of Assessment #2

This task is to build a working user interface by using React. Please note that the updated requirements stated below may require you to update your solution of Assessment #2.

Features:

- (1) Create a whole that acts as a container and gateway of accessing to parts – display information about the ‘Whole’
- (2) Add a part – allow user input – show the user adding a new part – use appropriate input controls
- (3) Sort parts – at least 2 different sort orders
- (4) Filter parts – filter “on/off” and at least 2 more filter options
- (5) Delete a selected part – show the user deleting a ‘part’
- (6) Save all parts to LocalStorage – demonstrate data being added, and then the browser being closed
- (7) Load all parts from LocalStorage – demonstrate the browser being started and data that was entered previously being displayed
- (8) Update/edit a part – show a ‘part’ having all its fields edited
- (9) Discard /revert edits to a part – show the user rolling the fields of a part back to the values before the last editing
- (10) Validate inputs – only accepts valid input. Complains about bad input or missing input
- (11) A calculation within a part – show the calculation on the screen for each part
- (12) A calculation across many parts – show the result of the calculation across the whole on the screen
- (13) Provide default values – show the default value appearing when the user starts to add a new part
- (14) Find a part given a search criterion – show a successful search – show an unsuccessful search
- (15) Get all parts – show a list / table of all the parts on the screen

Note that:

The submitted PowerPoint slides need to include code arranged by feature and screen recording for each feature demonstrating that the feature is working.

(30 marks)

2 Completing your solution of Assessment #2

If you scored less than half of the total marks of Assessment #2, you could upgrade and resubmit your updated solution of Assessment #2.

You may not score marks for anything that you successfully implemented in Assessment #2, but only for new code.

(25 marks)

3 Rewriting your solution of Assessment #2 in TypeScript if your solution of Assessment #2 was written in JavaScript

Configure TypeScript so that it targets ES2022 and ensure 'use strict' is always emitted.

(15 marks)

4 Implementing key PWA features in your React-based application

5 marks for each working key PWA feature, e.g., service workers and caching. Please refer to <https://web.dev/learn/pwa/caching/> for key PWA features.

5 Testing your JavaScript solution of Assessment #2

All the 15 features are required to be tested by Jest. The HTML coverage report is required to demonstrate the full coverage.

(5 marks)

6 Testing your TypeScript solution of Assessment #2

All the 15 features are required to be tested by Jest. The HTML coverage report is required to demonstrate the full coverage.

(10 marks)

7 Testing React Components with Jest

All the React components related to the 15 features are required to be tested by Jest. The HTML coverage report is required to demonstrate the full coverage.

(10 marks)

8 Creating a React application by using create-react-app in JavaScript/TypeScript

Creating a React application from scratch instead of embedding React into a web application in JavaScript. You can get max 10 marks. If you use TypeScript to complete this task, you can get max 20 marks. Please refer to <https://reactjs.org/docs/create-a-new-react-app.html>

(20 marks)

9 Do something else

Please **DISCUSS WITH TEACHER TO GET APPROVAL**

(10 marks)

MARKING GUIDE

The marks of each task will be awarded based on the correctness of the corresponding solution. The final marks for this assessment will also depend on how well coding style and standards have been applied in the solution of this assessment.

The final marks of this assessment will be the total marks awarded for all tasks plus the marks awarded for “coding style and standards”. Please note that the marks for “coding style and standards” are **negative**.

Marks	-5	-4	-3	-2	-1	0
Coding style and standards	Not attempted	Code hard to follow in one reading, poor formatted and structured	Relatively well-formatted, understandable code and relatively well-organized file structure	Well-formatted, understandable code and well-organized file structure	Relatively well-structured and presented (e.g., modularized, commented)	Non-redundant, Well-structured, properly presented (e.g., modularized, commented)

SUBMISSION

Learners must submit their work as a single .zip file to the drop box on BCDE211 course Moodle site by the deadline indicated. The .zip file should contain the followings.

- A digital copy of all your code for this assessment.
- A 10-minute maximum (optionally narrated) presentation explaining what you have coded for this assessment. Your presentation **MUST** be in PowerPoint format.

In general, your presentation needs to show:

- The expected (i.e., self-estimating) marks for each task you claim marks for
- Which task you are claiming marks for
- The evidence showing the task has been successfully completed
- The source code behind the completed task, i.e., snapshots of code

Learners should show the progress of their assessment work to the course tutor every week and get formative feedback.