College of Engineering, Construction and Living Sciences
Bachelor of Information Technology
ID608001: Intermediate Application Development Concepts
Level 6, Credits 15

Practical

Assessment Overview

In this **individual** assessment, you will provide documentation that addresses several aspects of the design and development process. In addition, you present both applications via a video recording.

Learning Outcomes

At the successful completion of this course, learners will be able to:

1. Apply design patterns and programming principles using software development best practices.
2. Design and implement full-stack applications using industry relevant programming languages.

Assessments

Assessment	Weighting	Due Date	Learning Outcome
Practical	20%	13-11-2024 (Wednesday at 4.59 PM)	1
Project	80%	13-11-2024 (Wednesday at 4.59 PM)	1, 2

Conditions of Assessment

You will complete this assessment during your learner-managed time. However, there will be time during class to discuss the requirements and your progress on this assessment. This assessment will need to be completed by **Wednesday, 13 November 2024** at **4.59 PM**.

Pass Criteria

This assessment is criterion-referenced (CRA) with a cumulative pass mark of **50%** over all assessments in **ID608001: Intermediate Application Development Concepts**.

Authenticity

All parts of your submitted assessment **must** be completely your work. Do your best to complete this assessment without using an **AI generative tool**. You need to demonstrate to the course lecturer that you can meet the learning outcome(s) for this assessment.

However, if you get stuck, you can use an **AI generative tool** to help you get unstuck, permitting you to acknowledge that you have used it. In the assessment's repository **README.md** file, please include what prompt(s) you provided to the **AI generative tool** and how you used the response(s) to help you with your work. It also applies to code snippets retrieved from **StackOverflow** and **GitHub**.

Failure to do this may result in a mark of **zero** for this assessment.

Policy on Submissions, Extensions, Resubmissions and Resits

The school's process concerning submissions, extensions, resubmissions and resits complies with **Otago Polytechnic | Te Pūkenga** policies. Learners can view policies on the **Otago Polytechnic | Te Pūkenga** website located at https://www.op.ac.nz/about-us/governance-and-management/policies.

Submission

You **must** submit all files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – https://classroom.github.com/a/lvW3JyHk. Late submissions will incur a **10% penalty per day**, rolling over at **5:00 PM**. If you do not have not one, create a **.gitignore**. The latest application files in the **main** branch will be used to mark against the **Functionality** criterion. Please test before you submit. Partial marks **will not** be given for incomplete functionality. Late submissions will incur a **10% penalty per day**, rolling over at **5:00 PM**.

Extensions

Familiarise yourself with the assessment due date. Extensions will **only** be granted if you are unable to complete the assessment by the due date because of **unforeseen circumstances outside your control**. The length of the extension granted will depend on the circumstances and **must** be negotiated with the course lecturer before the assessment due date. A medical certificate or support letter may be needed. Extensions will not be granted for poor time management or pressure of other assessments.

Resits

Resits and reassessments **are not** applicable in **ID608001: Intermediate Application Development Concepts**.

Instructions

Documentation - Learning Outcome 1 (50%)

In a **Microsoft Word** document called **documentation**, explain the following:

- Design Patterns
  - Explain the design patterns used in both applications.
  - For each design pattern, provide a code snippet that demonstrates how it is implemented.
  - Explain the advantages and disadvantages of the chosen design patterns.
- Programming Principles
  - Explain the programming principles used in both applications.

  - For each programming principle, provide a code snippet that demonstrates how it is implemented.
  - Explain how the programming principles contribute to code maintainability and readability.

Presentation - Learning Outcome 1 (50%)

- Present both applications via a video recording. In addition, you need to answer the following:
  - Explain what tools and technologies did you utilise to streamline your design and development workflow?
  - Explain what challenges did you encounter during the design and development process and how did you overcome them?
  - Explain what strategies did you employ to maintain code quality and best practices?
  - Explain how did you handle testing and debugging of both applications?
- The presentation must not exceed **30 minutes** in length.
- Upload your presentation to **OneDrive**. Include a link to your presentation in your repository's **README.md** file.