Assessment Brief

Subject Code and Name	DEV1003 - Advanced Applications
Assessment Number	2
Assessment Title	Construct a Back-End web Application
Assessment Type	Group Programming Project
Words, Size or Duration	1 programming project
Subject Learning Outcomes	SLO3, SLO4, SLO5, SLO6
Submission Date / Time	Due by 11:55pm AEST Sunday end of Module 4.
Weighting	40%

Assessment Purpose

A “full-stack” application is actually composed of two applications: a back-end application and a front-end application. To be considered a full-stack developer, you should be able to create both to a professional level. Modern software developers should also be able to work collaboratively to create larger applications, as it’s extremely rare to find workplaces where a developer works completely on their own.

To solidify your knowledge of modern software development concepts and show your ability to work with others effectively, you should be able to create a back-end web application.

Assessment Task / Item

For this assessment, you must submit an application that meets the programming, design & content requirements to showcase your skills as a software developer. While this is a group project, each individual will also be required to submit a peer review, commenting on the contribution of each individual. Responses on peer reviews will contribute to each individual’s result.

Assessment Instructions

Back-End Application

A back-end application is one of two major components to a full-stack web application. While you must keep in mind the overall full-stack application when working on either front- or back-end applications, they must stand on their own as valid, functional applications.

Your back-end application must meet the following requirements:

- Demonstrate usage of a defined style guide
    - If you're using an existing style guide, reference that in your project's readme documentation.
    - If you're inventing your own style guide for this project, explain that in your project's readme documentation.
    - All software code developed must reflect the chosen style guide.
- Demonstrate DRY (Don't Repeat Yourself) principles in all software code developed
- Demonstrate appropriate usage of libraries, packages or other externally-sourced code (eg. NPM or Yarn packages for JavaScript, PIP packages for Python)
- Create secure web server endpoints using web request features like headers, body content, params, and authorization
- Create web server endpoints using appropriate HTTP verbs for CRUD operations
- Create tests covering functionalities of the app (eg. each CRUD route should have an associated test)
- Use appropriate error handling techniques throughout the code

The project must contain a README.md file that includes the following information:

- An explanation of all used technologies within the project, such as:
    - dependent software and packages
    - hardware required to run the application
    - comparisons to alternative technology choices
    - purposes of chosen technologies
    - licensing of chosen technologies
- An identification and/or explanation of the style guide used in the programming throughout the project

Peer Review

In addition to the above items, each team member must complete a peer review using the peer review template. This should consider each individual's efforts during this assessment. The peer review contributes to each team member's individual marks.

Submission

All work must be submitted via Canvas, in the assignments section appropriate to this brief.

The **Back-End Application** submission must be made by one team member on behalf of the team. The **Peer Review** submission must be made by each individual team member.

Please ensure the above mentioned submission date and/or time are adhered to, or penalties may apply.

When submitting your work, please save your files using the naming convention below.

[Student_ID]_[Surname]_[First Name]_[SubjectCode]_[Assessment_#].zip

E.g.: **1234_Singh_Visha_PRG1002_Assessment_01.zip**

For more information on late submission, please see the **Assessment Policy**.

Academic Integrity

The integrity of the assessment process is fundamental for ensuring appropriate evaluation at AIT. All work submitted should be your own, and where additional resources are used, they must be referenced according to the Harvard style. Additionally, TurnItIn is available in the LMS to test plagiarism in your writing.

For more information on academic integrity, please see the **Academic Integrity** and **Academic Integrity Penalties Policies**.

Appeals

Fair application of the assessment rubric, rules and guidelines should be administered for each assessment. If you feel an evaluation requires further consideration, you may be entitled to an appeal.

For more information on your right to an appeal, please see the **Assessment Appeals Procedure and Policy.**

Policies

For access to the policies mentioned above and related to education at AIT, please see the footer of the AIT website, and follow the link named **Education Policies and Procedures.**

Website: https://www.ait.edu.au

Assessment Rubric

Task Descriptor	**(HD)** **High Distinction** *(85-100%)*	**(D)** **Distinction** *(75-84%)*	**(C)** **Credit** *(65-74%)*	**(P)** **Pass** *(50-64%)*	**(F)** **Fail** *(0-49%)*
EXPLAINS the relevance and impact of the utilised hardware and software technologies within a software project. **15%** SLO 3, SLO 4	Appropriate explanation of ALL of the hardware and software technologies used within a software project, including information on ALL of the following: industry-relevance of chosen technologies, comparisons to alternative technology choices, purposes of chosen technologies, and licensing of chosen technologies.	Appropriate explanation of ALL of the hardware and software technologies used within a software project, including information on MOST of the following: industry-relevance of chosen technologies, comparisons to alternative technology choices, purposes of chosen technologies, and licensing of chosen technologies.	Appropriate explanation of ALL of the hardware and software technologies used within a software project, including information on AT LEAST ONE of the following: industry-relevance of chosen technologies, comparisons to alternative technology choices, purposes of chosen technologies, and licensing of chosen technologies.	Appropriate explanation of SOME of the hardware and software technologies used within a software project, including information on AT LEAST ONE of the following: industry-relevance of chosen technologies, comparisons to alternative technology choices, purposes of chosen technologies, and licensing of chosen technologies.	Does not provide an appropriate explanation, or provides an incorrect explanation.
APPLIES established code style and conventions in the specified programming language consistently in all code produced. **5%** SLO 3	Applies code style and convention consistently across ALL CODE produced with NO BREACHES of the used code styles or conventions.	Applies code style and convention consistently across ALL CODE produced with NO MORE THAN TWO BREACHES of the used code styles or conventions.	Applies code style and convention consistently across ALL CODE produced with SOME MINOR BREACHES of the used code styles or conventions.	Applies code style and convention consistently across THE MAJORITY OF CODE produced with SEVERAL BREACHES of the used code styles or conventions.	Fails to apply code style & conventions to the majority or any of the code produced.
APPLIES DRY (Don't Repeat Yourself) coding principles to code. **10%** SLO 3	Applies DRY coding principles to ENTIRE codebase.	Applies DRY coding principles to MOST OF codebase.	Applies DRY coding principles IN SEVERAL PLACES within the codebase.	Applies DRY coding principles IN FEW PLACES within the codebase IN A BASIC WAY.	Does not apply DRY coding principles.
UTILISES an external library with valid import statements and sensible usage of the library. **5%** SLO 3	Correctly imports FOUR OR MORE relevant external libraries into the code, and uses all imported libraries appropriately.	Correctly imports THREE relevant external libraries into the code, and uses all imported libraries appropriately.	Correctly imports TWO relevant external libraries into the code, and uses all imported libraries appropriately.	Correctly imports ONE relevant external library into the code, and uses all imported libraries appropriately.	Does not correctly import any relevant external libraries into the code.

DEVELOPS a web server that uses industry-standard internet communication features **10%** SLO 3, SLO 4	Correctly and sensibly uses ALL of the industry-standard features listed below in code developed for the web server: headers, body content, params, authorization.	Correctly and sensibly uses THREE of the industry-standard features listed below in code developed for the web server: headers, body content, params, authorization.	Correctly and sensibly uses TWO of the industry-standard features listed below in code developed for the web server: headers, body content, params, authorization.	Correctly and sensibly uses ONE of the industry-standard features listed below in code developed for the web server: headers, body content, params, authorization.	Does not correctly or sensibly use ANY of the industry-standard features listed below in code developed for the web server: headers, body content, params, authorization.
DEVELOPS a web server that conforms to industry-standard communication protocols **10%** SLO 4	COMPLETELY APPROPRIATE usage of industry-standard HTTP verbs, with NUMEROUS verbs implemented within the web server code appropriate for CRUD functionality.	COMPLETELY APPROPRIATE usage of industry-standard HTTP verbs, with ONE OR TWO verbs implemented within the web server code appropriate for CRUD functionality.	MOSTLY APPROPRIATE usage of industry-standard HTTP verbs, with NUMEROUS verbs implemented within the web server code appropriate for CRUD functionality.	MOSTLY APPROPRIATE usage of industry-standard HTTP verbs, with ONE OR TWO verbs implemented within the web server code appropriate for CRUD functionality.	Implementation of industry-standard HTTP verbs is either non-functional, not attempted, or otherwise not appropriate for a web server.
CREATES tests appropriate to essential application functionality. **10%** SLO 5	Creates at least FIVE working tests for AT LEAST FOUR different functions required and used by the application.	Creates at least FOUR working tests for THREE different functions required and used by the application.	Creates at least THREE working tests for TWO different functions required and used by the application.	Creates at least TWO working tests for ONE function required and used by the application.	Insufficient working tests provided, or tests developed do not test functions used by the application.
CREATES an application which handles errors. **15%** SLO 5	Application handles ALL CATEGORIES of errors GRACEFULLY.	Application handles MOST CATEGORIES of errors GRACEFULLY.	Application handles FEW CATEGORIES of errors GRACEFULLY.	Application handles ONE CATEGORY of errors GRACEFULLY.	Does not create an application which handles any category of errors.
COLLABORATES professionally and efficiently with a team during a project. **20%** SLO 5, SLO 6	Project work performed EXCEPTIONALLY within a team, contributing to the work to a HIGH degree and communicating with the team to solve problems EXTREMELY EFFICIENTLY.	Project work performed VERY WELL within a team, contributing to the work to a MODERATE degree and communicating with the team to solve problems EFFICIENTLY.	Project work performed WELL within a team, contributing to the work to a MODERATE degree and communicating with the team to solve problems SOMEWHAT EFFICIENTLY.	Project work performed WELL within a team, contributing to the work to a SMALL degree and communicating with the team to solve problems SPORADICALLY.	Contribution to the project is non-existent, struggles to work as a team member, or doesn’t solve any problems within a team.