Team: TEAM 2
Team members: Hainadine Chamane, Sebastien Pires, Adesola Sogunwa
Marker: Dr Cathryn Peoples
Date: August 2023

Criteria	**Level**	**Comments**
Knowledge and understanding of the topic /issues under consideration (30%)	Pass	A solution has been developed for a Learning Management System. Separation of roles to include admin, teacher, and student users. Personal information entry not obfuscated – beware of 'over the shoulder attack'. No restrictions applied to the amount of data which can be uploaded. No evidence that system logging has been enabled, monitoring for potentially vulnerable scenarios such as a potential brute force attack through a number of failed login attempts in rapid succession. There are other opportunities which might have been explored to provide further breadth to the security aspect, such as restricting the number of login attempts from a specific IP address and/or by a single user within a pre-defined period of time.
Application of knowledge & understanding (30%)	Merit	A check is performed to see if a user exists in database before they are added. Password complexity has not been enabled. Password is hashed using bcrypt. Session token stored in auth file. Technology stack includes Python 3.11, CLI, Postgres database, and Flask API. Authorise_admin decorator used. Nice attention given to the use of factories to write tests. Nice attention also given to removing the result once the test has been run. No evidence of use of regex. Sessions set to the default of 30 seconds. Nice attention to the creation of a setup file. Exception handling using try..catch could have been used to greater effect.

		Opportunity to examine code security quality using Bandit.
Structure & Presentation (30%)	Merit	Object-oriented design, with classes including users, assignment, module and grade. Evidence of careful consideration of design approach taken through having a separate grade class, supporting ability to get all grades for a module or all grades for a tutor. The MS Word document provided is described as being a README file, however, a README is generally presented as a .txt file and I note that a separate README has been provided in the zipped folder of code files. This confusion may indicate a miscommunication within the team. Good detail is provided in the README to prepare the system for use, including how to install docker, and how to get Python 3.11 and a Postgres database. Code files might have been presented according to the design pattern being used, for example, with a model, view and controller file. This would perhaps support a more intuitive structure, and therefore improve future maintainability. The testing document gives a lot of attention to demonstrating that the functional requirements of the system operate as planned. However, please remember that this is a module on secure software development, and it is the security features which we are more concerned with. The functional capabilities are there only to provide a surface on which the non-functional security requirement can be applied.
Academic integrity (10%)	Merit	80 unit tests written. pytest used to check code quality. Git and GitHub used for version control. It is not necessary to use bullet points to present a reference list. Present the reference list in alphabetical order. Relatively good use of commenting across the code. More evidence could have been provided in code to justify how the design decisions made align with community approaches e.g., OWASP. Furthermore, it is helpful to

		explain the goals of the script in a paragraph at the beginning.

Overall comments
<u>Positives:</u> - Great use of unit testing to support code quality.
<u>Points for development:</u> - There is an opportunity to make explicit in your code how each security-related feature complies with a recommended secure software good practice, such as the OWASP proactive controls. This would introduce a greater level of academic integrity to your work. - There is an opportunity to apply more exception handling for dealing with errors and erroneous situations. - There is opportunity to demonstrate more creativity in your design, through strategies such as monitoring for data being deleted in high volumes at unusual times.
<u>Overall Grade:</u> Merit