Module:	**SEN381**
Assessment:	**Final Project**
Total:	**50 Marks**

Final Project

Presentation Requirements:

- PowerPoint slides (a sample will be provided).
- Live demo of the working system.
- GitHub repository.

Description **Marks**

You are required to submit a comprehensive and functional CampusLearn application that demonstrates your understanding of full-stack development, system integration, and modern software engineering practices. while also assessing teamwork, collaboration, and presentation skills. Your presentation will be evaluated across the following 10 criteria:

- **Project Overview & Objectives:** Clear introduction of the problem, proposed solution, overview of CampusLearn platform, technologies utilized, and key features of system. **[5]**

- **Functional Requirements:** Clearly demonstrate that all core features of the CampusLearn platform are implemented, including: **[5]**
  - User registration and login
  - Topic creation and assignment
  - Tutor-student interactions
  - Learning material uploads

- **Non-Functional Requirements:** Show evidence of meeting quality standards such as: **[5]**
  - System responsiveness
  - Platform compatibility
  - Security measures
  - Performance optimization

- **Design Pattern(s)/Architecture Usage:** Describe and implement the use of appropriate software design patterns or architectural styles, such as: **[5]**
  - MVC or (MVP or MVVM)
  - Layered architecture
  - Modular component design

- **Graphical User Interface (GUI) Design:** Submit a polished frontend with functional user interface. Include: **[5]**
  - Responsive layouts
  - Accessibility features
  - Intuitive navigation

- **API Integration and Functionality:** Demonstrate proper backend integration using RESTful APIs or GraphQL. Include: **[5]**
  - Secure data exchange
  - Well-structured endpoints
  - Integration of third-party APIs (e.g., Copilot, Twilio, WhatsApp)

- **Threading and Socket Programming:** Implement background processing or real-time communication where applicable, such as: **[5]**
  - Chat between tutor and student (sockets)
  - Content uploading/downloading in the background (threads)

- **Version Control Usage:** Usage of a version control system like Git to track and manage the codebase. Show: **[5]**
    - Commits with meaningful messages
    - Branching strategy
    - Evidence of group collaboration

- **Presentation & Team Collaboration:** Document and demonstrate effective teamwork, including: **[5]**
    - Task delegation
    - Evidence of communication and group decision-making
    - Flow of presentation

- **Lecturer Evaluation:** Based on the lecturer's discretion, this score will consider: **[5]**
    - Problem-solving approach
    - Originality, creativity
    - Completeness

Marking Rubric:

Excellent	5
Good	4
Average	3
Below Average	2
Poor	1
No submission	0

Additional Information

- This is a Group Project
- 5 Students per group
- Belgium Campus consists of software that can **scan for plagiarism** and a student caught doing this will get 0 marks for this assignment.
- Late assignments will not be accepted; missing the deadline is an automatic 0.