Lab Assessment 2 – Practical Assessment

Due date: Refer to Assignment Submission Box on VU Collaborate

Weighting: 15% of total assessment

Scenario

The Discipline of Information Technology in Victoria University (VU) would like to provide students with some study resources, e.g., iPads, Android phones, high performance computers, textbooks and etc, for supporting their studies in IT courses.
Current students from IT courses are able to submit a request to get relevant resource for their study in one specified unit in IT courses.
Some restrictions are listed:

1. There are more than one IT courses in VU now, and each course has multiple units.
2. Different courses may share units.
3. Each unit has some specifications for extra resources requirements.
4. Students' requests should be approved by the unit teacher in prior to getting the resources.
5. All recourses will be allocated in a first-come-first-serve priority
6. For some valuable resources, for example the high-performance computers, each student can only get one at the same time.

You are appointed as an analyst programmer to develop a web application – Study Resources Request and Allocation for IT discipline of VU.

Your task is to investigate and identify potential resources through the introduction aforementioned or any other resources available online. You will design and develop a web application for the IT discipline of VU that allows teachers to manage the recourses and the students to request resources. The system should render course and unit details, teachers and students' details, resources details as well as the request, allocation details and allow perform transaction.

Model Design Requirements

You are going to design a model to store data for your application with the following specification:

- You have freedom to design your own data architecture and properties for the classes, however it should reflect your understanding about complex entity relationships, including one-to-one, one-to-many, many-to-many, and inheritance.
- The application should store information for every class and relationship. Inheritance should be used to model classes with similar properties. For example, human being class may have common properties such as ID, Name, Contact details (email, phone, address, etc...). Besides, each unique class should also have its own properties. You should provide **justification** in your report, on the proposed classes, properties and their relationships

stating why they are necessary.

- Appropriate **data annotations** and **input validations** should be provided in all model classes. You should customize your own error messages and NOT use the default error messages provided by the template.

An example model is provided in "Sample Report for Lab Assessment 2.pdf" file (VU collaborate), to help you better understand how the application should be implemented. However, you must NOT use the provided model, as it is not suitable specifically for the case in this assignment.

In this Assessment, you are NOT required to complete the entire project.

To fulfill the requirements for assessment 2, you need to submit a report including the following contents:

1. E-R diagram for all possible classes you will use in your project.
2. Detailed description of the classes
3. List all possible views, controllers and actions, with detailed description
4. Screenshots for all views and pages you have designed for this project.

And you are also required to submit a zipped file with all source code you have done. You need to complete the following points at this moment:

1. Initialize ASP.NET Core MVC solution for this project
2. Create corresponding views, controllers and actions according to your report, but no implementation required.

Rubrics:

Marking Criteria	Weight	Marks	Comments
Report (65%)			
E-R diagram shows classes designed for the project	15		
Detailed description for all classes	10		
Views (List, design and description)	15		
Controllers (List and description)	15		
Actions (List and description)	10		
Source Code (35%)			
Initialisation of all Views	10		
Sufficient comments added	5		
All controllers and actions initialized	10		
Project can be run without errors	5		
Coding style	5		