ICT40102 (Programming Stream)

Shared Practical Assessment – Structured Activities

# UNITS

This practical activity assesses the following units:

ICTWEB430 Produce server-side script for dynamic web pages

ICTWEB451 Apply structured query language in relational databases

ICTPRG430 Apply introductory object-oriented language skills

ICTPRG433 Test software developments

# SCENARIO

Refer to **Scenario – Project Brief.pdf** provided on Blackboard – hereafter referred to as *Scenario*.

# DELIVERABLES

You must complete each of the following deliverables by the deadlines specified on Blackboard. These deliverables may be submitted as a single solution file (e.g. Visual Studio Solution) or as separate projects.

## Client Meeting Documentation (Applies to All Units)

To fully understand the requirements of the site, meet with your client (lecturer).

You must clearly articulate the requirements gathered during the interview using appropriate language to clearly discuss ideas, requirements, and solutions.

Conduct an interview to determine and *document* each of the following:

1. Confirm and clarify with your client, your understanding of the project requirements as outlined in the *Scenario*.
2. The dynamic functionality of the site.
3. Discuss and document at least 3 *language* options that could be used to implement this functionality. Discuss and document the advantages and disadvantages of each and give your recommendations for this *Scenario*.
4. Any other requirements of the site.
5. Finalise and discuss your documented requirements with your client.
6. Review and adjust to the documented requirements as suggested by your client.

## Server-side Web Application (ICTWEB430)

* 1. Create a server-side web application in line with the *Scenario*.
  2. Your application must gracefully handle and recover from error conditions.
  3. Markup developed must be HTML 5.0 compliant.
  4. Configure your webserver to deliver over HTTPS.
  5. Configure your app to minimise potential database attacks. *List and describe 2 ways in which you have ensured this.*

## Database Scripts (ICTWEB451)

* 1. E-R diagram for a database to support the requirements of the *Scenario*.
  2. DDL script to create a database and structures necessary to efficiently support the requirements of the *Scenario*.
  3. DML script to populate any sample entries to ensure the web application is functional.
  4. DQL script for testing queries to support the reports specified in the *Scenario [Note: If a CRUDS OOP library is developed (see 4 below) then in-code DQL statements may be embedded in CRUDS methods instead of/or in addition to a DQL script]*.

## Object-oriented Class Library Project (ICTPRG430)

1. Develop an OOP class library layer for this *Scenario* containing public classes to represent each table implemented in section 3 above. Classes to provide efficient **CRUDS** functionality:

**C**reate

**R**ead

**U**pdate

**D**elete

**S**earch (return value type must be a subclassed List)

1. **CRUDS** classes to contain at least two constructors.
2. Additional functionality as required for each class.
3. **CRUDS** classes may optionally derive from abstract base class *CEntity* in the *LibEntity* class library (to be developed with your lecturer in class).

## Test Project and Documentation (ICTPRG433 & ICTPRG430)

1. Develop a unit testing project/test harness that tests CRUDS operations in isolation by referencing the OOP class library project (detailed in section 4 above).
2. Automate testing runs to generate test result files containing the data listed below, as comma-separated values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Name** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |

1. Seek and obtain client feedback on your proposed test plan via email, prior to development.
2. Complete as many test plan iterations as needed until all tests successfully pass.