Contents

[ICTPRG549 - Apply intermediate object-oriented language skills 1](#_Toc129071843)

[Part 2 – design 2](#_Toc129071844)

[Build 2](#_Toc129071845)

[VCS GitHub 4](#_Toc129071846)

[PART 1 Research Report 4](#_Toc129071847)

[PART 2 Test Content Version System 4](#_Toc129071848)

[Part 3 Respond to feedback 5](#_Toc129071849)

[ICTPRG438 - Configure and maintain databases 6](#_Toc129071850)

[Notes: 6](#_Toc129071851)

[REQUIRED EQUIPMENT AND RESOURCES 6](#_Toc129071852)

[Use MySQL Workbench to complete the following tasks: 7](#_Toc129071853)

[SUBMISSION REQUIREMENTS 7](#_Toc129071854)

[Cloud/ AWS 8](#_Toc129071855)

[ICTCLD401 - Configure 8](#_Toc129071856)

[ICTCLD301 - Evaluate 8](#_Toc129071857)

[***WEB DESIGN WORK*** 9](#_Toc129071858)

[ICTWEB517 - Create web-based programs – This can include Clientside script/ Serverside script/ database logically/ potentially Obj orientated language 9](#_Toc129071859)

[PART 1 Note: 9](#_Toc129071860)

[PART 2 Note: 9](#_Toc129071861)

[PART 3 Note: 9](#_Toc129071862)

[PART 4 10](#_Toc129071863)

## ICTPRG549 - Apply intermediate object-oriented language skills

#### Notes:

* In theory Java script can handle all of this
* Git hub is an easy addition.
* It says online Mysql required, but any database using SQL should be acceptable
* Username and password are required, but can be linked on a SQL table as no other conditions are listed.

### Part 1 – Roleplay

### Part 2 – design

Design Design your application using a User Interface (UI) event flow diagram, UI case diagram and UI mockup diagram.

Your design will show:

• GUI requirements and expectations

• Application requirements

o Connection to an online MySQL database

o Transactional integrity

o User name and password

o Display score o Display high score

o Leader ladder of the top 10 players

• Animation requirements

• UI event flow diagram

• UI case diagram

• UI mockup diagram and customisable interface eg: colour, font, etc… Day mode/Night mode

• UI actions

• UI events

• Risk assessment

• Organisational standards and guidelines

• GitHub automation tools

• Unit test

• Sign-off

### Build

Build Two-up Game Now that you have developed a design, you will now develop the application. There are some requirements you must meet to ensure the game meets requirements. The application must:

• Build GUI with required functionality according to task and organisational requirements

• Deliver a Two-up game

• Application requirements

o Connection to an online MySQL database

o Transactional integrity

o User name and password

o Display score o Display high score

o Leader ladder of the top 10 players

• Animation requirements

• GitHub automation tools

• User name and password

• Record each score

• Keep score for each session

• Be customisable in look and feel

• Follow AIT-CS Coding Standards and Best Practices

• Debug code

• Unit test Debug your code by uploading it to an open source online debugger eg:

• W3 schools <https://www.w3schools.com/java/java_compiler.asp>

• OnlineGDB <https://www.onlinegdb.com>

• Screenshots of output results of a successful run. Also include screenshots of previous run errors in results document. Create a unit test using the preferred IDE and save the results.

• A screen recording of running test and test documentation.

• Include o Examining variables

o Trace running code

o Test produced code and is compliant with program specifications

## VCS GitHub

PART 1 Research Report While the management team has identified the requirement for a version control system (VCS) to be used by the development teams at AIT\_CS. You have been tasked to research and report on VCS that is suitable for AIT-CS organisational requirements. GIT VCS has been mentioned during informal discussions. Outcomes of Research

• Research at least three (3) VCS

• Select a preferred VCS

• Identify pre installation requirements for the selected VCS

• Identify installation procedures for VCS

• Identify any installation disruptions to normal daily operations and communicate disruption via email to the appropriate person

PART 2 Test Content Version System This assessment is a continuation of the previous assessment. You are required to install and test the content version system. Requirements for the content version system are as follows:

• Pre-Installation requirements

• Setup/install VCS

• Create a repository

• Create a work flow

• Upload test code files

• Pull files and modify

• Sync files back

• Review changes

• Revert file back to previous version

• Document test outcomes

Assessment Upload Files to be submitted for marking:

• Assessment document (docx, PDF format) Create a GitHub account or use current GitHub account If you already have a GitHub login please go to “Create a repository and work flow” If you do not have a GitHub login the following steps will help you create a GitHub login. Go to GitHub join web page:

• Create your account

• Username – please use a conventional style naming convention

• Email address

• Create Password (at least 15 characters OR at least 8 characters including a number and a lowercase letter)

• Click Verify your account (security feature)

• Click Create account Page 5 of 8 Version: 1.0 Create a repository and work flow Log in to GitHub and create a repository and work flow. Task Create your own repository Repository name: eg: 42project Make public Create a work flow Create a directory (folder) Structure Create/upload three (3) code files Upload 3 working code documents. Task Upload an existing code project file:

• Upload code 1

• Upload code 2

• Upload code 3 Commit changes Page 6 of 8 Version: 1.0 Share code files Share code documents with project team, (two students). Task Share code documents with project team Modify code files Code documents to be modified by team member’s, (two students). Task Code 1:

• Team member to pull document and modify code document then sync (push) back to work flow

• Review changes Code 2:

• Team member to pull document and modify code document then sync (push) back to work flow

• Review changes Code 3:

• Team member to pull document and modify code document then sync (push) back to work flow

• Review changes Page 7 of 8 Version: 1.0 Check version control After code documents have been sync’d back to work flow version control check Task Check/review version control:

• Code document 1

• Code document 2

• Code document 3 Restore previous version Restore previous version for code documents Task Restore previous version:

• Code document 1

• Code document 2

• Code document 3 Complete documentation with outcome of test results and submit to required person

### Part 3 Respond to feedback

## ICTPRG438 - Configure and maintain databases

### Notes:

REQUIRED EQUIPMENT AND RESOURCES You will need the following resources, tools and equipment to complete this assessment task:

• Access to the internet

• MySQL - <https://dev.mysql.com/downloads/mysql/>

• MySQL Workbench - <https://dev.mysql.com/downloads/workbench/>

• Access to a word processing software

• Access to a spreadsheet software

• AIT-CS 2-DB\_Design\_Template.doc

• List of Dogos.xlsx spreadsheet ASSESSMENT INSTRUCTIONS You are required to install the following software to complete the assessment:

• MySQL - <https://dev.mysql.com/downloads/mysql/>

• MySQL Workbench - https://dev.mysql.com/downloads/workbench/ SUBMISSION REQUIREMENTS To complete this task you are required to upload the Part 1 completed document design:

• Complete the assigned tasks

• It is your responsibility to check your submission for spelling, grammar, readability, etc.

• Ensure your submission addresses the assessment instructions and any other requirements.

• Identify code standards and guidelines to be used in the project

• Each student is required to submit their own work.

• Ensure you have provided references for material that has been sourced from elsewhere. SUBMISSION INSTRUCTIONS To complete this task you are required to upload the task document here and follow these guidelines:

• Completed database design documentation.

• Your file MUST be in either a DOC, DOCX or PDF format.

• Check your submission addresses the assessment instructions and any other requirements

Use MySQL Workbench to complete the following tasks: • Develop and build the database • Administer o Create user accounts o Administer user access o Confirm required user access o Perform backup o Perform restore o Inspect and audit data and database integrity o View database health • Visual Performance Dashboard o Improve database performance and tuning • Database Migration o List of Dogos.xlsx spreadsheet You will be required to create a build document with screenshots and explanation of each of the above tasks in reasonable detail.

SUBMISSION REQUIREMENTS To complete this task you are required to upload the Part 2 completed database and build document:

• Complete the assigned tasks

• It is your responsibility to check your submission for spelling, grammar, readability, etc.

• Ensure your submission addresses the assessment instructions and any other requirements.

• Each student is required to submit their own work.

• Ensure you have provided references for material that has been sourced from elsewhere. SUBMISSION INSTRUCTIONS Page 6 of 6 Version: 1.0 Assessment Information for the Candidate: ICTPRG438 - Configure and maintain databases To complete this task you are required to upload the task database and build document here and follow these guidelines:

• Completed database.

• Build document.

• Your file MUST be in either a DOC, DOCX or PDF format.

• Check your submission addresses the assessment instructions and any other requirements

## Cloud/ AWS

### ICTCLD401 - Configure

### ICTCLD301 - Evaluate

## ***WEB DESIGN WORK***

## ICTWEB517 - Create web-based programs – This can include Clientside script/ Serverside script/ database logically/ potentially Obj orientated language

Note While this assessment is set up to be a stand-alone assessment it can also be part of a larger project. If your course has the following two units, then you should review the assessment instructions for those two units before you commence this assessment. 1. ICTWEB430 2. ICTWEB441

PART 1 Note: Only do this part if you have not completed it in a previous assessment.

Use the information above and design the dynamic web pages. You must include a database in your design for the website. Meet with the client, who will be played in this role play by your assessor or other person as directed by your assessor.

PART 2 Note: Only do this part if you have not completed it in a previous assessment

Write the Web pages Now that you have a draft design you will write the pages. Create the dynamic pages using server-side scripting methods. Considerations: Accessibility How embedded scripts will run Testing and debugging the web page and document functionality Security of the site and pages and data Webserver and database Test the website and reiterate as required. Page 5 of 6 Version: 1.0 Assessment Information for the Candidate: ICTWEB517 - Create web-based programs Use debugging from IDE or other sources. Test page security

PART 3 Note: Only do this part if you have not completed it in a previous assessment. User Accounts for dog owners User accounts need to be kept in the database too, with the requirements as above. Member fields:

• First name

• Surname

• contact phone number

• email address

• suburb

• postcode

• pet name

• pet breed

• pet age

• pet gender.

• Pet photo (image) Note: Postcode and suburb must be validated against an external data source. Use and API link to the source, as described here. Create the documents to register as a member of the Vet Dog community. Add functionality to Join/Login to the site. Confirm the website meets the initial specification and requirement provided by your client by running the website up and conducting user level testing. These tests are to be documented with success and failure results recorded.

### PART 4

Create a web application so members can arrange puppy play dates. Build the app and integrate it with the website. It should be called from the webpage for members only to create play date requests. Referring documentation for relevant business standards and procedures, design functionality as outlined below incorporating the required security, privacy and WCAG requirements. The webapp will be using the website membership database for members to send a Puppy Play Date request to other members. Page 6 of 6 Version: 1.0 Assessment Information for the Candidate: ICTWEB517 - Create web-based programs The app will allow the members to search for other Dogs of the same breed or age or gender and send a request with a suggested day and time of the play date. The members should not be able to see any personal data of the dog owners. The receiving members should be notified that their name and phone number will be given to the requested member if they accept the play date. The receiving member should receive a notification when they log in if an invitation has been accepted and the name and phone number of the members provided. The app needs to keep track of requests and acceptances and show the name and number of each party if the date is accepted. The members should be able to remove old requests and the system should remove declined requests after 7 days. A key requirement is for the application to be able to continue to work if it loses contact with the server. This will require client-side coding and use of local state management. Activity undertaken by the user is to be collected and held in state management at least until the connection with the server is restored. The design must outline which state management process will be utilised and why. To save traffic resources, you are to also incorporate server-side application state management to hold required information about users so there is not a need to access the database every time a request is made for user information. Your submission document must include information about your use of client and server-side state management, why it is important to the application, and how you have implemented it. Incorporate an admin process via a link at the bottom of a page that collects and displays the information being held in both the client and server-side states. Your application outline document to include with the submission must address the following points: • Your choice of client-side state management and why (outlining limitations and benefits) • Your choice of server-side state management and why (outlining limitations and benefits) • An outline of why state management is required when developing a web application (outlining limitations and benefits of HTTP/HTTPS) • Testing processes and any issues arising when checking application suitability in two different browsers and rendered in at least two different screen resolutions. Submit your webapp and website and relevant project documentation to get feedback on it. This will be done by submitting your app, site and database to your assessor for grading. You assessor will provide feedback on your work and may request changes for resubmission.