Develop a Database

**Internal Assessment Resource**

**Digital Technologies: Level 1**

This resource supports assessment against Achievement Standard 92005 and US18740

**Standard title:** Develop a digital technologies outcome

**Credits:** 5 Credits

+

**Standard Title:** Create a simple computer program to meet a set brief

**Credits:** 3 Credits (Level 2)

**Resource title: My Database Application**

| Authenticity of evidence | Assessor involvement during the assessment event is limited to providing general feedback which suggests sections of student work that would benefit from further development or skills a student may need to revisit across the work. Student work which has received sustained or detailed feedback is not suitable for submission towards this Standard. |
| --- | --- |

**Student/Ākonga instructions**

**Introduction**

You are going to make a simple database application using Python and SQL in Visual Studio Code over four weeks.

This application can be your own design or a design can be given to you by your teacher. Please note that **databases should NOT be more than 3 tables.** A single table is acceptable as long as the design is correct and you may have to compromise to ensure the database does not become too complex.

Appendix 1 contains a few examples.

Please ensure you have a sound grasp of SQL and Python before attempting this assessment. You must have completed all the relevant coursework before beginning this assessment as your teacher is NOT allowed to give you significant help beyond basic feedback during the assessment period.

You will be expected to be able to follow basic software engineering and programming conventions during this project.

Conventions include but are not limited to:

* Using an organised file structure for your project files
* Using well named files and folders in your project
* Has appropriate variable and function names that describe their purpose
* Using code comments where appropriate
* Following common language conventions for Python and SQL
* Following Database Design standards (eg normalized data to 3NF- well structured database)
* Using version control including descriptive commit messages

You are encouraged to incrementally develop your application through weekly iterations of “plan-develop-test”, keeping a record of your progress as you go inside of this document. Git Version control should also be used as evidence of this.

If you are unable to use git, multiple files in your source code folder must be made with version number included. Failure to do so will limit your ability to gain a mark better than achieved.

You will submit the working application (stored on github or a zip of your entire source code folder) as well as this document for assessment against the criteria listed above.

Now complete the following questions.

# Before Development

### What is the purpose of this Application

|  |
| --- |

### Who might use this Application?

|  |
| --- |

### What are the requirements and specifications for this Application?

| **Requirements** | **Specifications** |
| --- | --- |
|  |  |

### Database Design- Your Entity Relationship Diagram.

|  |
| --- |

### Database Testing Table: SQL Statements

Write down all the “purposes” that you think of for your database and then work out the SQL query that you might use. Test it and make sure it works here before moving on.

| **Purpose** | **SQL Statement** | **Result Success?** |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Program Planning

| **What does your program need to do?** |
| --- |
|  |

| **What are the steps you’ll need to take to get there? (Functionality Milestones)** |
| --- |
| Sprint#1 Goal: |
| Sprint#2 Goal: |
| Sprint#3 Goal: |

| **Initial Plan: Place your flowchart or PseudoCode in the box below** |
| --- |
|  |

During Development

Now you are ready to start an iterative process of development. You have one week of development per “Sprint”.

At the end of each week/Sprint, you will test the application with other people in your class, your teacher or stakeholders and summarise the feedback that they give you to help you to refine the application further.

Note: **Testing** is ongoing throughout development. Every time you write code and run it to see that it works you are testing. **Trialling** is deliberate, organised testing WITH SEVERAL OTHER PEOPLE. Watch them use your applications without prompting and see what they have difficulty with, what they do and what they break.

| **Sprint#1 Trialling** | | **Date:** |
| --- | --- | --- |
| **Who** | **What did they say?** | **How does this help?** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Summary of Testing and Feedback and next steps** | | |
|  | | |

| **Sprint#2 Trialling** | | **Date:** |
| --- | --- | --- |
| **Who** | **What did they say?** | **How does this help?** |
|  |  |  |
|  |  |  |
|  |  |  |
| **Summary of Testing and Feedback and next steps** | | |
|  | | |

| **Final Trialling** | | **Date:** |
| --- | --- | --- |
| **Review: Evaluate how your program meets the original goals and specifications:** | | |
|  | | |
| **Give three examples of how and why you improved your outcome** | | |
| Example 1. | | |
| Example 2. | | |
| Example 3. | | |

### Git Log

| **Link to Github Repo:** |
| --- |
|  |
| **Screenshot of your commit history.** |
|  |

### Final Product

### Github repo link or zip file link (through Google Drive)

|  |
| --- |

# Appendix 1

| **GPU Database**  A database of modern graphics cards, their speeds and their prices for PC gamers who want to be able to compare cards by manufacturer, price and speed. |
| --- |

| **Sports Team Tracker**  A sports team database for a coach to keep track of players attendance, games and points scored over the season. |
| --- |

| **Library Catalog Database**  A comprehensive catalog system allowing librarians to manage book details, borrower information, and lending status. |
| --- |

| **Student Grade Tracker**  An efficient tool enabling teachers to record and analyze student grades for assignments, quizzes, and tests. |
| --- |

| **Inventory Management System**  A streamlined solution for businesses to monitor product inventory, suppliers, and reordering needs**.** |
| --- |

| **Study Guide**  A program to help students study by asking quick quiz questions and checking their answers. They can customise it by being able to add or delete questions from the database. |
| --- |

| **Recipe Organizer**  A user-friendly platform for culinary enthusiasts to collect and categorize recipes, along with ingredients and cooking steps. |
| --- |

| **Fitness Progress Database**  A fitness tracking tool allowing individuals to log workouts, monitor progress, and set fitness goals. |
| --- |

| **Task Manager Database**  An intuitive task management system for organizing tasks, setting priorities, and monitoring progress. |
| --- |

**For Teacher Use**

## Develop a digital technologies outcome

**Domain:** Digital Technology 1.2

Standard: AS92005 v1

**Credits:** 5 (Internal)

Link: <https://ncea.education.govt.nz/technology/digital-technologies/1/2?view=standard>

| **Achieved**  **Develop a digital technologies outcome** | **Comments** |  |
| --- | --- | --- |
| Identifying the purpose, potential users, requirements, and specifications of the outcome | Complete section 1 correctly |  |
| Using appropriate tools and techniques of a digital technologies domain to produce an outcome that addresses the requirements and specifications | Use Python and SQLite to create a program to interface with a database that functions as expected and hands it in on time.  Application has at least read functionality and can output data from the database from users input.  Handed in all project code as a github link or as zip file.  May not have used github  Final project may not look exactly like design and might not have all the intended functionality. |  |
| Testing the outcome to ensure basic functionality. | Teacher observation of testing and the application looks and functions mostly as expected.  May not have done sql query table. |  |
| **Merit**  **Refine a digital technologies outcome** |  |  |
| Following relevant conventions of a digital technologies domain | Meet at least 4 of the following (teacher discretion)  Conventions include but are not limited to:   * Using an organised file structure for your project files * Using well named files and folders in your project * Has appropriate function names that describe the purpose * Using code comments where appropriate * Following common language conventions for Python and SQL * Following Database Design standards (eg normalized data to 3NF- well structured database) * Using version control including descriptive commit messages |  |
| Using information from testing to make improvements to the outcome's fitness for purpose. | Did testing during development. This can be observed or exemplified through a git commit log or multiple versions of code with incremental improvements. |  |
| **Excellence**  **Enhance a digital technologies outcome** |  |  |
| using information from trialling the outcome with others to improve its fitness for purpose | Lots of improvement through at least three versions. Including relevant feedback from others and implementing suggested changes where appropriate.  Teachers should set “Checkpoints” with the whole class to help facilitate this. |  |
| applying tools and techniques effectively in the production of a fit-for-purpose outcome. | The final outcome meets the design and specifications and shows an excellent grasp of Python and SQL to create an above average program that follows all conventions.  Adds features not included in the tutorial like Login/admin/passwords etc. |  |

Teacher Comments:

Standard: US18740v7

Title: Create a simple computer program to meet a set brief

Level: Level 2

Credits: 3

Link: <https://www.nzqa.govt.nz/nqfdocs/units/pdf/18740.pdf>

| **US18740: Create a simple computer program to meet a set brief** | **Comments** |  |
| --- | --- | --- |
| 1.1: A plan is developed to meet the specifications of the given brief and outline a logical design or model to be used as a basis to code the program. | plan includes but is not limited to – milestones, resources; purpose, structure and specifications of the program; computer language to be used; input and output requirements; outline of the proposed testing procedures. logical design may include – flow charts, pseudocode, natural language |  |
| 2.1: The code is created in accordance with the plan and follows the conventions of the chosen language. | Follows PEP-8 conventions for Python  Has a logical structure as defined in the plan. |  |
| 2.2: Following the conventions of the chosen language, meaningful names are applied to all user-defined elements of the program code to facilitate understanding of the program. | Code has appropriate variable names and function names. |  |
| 2.3: The code is internally documented according to the requirements of the brief. | each section of code must contain sufficient comments to explain its purpose |  |
| 3.1: The program is tested for compliance with the brief, to find and eliminate any errors, and code is modified as required. | The program works as intended, including valid, invalid, and boundary cases. |  |
| 3.2: The operation of the program is evaluated and confirms that the specifications set out in the brief are realised. | The program has been evaluated against the specifications outlined at the start |  |

Teacher Comments: