Web Application Development Log

## Design Section

| **Project link on Github:** |
| --- |
|  |

| **Project name:** |
| --- |
|  |

#### 

| **Project Summary: Include purpose and target audience** |
| --- |
|  |

#### 

| **Initial Database Design ERD:** |
| --- |
|  |

#### 

| **WebPage Design- Wireframe and/or Palette** |
| --- |
|  |

#### 

| **Relevant Implications (explain at least 3)- Before Development.** |
| --- |
|  |

## 

## Sprint #1

| Sprint #1 Goals **(before starting sprint 1):** |
| --- |
|  |

#### Sprint #1 Testing

Include tests for everything in your application including all links, input forms, correctness of data and correctness of queries. Add more columns as required.

| **Testing Table** |  |  |  |
| --- | --- | --- | --- |
| **What are you testing** | **How are you testing it** | **Expected Result** | **Pass/Fail** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### Sprint #1 Review

| **Progress Review (100+ words)** |
| --- |
|  |

## Sprint #2

| Sprint #2 Goals **(before starting sprint):** |
| --- |
|  |

#### Database improvements (if any)

| **Entity Relationship Diagram** |
| --- |

#### WebPage Design Improvements (if any)- Wireframe/Sketches/Pallette/Font etc

|  |
| --- |

#### Sprint #2 Testing

Include tests for everything in your application including all links, input forms, correctness of data and correctness of queries. Add more columns as required.

| **Testing Table** |  |  |  |
| --- | --- | --- | --- |
| **What are you testing** | **How are you testing it** | **Expected Result** | **Pass/Fail** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### Sprint #2 Review

| **Progress Review (100+ words)** |
| --- |
|  |

## Sprint #3

| Sprint #3 Goals **(before starting sprint):** |
| --- |
|  |

#### Database improvements (if any)

| **Entity Relationship Diagram** |
| --- |

#### WebPage Design Improvements (if any)- Wireframe/Sketches/Pallette/Font etc

|  |
| --- |

#### Sprint #3 Testing

Include tests for everything in your application including all links, input forms, correctness of data and correctness of queries. Add more columns as required.

| **Testing Table** |  |  |  |
| --- | --- | --- | --- |
| **What are you testing** | **How are you testing it** | **Expected Result** | **Pass/Fail** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

### Final Review

What were the results of your testing? Did you get through all that you expected to? What is causing problems? What went well? What was a challenge? What would you do differently if you had a chance?

| Discuss how planning, testing and trialling led to the development of a high-quality digital technologies outcome. |
| --- |
|  |

| How I addressed the Relevant Implications As discussed before the project, what relevant implications had you identified at the beginning and how did you manage to address them as you went. Were there any others that you came across while making this game? How did you manage to address them? 100+ Words |
| --- |
|  |

# 

# Marking Schedules: For Teacher use only

## AS91902- Use complex techniques to develop a database

**Credits:** 4 (internal)

**NZQA:**  <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91902.pdf>

| **Achieved- develop a database** | **Evidence** | ✔ |
| --- | --- | --- |
| designing the structure of the data | Made an ERD with at least ONE MANY-TO-MANY relationship before the end of the project. |  |
| using appropriate tools and advanced techniques to organise, query and present data for a purpose and end users | Competently normalized data structure  Multiple queries across linked tables  Basic Python CRUD functionality with text input/output |  |
| applying appropriate data integrity and testing procedures | Testing Logs. Do the queries work and return what is expected? |  |
| addressing relevant implications. | the application meets key relevant implications like copyright, IP (correctly sourced and attributed imagery), Privacy (eg passwords hashed) etc. May have highlighted this in final Relevant Implications Task |  |
| **Merit- Develop and informed database** |  |  |
| using information from testing procedures to improve the quality of the outcome | Suggested improvements implemented- improve database design, add or refine entities or tables and relationships  Made improved ERD(s) |  |
| structuring, organising and querying the data logically | No problems with structure and datatypes in the table. No wildcard queries, JOIN’s used efficiently.  2NF at least. |  |
| **Excellence- develop a refined database** |  |  |
| iterative improvement throughout the design, development and testing process | Improved functionality of database and refinements to structure of the data. |  |
| using efficient tools and techniques in the outcome’s production | Effectively used Version Control, extensions or tools (like sqlitestudio) and VSCode. |  |
| presenting the data effectively for the purpose and end users. | Very refined outcome providing all needed functionality |  |

## 

## AS91903 - Use complex techniques to develop a digital media outcome

**Credits:** 4 (Internal)

**NZQA:**  <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91903.pdf>

| **Achieved** | **Evidence** |  |
| --- | --- | --- |
| applying appropriate tools and techniques to meet the purpose and end users requirements | Coded a website in HTML and CSS. Looks and functions mostly like intended in the design. |  |
| applying appropriate data integrity and testing procedures | It functions as intended and the right pages show with the correct data when the links are clicked |  |
| Applying user experience principles relevant to the purpose of the outcome | Site has good usability (HCI) |  |
| addressing relevant implications. | the application meets key relevant implications like copyright, IP (correctly sourced and attributed imagery), Privacy (eg passwords hashed) etc. May have highlighted this in final Relevant Implications Task |  |
| **Merit** |  |  |
| using information from testing procedures to improve the quality of the outcome | Testing and sprint reviews show evidence of specific improvement made based on the testing. No lorem ipsum anymore!! Accurate Data |  |
| applying user experience principles to improve the quality of the digital media outcome. | Good HCI. Eg. generally shows “consistency and standards”, and/or “help and documentation” by having an about page. No “annoyances” |  |
| **Excellence** |  |  |
| iterative improvement throughout the design, development and testing process to produce a high-quality outcome | Must be high quality. No obvious errors or inaccurate data. |  |
| using efficient tools and techniques in the outcome’s production. | Kept a good log, used github to keep backups and record of work, lots of commits on github. |  |

## 

## AS91906 - Use complex programming techniques to develop a computer program

**Credits:** 6 (Internal)

**NZQA:**  <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91906.pdf>

| **Achieved** | **Evidence** |  |
| --- | --- | --- |
| writing code for a program that performs a specified task | Applications works as intended. |  |
| using complex techniques in a suitable programming language | “programming or writing code for a graphical user interface (GUI)” flask requires that  “reading from, or writing to, files or other persistent storage ”- the sqlite file  “using additional non-core libraries”- using Flask. |  |
| setting out the program code clearly and documenting the program with comments | Some code comments, code layed out relatively cleanly with constants/variable at the top, routes and functions after. |  |
| testing and debugging the program to ensure that it works on a sample of expected cases. | Testing table shows the basic web page functionality was tested. |  |
| **Merit** |  |  |
| documenting the program with appropriate names and organised comments that describe code function and behaviour | Good variable and function names. Good and plentiful code comments. |  |
| following conventions for the chosen programming language | PEP8 Followed. Use Pylint or similar and check linting. |  |
| testing and debugging the program effectively to ensure that it works on a sample of both expected cases and relevant boundary cases. | Lots of testing including request for non-existent id returning an error 404 page. |  |
| **Excellence** |  |  |
| ensuring that the program is a well-structured, logical response to the specified task | Enhanced program, e.g. functions that aren’t routes to make the code cleaner (eg query\_db). Neat and logical code. |  |
| making the program flexible and robust | Great structure making code easy to extend. Unbreakable code. Eg. Error-404 and error 505 handlers |  |
| comprehensively testing and debugging the program. | Unbreakable code with a lot of tests done for every testable element of the program including unexpected input like a non existent route or page returning custom error 404 handler |  |

## 

## AS91907- Use complex processes to develop a digital technologies outcome

**Credits:** 6 (internal)

**NZQA:**  <https://www.nzqa.govt.nz/nqfdocs/ncea-resource/achievements/2019/as91907.pdf>

| **Achieved** | **Evidence** | ✔ |
| --- | --- | --- |
| using recognised and appropriate project management tools and techniques to plan the development of a digital technologies outcome | Github used. Goals defined.  Good commit messages. |  |
| decomposing the digital technologies outcome into smaller components | Evidence by regular commits with good messages describing the completed task |  |
| trialling the components of the outcome | Completed tasks committed should have been tested |  |
| testing that the digital technologies outcome functions as intended | The whole application should be tested between sprints |  |
| addressing relevant implications. | the application meets key relevant implications like copyright, IP (correctly sourced and attributed imagery), Privacy (eg passwords hashed) etc. May have highlighted this in final Relevant Implications Task |  |
| **Merit** |  |  |
| effectively using project management tools and techniques to manage development, feedback and/or collaborative processes | Plenty well named commits done regularly (eg not a rush at the end or large period of inactivity seen in git log) |  |
| effectively trialling multiple components and/or techniques | As above with larger number of small components tested and committed |  |
| effectively using information from testing and trialling to improve the functionality of the digital technologies outcome. | Specific examples given in end of sprint reviews showing how they found something out in testing and improved their outcome as a result. Can be observed too. |  |
| **Excellence** |  |  |
| synthesising information gained from the planning, testing and trialling of components | In the final summary and discussion gives specific information about testing and trialling and how it helped them to make their app better. |  |
| discussing how this information led to the development of a high-quality digital technologies outcome. | There is a section at the end specifically asking this question. In depth high quality answer. |  |

Final grades will be decided using professional judgement based on a holistic examination of the evidence provided against the criteria in the Achievement Standard.