# Assessment Handbook – Project 2

Your go to guide for all things assessment related including project ideas, assessment criteria and submission details.

# Table of Contents

[Portfolio Project 2 - Javascript Essentials](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#project-2)

[Readme](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#readme)

[Assessment Policy](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#assessment-policy)

[Completing your portfolio project](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#completion)

[Grade Notification](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#notification)

[Plagiarism](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#plagiarism)

[Appeals](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#appeals)

[Resubmissions](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#resubmissions)

[Final Grade Calculation](https://code-institute-org.github.io/5P-Assessments-Handbook/portfolio2-flight-check.html#final-grade)

# Portfolio Project 2

# JavaScript Essentials

Project purpose: Presentation of interactive data

In this project, you'll build an interactive front-end site. The site should respond to the users' actions, allowing users to actively engage with data, alter the way the site displays the information to achieve their preferred goals.

# Main Technologies

Required: HTML, CSS, JavaScript.

Optional: Google Charts

# Project Ideas

You can either choose to build the website based on Project Idea 0 or take inspiration from the other example ideas below.

## Project Idea 0

Bring your own idea(s) to life, based on providing value to users to address a specific real or imagined need. Use the relevant project assessment criteria as a guide to the minimum required functionality.

## Project Example Idea 1

**Build a Rock, Paper, Scissors, Lizard, Spock Game**

**External user’s goal:**

* The site’s users want to play an online game that has elements of chance.

**Site owner's goal:**

* The site’s goal is to provide a challenging game with increasing levels of difficulty to entertain online users.

**Potential features to include:**

* Pattern matching functionality
* A limited number of tries before the game is over
* A score tracking system
* The ability to play against the computer

## Project Example Idea 2

**Build an Online Quiz**

**External user’s goal:**

* The site’s users want to play an online quiz to test general/specific knowledge such as sports/movies/literary knowledge or personality assessment.

**Site owner's goal:**

* The goal of the site is to provide a challenging quiz to entertain online users.

**Potential features to include:**

* A score tracking system
* Images to support the correct/incorrect answers
* User can create a username
* Feedback

## Project Example Idea 3

**Build a NaturalWorld/Sport/Finance/Geopolitical Trends Website**

**External user’s goal:**

* The site’s users are interested in gaining insights into data relationships representing particular events or activities.

**Site owner's goal:**

* The site’s goal is to provide easily digestible data representations that lead to user actionable insights.

**Potential features to include:**

* Charting functionality to represent the data easily digestible manner
* Supporting facts surrounding the data and any insights gleaned from the data

# JavaScript Essentials Assessment Criteria

# Learning Outcomes

|  |  |
| --- | --- |
| LO1 | Design an interactive Front-End web application using HTML and CSS and JavaScript based on the principles of user experience design, accessibility and responsivity |
| LO2 | Test a front-end web application through the development, implementation and deployment stages |
| LO3 | Deploy a Front-End web application to a Cloud platform |
| LO4 | Maximise future maintainability through documentation, code structure and organisation |
| LO5 | Demonstrate and document the development process through a version control system such as GitHub |
| LO6 | Implement Front-End interactivity, using core JavaScript, JavaScript libraries or frameworks |

# All Pass criteria must be achieved for a pass to be awarded.

**LO1** Design an interactive Front-End web application using HTML and CSS and JavaScript based on the principles of user experience design, accessibility and responsivity

|  |  |
| --- | --- |
| 1.1 | Design a web application that meets accessibility guidelines, follows the principles of UX design and presents a structured layout and navigation model, and meets its given purpose |
| 1.2 | Design a website that meets accessibility guidelines (e.g. contrast between background and foreground colours to cater for the visually impaired) Optionally, add alt text for non-text elements |
| 1.3 | Design the organisation of information on the page following the principles of user experience design (headers are used to convey structure, information is easy to find due to being presented and categorised in terms of priority) |
| 1.4 | Ensure that foreground information is never distracted by backgrounds |
| 1.5 | Include graphics that are consistent in style and colour |
| 1.6 | Design interactivity for a web application that lets the user initiate and control actions, and gives feedback |
| 1.7 | Write custom JavaScript, HTML and CSS code to create a responsive front-end web application consisting of one or more HTML pages with significant interactive functionality |
| 1.8 | Write JavaScript code to produce relevant responses to user actions |
| 1.9 | Implement an interactive web application that incorporates images or graphics of usable resolution, legible, unobscured text, consistent styling, undistracted foregrounds |

**LO2** Test a front-end web application through the development, implementation and deployment stages

|  |  |
| --- | --- |
| 2.1 | Write JavaScript code that passes through a linter (e.g. Jshint) with no significant issues. |
| 2.2 | Write custom HTML code that passes through the official W3C validator with no issues. |
| 2.3 | Write custom CSS code that passes through the official (Jigsaw) validator with no issues |
| 2.4 | Code all external links to open in a separate tab when clicked |
| 2.5 | Use CSS media queries across the application to ensure the layout changes appropriately and maintains the page's structural integrity across device screen sizes. |
| 2.6 | Use Semantic markup to structure HTML code |
| 2.7 | Present the finished website with clearly understandable site-specific content, rather than Lorem Ipsum placeholder text |
| 2.8 | If used, implement clear navigation to allow users to find resources on the site intuitively. |

**LO3** Deploy a Front-End web application to a Cloud platform

|  |  |
| --- | --- |
| 3.1 | Deploy a final version of the code to a cloud-based hosting platform (e.g. GitHub Pages) and test to ensure it matches the development version |
| 3.3 | Use Git & GitHub for version control of an interactive web application up to deployment. |
| 3.4 | Remove commented-out code before pushing final changes to version control and deploying. |
| 3.5 | Ensure that there are no broken internal links |
|  |  |

**LO4** Maximise future maintainability through documentation, code structure and organisation

|  |  |
| --- | --- |
| 4.1 | Write a README.md file for the web application that explains its purpose, the value that it provides to its users, and the deployment procedure. |
| 4.2 | Insert screenshots of the project features, give a brief description of what each feature does and explain its value to the user. |
| 4.3 | Attribute all code from external sources to its original source via comments above the code and (for larger dependencies) in the README. |
| 4.4 | Clearly separate and identify code written for the website and code from external sources (e.g. libraries or tutorials) |
| 4.5 | Organise HTML, CSS and JavaScript code into well-defined and commented sections |
| 4.6 | Place CSS code in external files linked to the HTML page in the HEAD element. |
| 4.7 | Place JavaScript code in external files linked to the HTML page just above the closing body tag. |
| 4.8 | Write code that meets at least minimum standards for readability (consistent indentation, blank lines only appear individually or, at most, in pairs) |
| 4.9 | Name files consistently and descriptively, without spaces or capitalisation to allow for cross-platform compatibility. |
| 4.1 | Group files in directories by file type (e.g. an assets directory will contain all static files and may be organised into sub-directories such as CSS, images, etc.) |

**LO5** Demonstrate and document the development process through a version control system such as GitHub

|  |  |
| --- | --- |
| 5.1 | Use consistent and effective markdown formatting that is well-structured, easy to follow, and has few grammatical errors, when writing a README file. |

LO6 Implement Front-End interactivity, using core JavaScript, JavaScript libraries or frameworks

|  |  |
| --- | --- |
| 6.1 | Write JavaScript functions that correctly implement compound statements, such as if conditions and loops. |
| 6.2 | Write code that intelligently handles empty or invalid input data. |
| 6.3 | Implement appropriate working functionality for all project requirements. |
| 6.4 | Write code that does not generate internal errors on the page or in the console due to user actions. |
| 6.5 | Organise code and assets files in directories by file type. |

All Pass criteria must be achieved for a pass to be awarded.