|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assessment # and title | | | | C-WT-AT2-POR-Phase-2 | | | | Web Technologies Portfolio Phase 2 | | | | | |
| **Lecturer name** | | | | *Adrian Gould* | | | | | | | | | |
| **Student name** | | | | *Alexander Glover* | | | | | | | | | |
| **Student ID number** | | | | *20085553* | | | | | | | | | |
| **Telephone contact number** | | | | *0438725598* | | | | | | | | | |
| **Email** | | | | *20085553@tafe.wa.edu.au* | | | | | | | | | |
| **By completing and submitting this signed form to my lecturer, I am stating that:**   1. The attached submission is completely my own work 2. I have correctly cited all sources of information used in this work (if required) 3. I have kept a copy of this assessment (where practicable) 4. I understand a copy of my assessment will be kept by the NMTAFE for their records 5. I understand my assessment may be selected for use in the NMTAFE’s validation and audit process to ensure student assessment meets requirements | | | | | | | | | | | | | |
| **Student Signature** | | *AG* | | | | | | | **Date** | | *19/09/2023* | | |
| Assessors please note: Where verbal clarification has been sought from a student to gather additional assessment evidence from an assessment item, question/s and response/s must be recorded, signed, and dated by the assessor, against the relevant assessment item/s. | | | | | | | | | | | | | |
| NB: Feedback will be given via Blackboard when possible. | | | | | | | | | | | | | |
| **Submission 1** | | Result | Satisfactory / Not Yet Satisfactory | | | | | | | Date | | |  |
| *To satisfy requirements for this assessment, you need to complete the following:* | | | Feedback to student… | | | | | | | | | | |
| **Submission 2** | | Result | Satisfactory / Not Yet Satisfactory | | | | | | | Date | | |  |
| *To satisfy requirements for this assessment, you need to complete the following:* | | | Feedback to student… | | | | | | | | | | |
| **Student Feedback** | | | Feedback from student… | | | | | | | | | | |
| Lecturer Signature | | |  | | | | Student Signature | | | | | |  |
| **Assessment type (🗹):** | | | | | | | | | | | | | |
|  | Questioning (Oral/Written) | | | |  | 3rd Party Report | | | | | |  | Practical Demonstration |
|  | Other – Project/Portfolio (*please specify on the right):* | | | |  | | | | | | | | |

[Assessment Steps 10](#_Toc144130266)

[Complete the Front Page 10](#_Toc144130267)

[Setting Up for Phase 2 11](#_Toc144130268)

[Implement Reset/Normalise CSS on the Page 12](#_Toc144130269)

[Implement CSS to create the required Base Mobile Layout. 14](#_Toc144130270)

[Implement CSS to create the Base Desktop Layout. 15](#_Toc144130271)

[Implement CSS to Implement the Mobile Article Layout 16](#_Toc144130272)

[Implement CSS to Implement the Desktop Article Layout. 17](#_Toc144130273)

[Implement CSS for Aside Design 18](#_Toc144130274)

[Identify CSS3 Techniques/Features 19](#_Toc144130275)

[Validating the CSS 21](#_Toc144130276)

[Testing the page 21](#_Toc144130277)

[Demonstrate to Assessor 23](#_Toc144130278)

[Submission of Portfolio Work 23](#_Toc144130279)

[Appendix A: Code Style Guidelines 24](#_Toc144130280)

[Appendix B: Naming Conventions 25](#_Toc144130281)

[HTML, CSS and JS Files and folders 25](#_Toc144130282)

[File extensions 25](#_Toc144130283)

[Appendix C: Site Folder Structure 26](#_Toc144130284)

[Appendix D: Sample Wireframes 27](#_Toc144130285)

[Basic Layouts for Desktop and Mobile 27](#_Toc144130286)

[Basic Layouts with Articles for Mobile and Desktop 28](#_Toc144130287)

[Header & Navigation Layout Options 29](#_Toc144130288)

[Footer Layout Options (Desktop and Mobile) 29](#_Toc144130289)

[Article Layouts & Aside Layout 30](#_Toc144130290)

[Image Layout 30](#_Toc144130291)

[Aside Content Layout 31](#_Toc144130292)

|  |
| --- |
| Assessment Due Date |
| This assessment is split into components that have several due dates.  This component of the Portfolio of Work is due:   * Week 9 at 17:30 (5.30PM) on day of scheduled lecture   These are also shown in the Learning and Assessment Plan.  Also, refer to Blackboard for most accurate dates, which may alter due to unforeseen circumstances.  We also will endeavour to update these document(s) at the same time. |
| Required Resources | |
| The base requirements this assessment task, are listed below. They are listed as common (for both PC and Mac), and for the individual operating systems.  We presume that all assessment work is completed on a **PC** with the software as specified. This is to reduce configuration issues affecting the successful completion of the assessment item.  Whilst other applications and operating systems may be used, we are unable to give extensive support to ensure your environment is working as expected. Common:  * Access to Office 365 & Microsoft Word * WebStorm, PhpStorm * Access to Figma.com for diagrams * Access to TAFE Web Site Hosting (located on dev.ScreenCraft.net.au) * WinSCP for deployment of site * Git for version control, GitHub account for  PC:  * Web Browsers (Must have **at least** TWO different rendering engines)   + Chromium based: Edge, Chrome,   + Firefox   + Opera  Mac:  * Web Browsers (Must have **at least** TWO different rendering engines)   + Chromium based: Edge, Chrome,   + Firefox   + Opera   + Safari   ***Use of some of these items may not be used in this part of the assessment task.*** |
| Optional Requirements |
| An application to provide web services such as web server, database and more.   |  |  | | --- | --- | | Windows | MacOS | | Laragon  AMPPS  XAMPP  Docker | MAMP  AMPPS  XAMPP  Docker |   The lecturers will not be familiar with all the options listed, and it will be up to the student to investigate how to set up and use any alternatives.  The use of alternative IDEs is allowed (such as VS Code), but not supported. |
| Instructions |
| Follow the steps listed in this assessment item.  Submission of the documentation, code, and associated items is at the end of each part of the portfolio.  Provide evidence in the form of scanned documentation, completed assessment documents, screenshots, screencasts and other formats as required in this assessment. |
| Scenario |
| You are currently working as an intern for a small Perth-based start-up company called *Incredibly Obvious Technologies*.  The company is challenging you to work on a project to assess your skills as you develop them.  The project will build up in phases from the base content by adding semantic HTML, CSS to provide layout and suitable images to provided content.  The final section of the project is to add JavaScript to solve a specific problem.  You will find the project specifications; an outline of the phases of development and other details given below. |
| Portfolio Specifications and Development Outline |
| The following are the specifications for the project you are undertaking as part of this portfolio of work.  Each part of the project corresponds to another part of the portfolio, with each part submitted according to provided deadlines.   * Phase 0:   + Setting Up – not assessed but contains the required steps to initialise the portfolio. * Phase 1:   + Implement and test semantically and structurally correct HTML for the provided layout design and provided content. * Phase 2:   + Implement and test the required CSS to provide the layout as provided. * Phase 3:   + Add additional content and styling to amend the project to the given specifications. * Phase 4:   + Add and test a JavaScript solution for a predefined problem to the page.   Each phase will be version controlled and uploaded to the ScreenCraft Development Server.  Phase 0: Setting Up  In this phase, you are to complete the setting up of the portfolio.  This is submitted as part of Portfolio Part 1.  Phase 1: Text to Semantic HTML  In this development phase you will take the provided text and construct a suitable HTML page structure to suit it and the provided wireframes.  NO CSS is to be added.  Phase 2: Adding Structural Style  In phase 2 you now add the required CSS to style the site appropriately.  You are expected to style the site using Black, White, and Grey (#333333, #666666, #999999 and #cccccc only).  This is to demonstrate the use of CSS to lay out the site without any extra distractions such as colour and images. Use the provided layout wireframes as a guide to how it should be visually structured.  You will validate your custom CSS (not the CSS reset) using the relevant W3C validator.  Evidence of valid CSS will be submitted in the Phase 2 evidence folder.  Screenshots of the final product for this phase will also be included.  Phase 3: Pwity Pweese!  Aside: Imagine “Tweety Pie” saying Pretty Please  In this phase of the project, you are to implement the colour, typography and images to complete the basic page.  You will also amend the content by adding an extra story from phase-3.txt and an image.  In the “aside” you will add the content from phase-3-aside.txt  You will use the W3C validator to ensure your HTML is still valid.  Evidence of valid HTML will be submitted in the Phase 3 evidence folder.  Screenshots of the final product for this phase will also be included.  Once completed check in all your code to your local repository and push it to your private remote.  Phase 4: Shaking Things Up!  In the final phase, you will be implementing a simple problem that uses JavaScript.  The output of this JavaScript will be shown in the Aside area, in position 1.  The styling of the output must be in keeping with the rest of your choices of typography, colour, and other aspects of the site implementation you have undertaken.  Position 1 in the aside must contain a heading, the output from the JavaScript, and a footer. The footer of the section at position 1 must contain your name. |
| Site Structure Requirements |
| All files must apply the naming conventions outlined in Appendix B: Naming Conventions.  Code must apply suitable code style as advised in Appendix A: Code Style Guidelines.  The project has a defined structure outlined in the Appendix C: Site Folder Structure.  The primary page in each phase must be named index.html.  Content details and additional steps will be made more explicit in each subsequent part of the portfolio.  Ensure you DO NOT complete more than one phase of development at any time. |
| General Instructions |
| Complete each step of the document in the order given.  All work must be completed at the CLI, except for starting the development environments.  When a step requires you to confer with the lecturer, ensure you do so, and make notes as required in spaces provided in this document.  Clarifying Requirements  If you require requirements to be clarified, please follow the instructions below:  Send an email using your TAFE email address to:  [f2f@screencraft.net.au](mailto:f2f@screencraft.net.au)  You MUST include the following subject:  C4 Prog: Web Tech Portfolio Phase x  Ensure you replace the x with the phase you are asking about. In this case 2. |
| Answering Questions |
| When a step includes a question, you must attempt to answer it.  There is a minimum and maximum number of words to use for each answer.  Any step that requires answers to be provided will have a space in this document immediately after each step. The answer space will expand with the content you type or images you paste.  Resize images to fit the space provided, ensuring they are still legible.  Do not include a whole screenshot, just the required details.  If a step has more than one question, these maxima and minima are a total for all the questions in that specific step.  All answers must be in complete sentences unless indicated.  *If required, make sure to add any code you’ve written in a separate file to your submission.*  *DO NOT put long pieces of code (over 10 lines) in this document.* |
| Sources of Information |
| In industry, it is good practice to keep track of where information was obtained. This is especially true if it is a written document, or even code.  If you answer any questions using information from web sites, please include the site name and URL (Web site address) after the answer.  Likewise, include the title and author for books and magazine articles.  For example:   * RS Electronics Ltd:  <https://au.rs-online.com/> * Slack API Documentation, Users List Method:  <https://api.slack.com/methods/users.list> |
| Code Storage |
| We advise that you create a private GIT repository on GitHub and use this to store a copy of your work.  This assessment has instructions on setting up version control and a remote (private) repository. Do not initialise any version control until told to do so.  To ensure you have a recent copy of your code you **must** use **AT LEAST TWO** of the following methods, of which a version controlled copy is REQUIRED:   * Private Git repository,   and one of:   * Cloud Storage (OneDrive within your college Office365), and/or * Keep a copy on a USB thumb drive.   Backing up to One Drive or to USB is best done by compressing the project folder up before copying to either OneDrive or USB. |
| Code Style |
| Please see **Appendix A: Code Style Guidelines** for details on the code style(s) to use for the assessment. |
| Professional Code and Documentation Requirements |
| This document contains appendices that cover the professional requirements when creating files, code, project structures and more.  These requirements must be adhered to.  No external frameworks are allowed in this part of the portfolio.  This means that you may not employ frameworks such as Bootstrap, jQuery, Tailwind CSS, et al in your work. |
| Assessment Structure |
| To complete this assessment, you will be asked to complete four parts.  Do not start subsequent parts until completing and submitting previous part.  These parts include:   * Part 1: Development Phases 0 & 1, Setting up and creating semantic markup. * Part 2: Development Phase 2, Adding basic styling to create desired page layout. * Part 3: Development Phase 3, Adding further style, content, and imagery to enhance page. * Part 4: Development Phase 4, Adding JavaScript to solve a given problem and display result on page. |
| Assessment/Development Process |
| In the steps that are contained in this document, you will design several pages, and create the layout for the home page, with your code being version controlled as you progress.   * Step 1: Set up. * Step 2: Integrate a CSS reset/normalise file into the page. * Step 3: Create the required CSS to facilitate a layout that is similar to the one provided. * Step 4: Test / Validate your CSS (not the reset/normalise file). * Step 5: Compress and Submit your work. |

# Assessment Steps

Complete the steps in the order given.

| **STEP** | **Task to perform** |
| --- | --- |
| 00 | Complete the Front Page Make sure you have filled out the front page of this document.  Familiarise yourself with the content and document your progress in this assessment. This means, **READ** the **WHOLE** assessment **AT LEAST ONCE** before starting any work.  Make notes on the requirements of this assessment, as details appear as you progress through and are not given in one location.  This is very similar to how details emerge during the continuous development interviews and meetings with a client.  At any stage during this assignment, you may consult the stakeholder(s) or their representative(s).  Max Sentences: N/A |
|  | Intentionally left blank |
| 01 | Setting Up for Phase 2 In the previous phase of this assessment, you created suitable semantic and structural HTML to wrap the content provided.  This was saved in a folder named phase-1.  Either:   * Duplicate the phase-1 folder and call it phase-2, or * Copy the contents from the phase-1 folder into an existing phase-2 folder.   Using version control, you will now add the new code to version control, and commit to the local repository:   * Add the duplicated files to version control. * Commit these files to your local repository with a commit message based on the one below: * “feat: start of site phase-2”   Push the code to your private remote repository.  Max Words: N/A |
| 02 | Implement Reset/Normalise CSS on the Page Investigate the 5 CSS reset/normalise methods shown below.  Select one of these to use in your portfolio work.  To simplify your work, we have provided the files in a separate compressed file (CSS-Resets.zip) that you will expand and move the CSS files into the root assets folder.  Your options are (link to details, filenames in compressed file):  <https://elad2412.github.io/the-new-css-reset/> (elad-reset.css)  <https://meyerweb.com/eric/tools/css/reset/> (meyer-reset.css)  <https://css-tricks.com/modern-normalize/> (modern-normalise.css)  <https://csstools.github.io/sanitize.css/> (csstools-sanitise.css,  -typography.css, -forms.css)  <https://necolas.github.io/normalize.css/> (necolas-normalise.css)  Once you have selected the CSS file, you should link to file that is contained in the root assets/css folder.  Add the changed/added files to version control.  Use a commit message based on the one below (note this is incomplete):   * “feat: add CSS to …”   After this first line, you are able to add further details to your commit messages.  To do this at the command line, as your lecturer for guidance. In the WebStorm IDE it is as simple as adding a blank line after the 1st line of the commit, then each note on a separate line.  Push the code to your private remote repository. |
| A02 | Which of the CSS Reset/Normalise files did you select and why? (2 – 4 sentences)  I just Necolas-normalize.css. I chose this css reset/normalisation type as I understood it best and was easy to read. Using this style also worked well with my current css file. What is the HTML you created to include the CSS file? (Include a code snippet)   Where is the above HTML code located in your phase-2/index.html file? (Include a code snippet)   What is the Commit message you used? [adding phase-1 html file to phase-2](https://github.com/GooseTafe/web_tech/commit/dae755f05d8dcac5bbdea6e8b7d80a9f837d5f07) |
| 03 | Implement CSS to create the required Base Mobile Layout. You will implement a mobile-first layout.  In this step we want you to create the header main area and footer.  The Aside and Navigation areas from the desktop layout will be hidden.  The Navigation will be shown when the “hamburger menu” is clicked.  You DO NOT implement this action in this stage.  The most important part of this stage is demonstrating the ability to create a desktop-based layout with valid CSS code, and a very simple mobile friendly version of the layout.   Implementing steps You must implement the layout in the following order, with EVERY step being committed to version control using Semantic Commit Messages:   1. Header 2. Main 3. Footer   Remember: Navigation and Asides are to be hidden in the mobile view.  An example commit message for the Header could be:   * feat: site mobile header – add CSS to lay the site header |
| 04 | Implement CSS to create the Base Desktop Layout. The layout we present in the image below is for a desktop browser and is to be used as the layout for this type of device.  The most important part of this stage is demonstrating the ability to create a desktop-based layout with valid CSS code, and a very simple mobile friendly version of the layout.   Implementing steps You must implement the layout in the following order, with EVERY step being committed to version control using Semantic Commit Messages:   1. Header 2. Navigation 3. Main 4. Footer   An example commit message for the Header could be:   * feat: desktop base header – add CSS to lay the site header |
| 05 | Implement CSS to Implement the Mobile Article Layout In this stage you will now add the Articles to the Mobile layout.  The layout is as shown below.  One important item is that the Footer will be fixed to the bottom of the articles (as in after all the articles are shown), the Header will stay in place.   Implementing steps You must implement the layout in the following order, with EVERY step being committed to version control using Semantic Commit Messages:   1. Article(s)   An example commit message for the Header could be:   * feat: mobile article – add CSS for articles for mobile design |
| 06 | Implement CSS to Implement the Desktop Article Layout. In this stage you will now add the Articles to the Desktop layout.  As with the mobile layout, the Footer will be fixed to the bottom of the articles (as in after all the articles are shown).  The header and navigation, in the desktop layout will be fixed to the top of the screen and the content will slide behind them.  You may layout the desktop version in any form, the only requirements is that the aside MUST be on the right.   Implementing steps You must implement the layout in the following order, with EVERY step being committed to version control using Semantic Commit Messages:   1. Article(s)   An example commit message for the Header could be:   * feat: mobile article – add CSS for desktop design |
| 07 | Implement CSS for Aside Design In this stage you will now add the Aside to the DESKTOP layout.  The layout is as shown below.  The aside is ONLY visible for the Desktop layout.   Implementing steps You must implement the layout in the following order, with EVERY step being committed to version control using Semantic Commit Messages:   1. Aside   An example commit message for the Header could be:   * feat: desktop aside – add CSS for aside layout in desktop design |
| 07 | Identify CSS3 Techniques/Features During your development you will have used different CSS3 techniques to lay out the content of the site.  Identify FOUR techniques and/or features that you employed in styling the site layout.  Why did you use these techniques and/or features?  Add your answers to the answer section A07. Important: The articles may be presented as the same size or like masonry work, with different heights depending on the content.  This portfolio stage is about reproducing a layout, complete with the site header, navigation and footer area positions.  The main area should be between **70 – 90%** of the width of the screen. |
| A07.0 | What CSS features did you use in your code to create your layout? Identify four key items and give one to two sentences to explain why you chose each one.  The key features that I selected and used were:   1. using css selectors 2. using grid layout 3. using Pseudo-classes 4. min/max dimensions |
| A07.1 | CSS Feature 1 (1 – 2 sentences)  By using selectors i.e. html or body or ID selectors #header or #nav to target specific HTML sections |
| A07.2 | CSS Feature 2 (1 – 2 sentences)  Using a grid layout to create the article and aside layout in the main section |
| A07.3 | CSS Feature 3 (1 – 2 sentences)  Using Pseudo-classes i.e. :hover to create a conditional style i.e if a link is hovered over change colour |
| A07.4 | CSS Feature 4 (1 – 2 sentences)  Setting min and max dimensions so that the styling doesn’t go above or below the set dimensions |
| 08 | Validating the CSS Use the WC3 Validation engine to verify your CSS is valid.  Copy the evidence for validity into the evidence folder inside phase-2.  Screenshots of the final product for this phase will also be included as evidence in the above folder.  Add the evidence folders to the “.gitignore” file in the repository.  Once completed add your updated code to version control.  Check in all your code to your local repository (use the following as a base for your message: “feat: implement layout for…”  Push these changes to your private remote. |
| A08 | Add one or more screenshots of the validation results for your CSS code. |
| 09 | Testing the page During the previous steps, you have created and updated a single web page.  During this it is possible that you have been checking what it looked like on at least one web browser.  You are now to verify that the site still works on at least two browsers that use different rendering engines.  Take a screenshot of the page in the two browsers.  Paste these into the two sections below.  Make sure the images are no larger than 14cm and no smaller than 12cm wide. |
| A09.1 | Which browser is this screenshot from? Google Chrome Screenshot |
| A09.2 | Which browser is this screenshot from? Mozilla Firefox Screenshot |
| A09.3 | Which browser is this screenshot from? Brave Screenshot |
| 10 | Demonstrate to Assessor Demonstrate to the assessor.  Obtain their feedback, update your work as required based on their feedback, ensuring all changes are notes as semantic commits using the “fix:…” tag at the start.  Prepare for phase 3 of the site development. |
| END | Submission of Portfolio Work To submit the portfolio, do the following:   * Save this document with your answers. * Export the pages of your Figma file to PDF(s) * Compress the portfolio folder using 7-Zip * Open Blackboard, locate and open the “AT2-PT1 Por Phase 2” item * Upload this completed assessment document. * Upload the PDF and Word document(s). * Upload the compressed site files. * Click submit.  All answer documents MUST be submitted in Microsoft Office 365’s Word format. It is important that the PDF and Word documents are separate from the portfolio compressed folder as this makes it much easier for assessment. |

# Appendix A: Code Style Guidelines

The following guidelines should be applied to your code as it is developed.

Many may be applied via the use of PhpStorm, PyCharm or similar plugins and code formatting.

### PHP Code (General)

Please refer to the PHP PSRs:

* <https://www.php-fig.org/psr/psr-1/>
* <https://www.php-fig.org/psr/psr-12/>
* <https://www.php-fig.org/psr/psr-4/>

### Applications Built with Laravel

Please refer to the following articles:

* <https://dev.to/lathindu1/laravel-best-practice-coding-standards-part-01-304l>
* <https://dev.to/lathindu1/laravel-best-practice-coding-standards-part-02-a40>

### HTML Code

Please refer to the Google Style guide:

* <https://google.github.io/styleguide/htmlcssguide.html>

Note that PhpStorm and WebStorm will format code to their own standard, which is acceptable.

### JavaScript

Refer to the Google JS Style guide:

* <https://google.github.io/styleguide/jsguide.html>

Note that PhpStorm and WebStorm will format code to their own standard, which is acceptable.

### Python Code

Your code will follow the PEP 8 standard.

PyCharm will format code very close to the PEP-8 standard.

### JSON Code

JSON should be formatted in an appropriate manner.

*Readability Counts   
- Zen of Python*

# Appendix B: Naming Conventions

## HTML, CSS and JS Files and folders

* No spaces.
* Alpha-numeric characters only except minus/hyphen (-) and full stop (.).
* All lowercase letters.
* Use hyphens (dash) between words.

## File extensions

The following file extensions must be used:

|  |  |
| --- | --- |
| HTML pages | .html |
| JS | .js |
| CSS | .css |
| PHP | .php |
| Node | .js |
| React | .jsx |
| Python | .py |
| JSON | .json |
| XML | .xml |

# Appendix C: Site Folder Structure

The diagram below outlines a folder structure for the portfolio from Phase 1 to Phase 4:

# Appendix D: Sample Wireframes

These layouts provide the options you will use in your work.

A basic Mobile First layout (Mobile Phone) and a basic Desktop (1024px wide screen) layout is provided, with options for various components of the page.

Each page has:

* Header
* Navigation
* Main
* Footer

## Basic Layouts for Desktop and Mobile

Desktop Layout on Right, Mobile on Left

 

## Basic Layouts with Articles for Mobile and Desktop

Desktop Layout on Right (3 options), Mobile on Left

 

 

## Header & Navigation Layout Options



## Footer Layout Options (Desktop and Mobile)



## Article Layouts & Aside Layout



## Image Layout



**Aside Section Layout**



## Aside Content Layout

