|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assessment # and title | | | | C-WT-AT2-POR-Pt3 Web Technologies: Portfolio Part 3 | | | | | | | | |
| **Lecturer name** | | | | *Adrian Gould* | | | | | | | | |
| **Student name** | | | | *Yoseph Campbell* | | | | | | | | |
| **Student ID number** | | | | *20085059* | | | | | | | | |
| **Telephone contact number** | | | | *Student to fill this section out* | | | | | | | | |
| **Email** | | | | *20085059@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** | | *Yoseph Campbell* | | | | | | **Date** | | *25/10/2022* | | |
| 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 8](#_Toc97544937)

[Complete the Front Page 8](#_Toc97544938)

[Setting Up for Development 9](#_Toc97544939)

[Data Structure Design 11](#_Toc97544940)

[Database and Collection Creation 12](#_Toc97544941)

[Creating Indexes 14](#_Toc97544942)

[Importing Data 15](#_Toc97544943)

[MongoDB Queries 1 16](#_Toc97544944)

[Updating Data 17](#_Toc97544945)

[Deleting Data 18](#_Toc97544946)

[Queries 2 20](#_Toc97544947)

[Submission of Portfolio Work 22](#_Toc97544948)

[APPENDIX A: CODE STYLE GUIDELINES 23](#_Toc97544949)

[Appendix B: WAWD Data Structure 24](#_Toc97544950)

[Appendix C: WAGS Data 25](#_Toc97544951)

|  |  |
| --- | --- |
| 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. Common:  * Access to Office 365 & Microsoft Word * PhpStorm or similar IDE  PC:  * Laragon (with following components):   + Apache,   + NginX,   + MariaDB 10+/MySQL 8+,   + MongoDB 4+,   + PHP 8.0+,   + NodeJS 14+,   + npm,   + composer 2+ * Web Browsers (Must have **at least** TWO different rendering engines)   + Chromium based: Edge, Chrome, …   + Firefox   + Opera  Mac:  * Docker * 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.*** |
| Assessment Due Date |
| This assessment is split into components that have several due dates:   * Week xx at 17:30 (5.30PM) on day of scheduled lecture (Part x) * Week xx at 17: 30 (5.30PM) on day of scheduled lecture (Part x) * Week xx at 17:30 (5.30PM) on the day of the scheduled lecture (Part x) * Week xx at 17:30 (5.30PM) on the day of the scheduled lecture (Part x) * Week xx at 17:30 (5.30PM) on the day of the scheduled lecture (Part x)   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. |
| 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 screenshots, screencasts and other formats as required in this assessment. |
| Important |
| 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. |
| Scenario |
| You are currently working as an intern for a small Perth-based start-up company called *Incredibly Obvious Technologies*.  ADD REMAINDER OF THE SCNEARIO DETAILS |
| General Instructions |
| Complete each step of the document in the order given.  Work will be conducted using your preferred development environment, but we recommend:   * ADD DETAILS OF RECOMMENDED SOFTWARE ETC   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. |
| Answering Questions |
| When a step includes a question, you must attempt to answer it.  Answers should be within the range provided (in sentences). For example, “1 and 5 sentences.” This range will cover individual parts of a question, unless indicated.  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.  Crop and/or Resize images to fit the space provided, ensuring they are still legible.  Do not include whole screenshots, just the required details, even if this means using two or more images.  **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 that is being produced as part of a product.  If you answer any questions, or use a code resource to assist in software production, using information from web sites, please include the site name and URL (Web site address) after the answer, or as a clear comment in the written code.  Likewise, when book, magazine, or other form of resource is used, include the title and author for books and magazine articles, or other suitable identification information.  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.  Alternatives include external or cloud-based data storage.  To ensure you have a recent copy of your code you must use AT LEAST one of the following methods:   * Private Git repository, and/or * 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. |

# Assessment Steps

Complete the steps in the order given.

| STEP | Task to perform |
| --- | --- |
| 00a | 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). |
| 00b | Setting Up for Development We recommend that all work for this and following portfolios are kept in:   * C:\Users\XXXXXXXX\Source\Repos * ~/Source/Repos   We then recommend that you keep this cluster’s work in a WebTech folder.  To create this folder use:  PC:   * cd %USERPROFILE% * mkdir –p Source\Repos\WebTech   Mac:   * cd ~ * mkdir –p Source/Repos/WebTech   Then use the command:   * cd Source/Repos/WebTech   All steps after this we will presume you will use this new folder as the starting point. |
| 00c | Create Portfolio Folder As stated above, we presume you are in the Source/Repos/WebTech folder.  Duplicate the previous portfolio, renaming the new copy to be:   * XXX-WebTech-22S1-AT2-Pt3   Remember to replace XXX with YOUR initials. For example, for Adrian is shown here:   * AJG-WebTech-22S1-AT2-Pt3   Make sure you have a ReadMe.md file as part of your root folder. See appendix … for the required details for the ReadMe.  If, at this point, you are not using version control, you are now required to add and commit all content into version control using Git.  Push the complete content to a PRIVATE repository on GitHub. |
| 00d | Setting Up Portfolio Content You will be working in your new copy of the portfolio.  Use the template page developed for the previous portfolio to host each of the exercises given in the following pages.  In the demos folder you need to create a folder for each demo.  Create and name three folders: demo-01, demo-02, demo-03.  Inside each folder create a new js folder with a .keep text file.  The demos that we create from this point will use the js folder for their specific JavaScript files. |
| 00e | Modify the “Demo” Page Open the index.html page from the demo folder.  In this page add a card that occupies the middle 1/3 of the page.  Add a header to the card with the text “**Demo page links**”.  In the body of the card, add three paragraphs with the following text:   * Demo 01: Times Table * Demo 02: Birthday Fun * Demo 03: Shopping Cart   Add links to each of the paragraphs that link to the folders demo-01, demo-02 and demo-03.  In the following stages we will add a new index page for each demonstration.  In the footer of the card, add “More demonstrations to come”.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
|  | The following page contains the first  of the demo problems to solve |
| 01a | JavaScript Demo #1 – Times Table In this demo you will show your ability to:   * Manipulate the DOM * Respond to a change in a drop-down selection * Follow requirements * Use version control * Perform calculations in JS  Preparation Make a copy of the template-base page, place it in the new demo-01 folder. Rename the file to: index.html  Make sure that this new page views correctly (it matches your base-template).  In the demo-01/js folder, create an empty JavaScript (JS) file called demo-01.js.  In the JS file make sure you add the header as required in Appendix…  To the page, now do the following:   * Add a card to the middle of the page that occupies ½ of the page width. * In the header add text that says “0 Times Table”. * In the footer, add a link back to the demos main page/folder. * In the body of the card add the following text, ensuring it is centred on the line: “000 x 000 = 000000”.   Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 01b | Times Table - JS To the page, now do the following:   * Add a card to the middle of the page that occupies ½ of the page width. * In the header add text that says “0 Times Table”. * In the footer, add a link back to the demos main page/folder. * In the body of the card add the following text, ensuring it is centred on the line: “000 x 000 = 000000”.   You have been asked to create a times table page, and to begin you are to display the 7 (seven) times table.  Write the JavaScript code to replace the “000 x 000 = 000000” placeholder with the display of the seven times table.  For example:  1 x 7 = 7  2 x 7 = 14  3 x 7 = 21  …  9 x 7 = 63  10 x 7 = 70  11 x 7 = 77  12 x 7 = 84  Test and make sure times table display works as expected.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 02 | Extending the Times Table - JS This step requires you to extend the times table to include a new feature.  Do the following:   * In the card header change the number to be a dropdown box that allows the user to select the times table to display. * Now add the JavaScript to use the value from this dropdown box for the times table. It must have the numbers 1 to 100 inclusive. * Finally, when the value in the dropdown box is changed, the times table must update as required.   For example, if we select the 11 times table then the display will look similar to this:  1 x 11 = 11  2 x 11 = 22  3 x 11 = 33  …  9 x 11 = 99  10 x 11 = 110  12 x 11 = 121  Test and make sure that the dropdown box and updated times table display works as expected.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 03a | JavaScript Demo #2 – Birthday Calculator – Set Up In this demo you will show your ability to:   * Manipulate the DOM (3b, 3c) * Work with dates and times in JavaScript * Perform calculations in JS * Follow requirements * Use version control  Preparation Make a copy of the template-base page, place it in the new demo-02 folder.  Rename the file to: index.html  Make sure that this new page views correctly (it matches your base-template).  In the demo-02/js folder, create an empty JavaScript (JS) file called demo-02.js.  In the JS file make sure you add the header as required in Appendix…  Create a card in the middle of the page, that has the following:   * Header Birthday Calculator * Body A form with a suitable method to enter a date * Footer Placeholder text on two lines   + You are YY years, MM months and DD days young   + NNN days until your birthday   The form should show three drop down boxes:   * Year (1900 to 2022), * Month (01-12 or Jan-Dec), and * Days (01-31)   Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 03b | Birthday Calculator – JS Now the page is set up you are to create the JavaScript to solve the following problem:   * Read the date from the field(s) from part 03a * Verify that the date is valid (remember to check the number of days in the month chosen) * Calculate how many days difference between today’s date and the entered date * Calculate the number of years (to 2 decimal places) difference between the entered date and today’s date * Display Happy birthday (in place of the DDD days to your birthday) if today’s day and month are the same as the user’s birthday day and month. * Update the footer to show the newly updated details   Test your code, and fix any bugs  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 03c | Birthday Calculator – JS Update Our next update to the Birthday calculator is to:   * Store a list of Birthday greetings using a list, object or other suitable method. * Randomly select a greeting and when “Happy Birthday” should be shown, use this random greeting instead. * Make sure the Language of the greeting is displayed in [ square brackets ] at the end of the same line, or on a line below (hint, you may use a <br>).   Test your code, and fix any bugs.  The greetings to be used are:   |  |  | | --- | --- | | Language | Happy Birthday | | English | Happy Birthday | | French | Joyeux anniversaire! | | German | Alles Gute zum Geburtstag! | | Italian | Buon compleanno! | | Welsh | Penblwydd hapus! | | Klingon | qoSlIj DatIvjaj | | Bulgarian | Chestit Rozhden den! | | Chinese | Shēngrì kuàilè! | | Danish | Tillykke med fødselsdagen! | | Finnish | Hyvää syntymäpäivää! | | Hindi | janmadin mubaarak! | | Japanese | Otanjōbiomedetōgozaimasu! | | Persian | تولدت مبارک! |   You may add more greetings. See: <https://reference.yourdictionary.com/other-languages/happy-birthday-in-100-different-languages.html>  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 04 | Update the Dropdowns The improvement to this problem is to do the following:   * When the user selects the month the number of days should update to show the correct range. * When it is a leap year, the number of days in February must be 29. * If the was selected before the month, if the value is invalid then the last day of the month should be selected.   Test your code works as required.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 05a | JavaScript Demo #3 – Shopping Cart – Set Up In this demo you will show your ability to:   * Manipulate the DOM (3b, 3c) * Work with dates and times in JavaScript * Perform calculations in JS * Follow requirements * Use version control * Use local storage  Preparation Make a copy of the template-base page, place it in the new demo-03 folder.  Rename the file to: index.html  Make sure that this new page views correctly (it matches your base-template).  In the demo-03/js folder, create an empty JavaScript (JS) file called demo-03.js.  In the JS file make sure you add the header as required in Appendix…  Now perform the following page layout tasks:   * Create a two-card layout, where they are side by side. * The left card will be the shopping cart with header of “Cart”, footer of “Total: $xx.xx” and the main area should contain “No items”, plus two buttons “Clear Cart” and “Purchase”. * The right card will contain a header of “Select products”, no footer, and the body will have a form containing a list of products, with a text input box with 0 in each input. * The products are shown in the appendix …   Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 05b | Shopping Cart – JS You are to implement the shopping cart.  When an item on the right side has a new value entered, the cart will either:   * Add the item to the left if it is not already present, or * Update the item quantity as required, or * Remove the item if the quantity is set to 0   Test and debug your code.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 05c | Shopping Cart – JS Next, the cart is to perform the following:   * When an item is updated/removed the cart will also update the total shown in its footer. * When “Clear cart” is pressed, the cart will be cleared, and the quantities on the right reset to 0.   Test and debug your code.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 05d | Create Index page The next stage of this phase of development is for the “Purchase” button.  When the “Purchase” button is pressed the following will occur:   * then the right hand side will then display a list of itmes purchased, the price for each, the quantity for each, and the total for that item (price \* quantity). * At the bottom of this Invoice, the total price will be shown, an extra row showing GST (10% of total), and then a last line showing the Total including GST.   Test and debug your code.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| 06 | Local Storage The final part of this assessment is to add local storage to the cart.  This means that when any action occurs the shopping cart should save the data into local storage.  If the user leaves the site (closes the browser), and returns later, the page should show the shopping cart and other details as it was left.  Test and debug your code.  Add and commit these latest updates into version control. Make sure you use a suitable message. |
| END | Submission of Portfolio Work To submit the portfolio, do the following:   * Save this document with your answers. * Open Blackboard, locate and open the AT2 Portfolio Task 1 assessment * Upload the completed assessment document. * Upload the exported JSON files (see the step “Export data”) to the same submission as separate files. * Upload the PDF and any other files. * Click submit.   All answer documents MUST be submitted in Microsoft Office 365’s Word format. |

# 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>

### JavaScript

Refer to the Google JS Style guide:

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

### Python Code

Your code will follow the PEP 8 standard.

### JSON Code

JSON should be formatted in an appropriate manner.

*Readability Counts   
- Zen of Python*

# Appendix 2: JavaScript File Header

# Appendix 3: ReadMe.md

The ReadMe.md file that is part of your project must contain the following information:

**#** PROJECT TITLE  
  
Short description of the project.  
  
This project forms part of the assessment for the Certificate IV in IT (Programming) as delivered at North Metropolitan TAFE.  
  
**##** Requirements  
  
This project requires the following to be installed:  
  
**-** web browsers ***(***at least two***)*-** a code editor or IDE ***(***WebStorm or Visual Studio Code***)*##** Copyright  
  
© Copyright 2022 YOUR NAME ***<***EMAIL\_ADDRESS***>***Copyright symbol ©: **`**ALT**`**+**`**0169**`** on numeric keypad  
  
**##** Licensing  
  
This project is only to be used in the form it is on this repository. No editing allowed.  
  
All uses must link back to this repository using ***[***PROJECT TITLE***](****https://PROJECT\_URL****)***.