# Student Version

| Section A – Program/Course details | | | |
| --- | --- | --- | --- |
| **Qualification code:** | ICT40120 | **Qualification title:** | Certificate IV in Information Technology (Web Development) |
| **Subject code:**  **Unit code:** | (CWEB4)   ICTWEB445  ICTWEB430  ICTWEB434 | **Subject title:**  **Unit title:** | (Dynamic Web Development)  Implement content management systems  Produce server-side script for dynamic web pages  Transfer content to websites |
| **Department name:** | Cyber Security and Digital Information Technology (CSDIT) | **CRN number:** |  |

| Section B – Assessment task details | | | |
| --- | --- | --- | --- |
| **Assessment number:** | 2 of 2 | **Semester/Year:** | 1/2024 |
| **Due date:** | Session 17 | **Duration of assessment:** | 6 sessions |
| **Assessment method** | Portfolio | **Assessment task results** | ☒Ungraded result (S/NS) |
| ☐Other: |

| Section C – Instructions to students |
| --- |
| **Task instructions:** |
| Learners are required to develop a solution for a client using Apache, PHP & MySQL, with an additional solution using WordPress.  This assessment task is divided into the following parts: Part 1 – Confirm Application Requirements  Part 2 – Design the Database Part 3A & B – Develop Dynamic Server & Custom CMS & Commercial Mini-CMS Exemplar Part 4 – Incorporate Additional Functionality Part 5 – Deployment (Optional) Part 6 – Test and Debug the Code Part 7 – Document Activities & Hand Over  **Learners are expected to submit the following documents:**   * The signed and completed Assessment Sheet; and * (x2) ZIP files containing the project file & mini project file. * This task is to be completed individually. * Using this assessment document, please read all questions in this document and record your answers in the document against each question. * To be satisfactory in this assessment task all answered questions must correctly meet the marking guide criteria. * Discuss with your assessor if you feel you require special consideration or adjustment for this task. * Learners can consult class learning material via Brightspace and other softcopy information, including information from the Internet. However, all answers must be in a learner’s own words. Where a quote is used the learner must cite the information source. * You must submit all required working files, documentation, and any other assets that you feel may be required in a zipped file, including the completed and signed coversheet. The assessment must be completed and submitted electronically to Brightspace by the due date. If this is not possible, you must contact your assessor to gain written approval for an alternative arrangement for submitting the assessment. * It is expected all documents will be completed and submitted electronically but if this is not possible, make alternative arrangements for submitting the documents with your assessor. * Leaner must contribute to and abide by organisational standards including intellectual property, privacy laws, and plagiarism and academic honesty. Further information is detailed at: https://holmesglen.edu.au/Students/Student-Resources/   ***See supporting documentation for further instructions.*** |

| Section D – Conditions for assessment | |
| --- | --- |
| **Conditions:**  Student to complete and attach Assessment Submission Cover Sheet to the completed Assessment Task. | |
| * This is an individual task. * You must meet all criteria listed in the marking guide to be satisfactory in this task. * You must submit all required working files, documentation, and any other assets that you feel may be required in a zipped file, including the completed and signed coversheet. The assessment must be completed and submitted electronically to Brightspace by the due date. If this is not possible, you must contact your assessor to gain written approval for an alternative arrangement for submitting the assessment. * If not successful within the enrolment period as per Holmesglen assessment procedure, you will be requested to resubmit within 7 days of receiving feedback. You will have the opportunity to resubmit if any part of the assessment is deemed unsatisfactory (you are permitted TWO (2) resubmission per assessment task). Resubmissions must be submitted by the resubmission due date provided by your teacher. * This task is open book. You may use the internet for research purposes only. All answers must be in your own words. Where a quote is used, you must cite the information source. * If you feel you require special allowance or adjustment to this task, please discuss with your assessor within one week of commencing this assessment. Any change to assessment arrangements must be reviewed by the Education Manager and approved by the Head of Department. * You can appeal an assessment decision according to the Holmesglen Assessment Complaints and Appeals Procedure. * You are expected to dedicate time to develop this assessment task both in and out of the classroom. * Leaner must contribute to and abide by organisational standards including intellectual property, privacy laws, and plagiarism and academic honesty. Further information is detailed at: https://holmesglen.edu.au/Students/Student-Resources/   The following technologies must be utilised: WordPress, MySQL, PHP Development tools should include but are not limited to: Visual Studio Code | |
| **Equipment/resources learners must supply:**  **Learners opting to BYOD laptop or intending to learn remotely will require access to:**  • An iMac or  PC/laptop with the following minimum specification: Quad-Core CPU, 16GB of RAM, 250GB of Storage, 2 GHz or faster processor, Windows 10 OS, or higher • Headset with microphone and webcam (if learning remotely) • Internet access  Access to applications used in learning is available through Holmesglen MyHorizon or can be downloaded via the link provided: • Brightspace (Learning Management System) - https://holmesglen.brightspace.com/ • 365 Microsoft office suite - https://portal.office.com • WebEx - https://holmesglen.webex.com/ • Google Chrome – recommended web browser • 7-Zip - https://www.7-zip.org/download.html • Visual Studio Code - https://code.visualstudio.com/ • FileZilla / cPanel services - https://filezilla-project.org/ • WAMP - https://www.wampserver.com/en/ • MAMP - https://www.mamp.info/en/downloads/ • Web server account  • GitHub Desktop: https://desktop.github.com • Apache - https://httpd.apache.org/ • PHP - https://www.php.net/ • MySQL - https://www.mysql.com/ • Wordpress - https://wordpress.com/ | **This is a blended learning course and as such a remote learner will access their own computer equipment as per the specification provided. The Institute will provide the following:**  Holmesglen Moorabbin CAIT computer classroom:  Computer with single screen per learner.  Minimum MAC or computer specification: • Quad-Core CPU, 16GB of RAM, 250GB of Storage, 4 GHz processor • Internet access  Applications access available at ZENworks and Holmesglen MyHorizon: • Brightspace (Learning Management System)  • 365 Microsoft office suite  • WebEx • Google Chrome – recommended web browser • 7Zip • Visual Studio Code • FileZilla / cPanel services • WAMP • MAMP • Web server account  • Postman  • GitHub Desktop  • Apache  • PHP  • MySQL  • Wordpress |

|  |  |  |  |
| --- | --- | --- | --- |
| Section E – Marking Sheet - Student Answer Sheet | | | |
| **Subject code:**  **Unit code:** | (CWEB4)   ICTWEB445  ICTWEB430  ICTWEB434 | **Subject title:**  **Unit title:** | (Dynamic Web Development)  Implement content management systems  Produce server-side script for dynamic web pages  Transfer content to websites |

| **Portfolio task** | | | | |
| --- | --- | --- | --- | --- |
| **Criteria for assessment** | | **Satisfactory** | | **Comment** |
| **Yes** | **No** |
| **Files:**  The following files must be submitted for this assessment: | | | | |
| 1. | Assessment Sheet (*signed*) | ☐ | ☐ |  |
| 2. | CMS Portfolio Project files (*zip file*) | ☐ | ☐ |  |
| 3. | WordPress Mini-Project files (*zip file*) | ☐ | ☐ |  |
| **Marking criteria:** Part 1 - Confirm Application Requirements | | | | |
| 1. | Learner has reviewed and assessed the user requirements in the brief & has confirmed the purpose of the web application, the required functionality, and the relevant standards & technologies, needed for the development of the project. | ☐ | ☐ |  |
| 2a. | Learner has confirmed the primary scripting language used by the server including at least ONE advantage and disadvantage | ☐ | ☐ |  |
| 2b. | Learner has listed two other scripting languages that could be used including at least ONE advantage and disadvantage for each | ☐ | ☐ |  |
| 3. | Learner has confirmed the core pages required by the user for the website & has included at least ONE wireframe prototype | ☐ | ☐ |  |
| 4. | Learner has confirmed the pages that will be dynamic and interact with the database | ☐ | ☐ |  |
| 5. | Learner has confirmed all relevant technologies required for the application | ☐ | ☐ |  |
| 6. | Learner has confirmed the submission includes all required documents | ☐ | ☐ |  |
| 7. | Learner has listed at least THREE Content Management Systems (“CMS”), provided ONE sentence describing each and confirm the tools used to build the Portfolio’s Content Management System | ☐ | ☐ |  |
| 8. | The learner has:   * identified, analysed and evaluated the user requirements, workplace instructions and technical documentation, to determine all organisational requirements. * considered and understood all necessary licencing policies, procedures, regulatory requirements and organisational security/privacy policies & guidelines. * determined appropriate language & technologies based on required functionality; and * obtained approval from the client about the pages required and the necessary dynamic functionality | ☐ | ☐ |  |
| **Marking criteria:** Part 2 - Design the Database | | | | |
| 1. | A SQL to store the dynamic website’s data has been created | ☐ | ☐ |  |
| 2. | All required tables to store the dynamic website’s data have been created | ☐ | ☐ |  |
| 3. | All required fields have been added to the tables | ☐ | ☐ |  |
| 4. | The learner has created an application directory with an entry point file that allows for scripts to interact with the database | ☐ | ☐ |  |
| 5. | The register functionality has been created and the password encrypted | ☐ | ☐ |  |
| 6. | The login and logout functionality has been created and the user can logout of the session | ☐ | ☐ |  |
| 7. | Post functionality have been written to allow for read, post, update and delete CRUD requests via a GUI | ☐ | ☐ |  |
| 8. | The create, edit and delete pages have been secured | ☐ | ☐ |  |
| 9. | The learner has ensured that the website meets accessibility and XHTML standards | ☐ | ☐ |  |
| **Marking criteria:** Part 3A - Develop Dynamic Server & Custom CMS | | | | |
| 1. | Confirm at least ONE web page is dynamic in accordance with user requirements and provide a screenshot | ☐ | ☐ |  |
| 2. | Confirm and list the pages which allow for user customisation, pages which allow for the user to write to the server & database to change what is displayed on the CMS | ☐ | ☐ |  |
| 3. | Confirm that a custom stylesheet has been implemented and confirm the responsive CSS framework used. Provide a screenshot example of the styling used by the CMS | ☐ | ☐ |  |
| 4. | Confirm the Custom CMS files have been backed up in a separate location to the root directory | ☐ | ☐ |  |
| 5. | Confirm the SQL database data has been exported and backed up in a separate location to the root directory | ☐ | ☐ |  |
| **Marking criteria:** Part 3B - Develop Commercial Mini-CMS Exemplar | | | | |
| 6. | **Configure a separate Commercial CMS solution using WordPress:**  (i) Confirm the CMS plugins installed & configured within this mini project (**at least one integrated**)  (ii) Confirm functionality provided by the WordPress Basic CMS  (iii) Confirm the WordPress Mini-CMS files have been backed up in a separate location to the root directory | ☐ | ☐ |  |
| **Marking criteria:** Part 4 - Incorporate Additional Functionality | | | | |
| 1. | List and confirm the security features incorporated into the custom CMS project to ensure the server is protected and minimise potential database attacks | ☐ | ☐ |  |
| 2. | A script that can upload and retrieve images has been created and works as required | ☐ | ☐ |  |
| 3. | The learner has ensured user friendly error messages are displayed to the user and errors gracefully handled | ☐ | ☐ |  |
| **Marking criteria:** Part 5 - Deployment (*Optional*) | | | | |
|  | Deploy the application to a web server | ☐ | ☐ |  |
| 2. | Implement HTTPS on the web server | ☐ | ☐ |  |
| 3. | Provide login details to allow the assessor to check functionality | ☐ | ☐ |  |
| **Marking criteria:** Part 6 - Test and Debug the Code | | | | |
| 1. | Learner has validated the HTML and CSS using W3C Validation Service & HTML Checker | ☐ | ☐ |  |
| 2. | The learner has tested that the CMS website functions as expected in at least THREE different browsers (Chrome, Firefox and Edge) | ☐ | ☐ |  |
| 3. | Learner has listed at least TWO debugging tools and provided ONE advantage and ONE disadvantage of the use of each tool | ☐ | ☐ |  |
| 4. | Learner has tested that the website meets the core functionality required by the user in the brief and recorded the results |  |  |  |
| 4a. | **Test:** That at least ONE CRUD GET request renders the expected output at a specified route | ☐ | ☐ |  |
| 4b. | **Test:** That at least ONE CRUD POST request can be created, saves data correctly to database and can be displayed to user | ☐ | ☐ |  |
| 4c. | **Test:** That at least ONE CRUD PUT request can occur, whereby posts can be edited and displayed to user | ☐ | ☐ |  |
| 4d. | **Test:** That the user can successfully login | ☐ | ☐ |  |
| 4e. | **Test:** That the outcome for an unsuccessfully login displays an error message | ☐ | ☐ |  |
| 4f. | **Test:** That new users are registered, and password is encrypted in the database | ☐ | ☐ |  |
| 4g. | **Test:** That an image can be uploaded, stored on the server and retrieve & displayed to the user | ☐ | ☐ |  |
| **Marking criteria:** Part 7 - Document Activities & Hand Over | | | | |
| 1. | The learner’s project meets all requirements outlined in the brief to attain sign off | ☐ | ☐ |  |
| 2. | The learner’s project meets all requirements outlined in the brief to attain sign off | ☐ | ☐ |  |
| 3. | The learner has used the language debugging facilities of an integrated development environment (IDE) to examine and output results and variables. | ☐ | ☐ |  |
| 4. | The learner has discussed their project including both the written and project working files. | ☐ | ☐ |  |
| 5. | The learner used a range of strategies to establish a sense of connection and to build rapport with the client and peers. | ☐ | ☐ |  |
| 6. | The learner has used strategies to establish a sense of connection and build a relationship with the client | ☐ | ☐ |  |

**Assessment Submission Cover Sheet (VET)**

|  |  |
| --- | --- |
| **Student Declaration – Must be signed before submission** |  |
| By submitting this assessment task and signing the below, I acknowledge and agree that:  • This completed assessment task is my own work.  • I understand the serious nature of plagiarism and I am aware of the penalties that exist for breaching this.  • I have kept a copy of this assessment task.  • The assessor may provide a copy of this assessment task to another member of the Institute for validation and/or benchmarking purposes. | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Student ID:** | **100683785** | **Student name:** | **Vorakorn Taweetawon** |

|  |  |
| --- | --- |
| **Submission or observation date:** | **12.06.2024** |
| **Student signature**  For electronic submissions: By typing your name in the student signature field, you are accepting the above declaration. | **Vorakorn Taweetawon** |

| Section F – Feedback to Student | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Has the student successfully completed this assessment task?** | | | | | **Yes** | **No** |
| **☐** | **☐** |
| **Additional Assessor comments (as appropriate):** | | | | | | |
|  | | | | | | |
| **Resubmission allowed:** | **Yes ☐** | **No ☐** | **Resubmission due date:** |  | | |
| **Assessor name:** | Alex Bicknell | | | | | |
| **Assessor signature:** |  | | | | | |
| **Assessed date:** |  | | | | | |

**Supporting document**

# Dynamic Web Development – Portfolio (Brief & Documentation)

|  |  |  |  |
| --- | --- | --- | --- |
| **Subject code:**  **Unit code**: | (CWEB4)   ICTWEB445  ICTWEB430  ICTWEB434 | **Subject title:**  **Unit title:** | (Dynamic Web Development)  Implement content management systems  Produce server-side script for dynamic web pages  Transfer content to websites |

|  |
| --- |
| **Project Brief** Read through the brief below take note of the needs and requirements outlined in the brief. |
| **Overview of project**  Learners are required to develop a solution for a client using PHP, MySQL & implement a content management system. The learner will produce server-side scripts using PHP to integrate with the MySQL database. Further, learners will be required to learn the steps to transfer content to websites using FTP involving FileZilla and/or cPanel.  Learners will also be required to develop a separate Mini-CMS exemplar solution using WordPress and at least one plugin, to demonstrate a commercial alternative to the Custom CMS.  **This assessment task is divided into the following parts:** Part 1 – Confirm Application Requirements  Part 2 – Design the Database Part 3A & B – Develop Dynamic Server & Custom CMS & Commercial Mini-CMS Exemplar Part 4 – Incorporate Additional Functionality Part 5 – Deployment (*Optional*) Part 6 – Test and Debug the Code Part 7 – Document Activities & Hand Over  **Client Scenario**  You are a freelancer website developer and have been recently engaged by Jumblr Ltd. to build a stylised business blog for their personal review business. **You may choose any name & topic for review of your liking such as food reviews, car reviews, website reviews, and so on.** Jumblr plans to launch the new venture late the year in time for the Christmas rush.  Broadly, Jumblr would like the application to be built as a customised WAMP/MAMP stack, where the Apache server is to be run with PHP, interact with a MySQL database and resemble a content management system (CMS), mirroring commercial alternatives like WordPress. This business blog must be secured by a login system.  **Additionally, the client would like you to develop a separate Mini-CMS application using WordPress.** Jumblr would like to see the basic functionality of a commercial CMS and the benefits of integrating at least one open-source plugin to enhance the functionality of the CMS.  Your assessor will be playing the role of the client.  They have summarised the specific features and requirements as follows (***see next page***):  **Required “Client-side” UI/UX Functionality**  The complete project must include at least the following features:   * As the website is a prototype, **a minimum of at least SIX different pages**, including Home, Posts, Create Post, Edit Post, Signup, Login/Logout. * Format the data so it is easily understood by the user * Completed login system providing users the ability to register, login & logout * Completed content management system, providing users the ability to create, view, edit and delete posts * Completed page/component that allows a user to upload & retrieve images. * Displaying of data is to use appropriate controls, such as text boxes, dropdowns, radio buttons * User-friendly error pages must be designed and created   **Required Server-side Functionality**  Your complete project must include the following technical requirements:   * Utilisation of a MAMP / WAMP / XAMP stack, with Apache, PHP & MySQL mandatory technologies * A **minimum of TWO (2)** database tables must be created * The user’s data & posts data must be stored in a database (*password must be encrypted*) * Only logged-in users are authorised to create, edit & delete posts * All users must be permitted & authorised to view the posts page * Sessions must be used to save the state of the application & be destroyed on users logging out * Create PHP script that will upload and retrieve images. * The images should be saved in a folder on the same server as the PHP script.   Your additional mini-CMS project must include the following technical requirements:   * WordPress.org CMS with a home page built * Integration (installation & configuration) of at least one open-source plugin * Confirmation of basic functionality of the WordPress CMS * Confirmation of the WordPress Mini-CMS backup   **NOTE*:*** *You may choose to deploy either website to a web host server:*   * You will need a domain name and server space if you wish to work directly on a server * If the application is uploaded to a webserver, it must also be secured with HTTPS   **Security Requirements**   * Appropriate and user-friendly error messages must be returned, without error stacks * User passwords must be encrypted before saving them to the database * User data must be validated before saving to the database * Authentication and Authorization must take place before any data is saved to the database * For web-hosted websites:   + Any connections made to the server must not be on public networks   + FTPs must be used to establish a secure connection   + Each individual must be assigned unique FTP user accounts.   + No scripts may be placed on the live public server until they have been tested and approved   **Required Language Standards & Technologies**   * HTML5 & CSS3 languages to be used for UI/UX client-side development * PHP as the standard language for server-side scripting * SQL as the standard language for database creation & manipulation, via MySQL client * Apache HTTP server as the main web server software * Use of a responsive component library like Bootstrap * XHTML standards must also be adhered to * Application, where possible, to build accessible web content in accordance with web accessibility guidelines and standards (e.g. WCAG, WAI-ARIA, UAAG, etc.) * Fonts must be suitable for web design: web safe, local and/or google fonts * Sourced assets, icons & images must be used in accordance with attributed copyright |
| **Final Instructions & Guidance** |
| Complete and document the steps/questions below. Your database, table name, field names and file names may vary from the instructions below.  On completion of Part 1, we recommend the following order when building the project using a SQL database, PHP server-side scripting language and CMS system:  **Suggested Development Order**  Main Custom CMS   * Develop a HTML and CSS layout for the website * Develop a Database for the blog * Develop the tables and fields for the login system * Create the code to register a user * Ensure the users password in encrypted in the database * Create a page that allows a user to login * Display appropriate error messages if the login details are incorrect * Redirect the user once they have logged in * Create code to destroy a session and log the user out * Create code to create posts * Create code to display posts * Create code to edit posts * Create code to delete posts * Secure the create, edit, and delete posts so that the user must be logged in to user these features * Create PHP script that will upload and retrieve images. This may be simply added as an additional page or you may choose to incorporate it into other pages such as create posts and display posts. The images should be saved in a folder on the same server as the PHP script. * Ensure user friendly error messages are displayed to the user and errors are gracefully handled * Ensure the site meets web accessibility requirements * Export and backup the CMS application files   Additional Mini-CMS   * Setup the mini-CMS project using WordPress * Create basic home page and functionality * Select, install, and configure at least one plugin * Export and backup the WordPress application files |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part 1 - Confirm Application Requirements** Based on the information you have gathered from the brief complete the following: | | | | | | |
| **Question / Criteria** | | | **Answer / Satisfactory Response** | | | |
| 1. Review and assess the user requirements *in the brief* above and confirm:  * the **purpose** of the web application, * the required **functionality**; and * the relevant **standards & technologies**, needed for the development of the project. | | | **Purpose of web application** | | | |
| * The web application is a styled business blog for personal review business. * Users can read reviews and potentially write their own. | | | |
| **List of Required Functionality** | | | |
| **Required “Client-side” UI/UX Functionality**   * As the website is a prototype, **a minimum of at least SIX different pages**, including Home, Posts, Create Post, Edit Post, Signup, Login/Logout. * Format the data so it is easily understood by the user * Completed login system providing users the ability to register, login & logout * Completed content management system, providing users the ability to create, view, edit and delete posts * Completed page/component that allows a user to upload & retrieve images. * Displaying of data is to use appropriate controls, such as text boxes, dropdowns, radio buttons * User-friendly error pages must be designed and created   **Required Server-side Functionality**   * Utilisation of a MAMP / WAMP / XAMP stack, with Apache, PHP & MySQL mandatory technologies * A **minimum of TWO (2)** database tables must be created * The user’s data & posts data must be stored in a database (*password must be encrypted*) * Only logged-in users are authorised to create, edit & delete posts * All users must be permitted & authorised to view the posts page * Sessions must be used to save the state of the application & be destroyed on users logging out * Create PHP script that will upload and retrieve images. * The images should be saved in a folder on the same server as the PHP script. | | | |
| **Relevant Standards & Technologies** | | | |
| * HTML5 & * PHP as the standard language * SQL as the standard language for database creation & manipulation, via MySQL client * Apache HTTP server as the main web server software   a responsive component library like Bootstrap   * XHTML standards must also be adhered to * Application, where possible, to build accessible web content in accordance with web accessibility guidelines and standards (e.g. WCAG, WAI-ARIA, UAAG, etc.) * google fonts * Sourced asset : unsplash for images | | | |
| 1. (a) Confirm the server-side scripting language required for the website and detail **at least ONE advantage and ONE disadvantage** of this language | | | The server-side scripting language required for the website is PHP.  Advantages of PHP: PHP has strong support for database integration, making it easy to connect to popular databases like MySQL. This makes it an excellent choice for building applications that rely heavily on data storage and retrieval.  Disadvantages of PHP: PHP does not have a built-in debugger, which makes it harder to find and fix bugs and errors in code. | | | |
| 1. (b) List TWO other scripting languages that could be used and ONE advantage and ONE disadvantage for each language | | | Python and Ruby  Advantages of Python: Python is known for its clean and concise syntax, often resembling natural language. This makes code easier to read, understand, and maintain.  Disadvantages of Python: Python can be slower than compiled language for computationally intensive applications.  Advantages of Ruby: offers a rapid development environment with pre-built components and conventions. This can significantly speed up development for projects that benefit from a structured framework.  Disadvantages of Ruby: Compared to PHP, Ruby has a smaller developer community. This might translate to fewer resources and libraries readily available. | | | |
| 1. List all pages/views required for the website (including navigation) and display in at least ONE wireframe prototype | | | As the website is a prototype, **a minimum of at least SIX different pages**, including   * Home * Posts * Create Post * Edit Post * Signup * Login/Logout.   *../../../Screenshot%202024-06-10%20at%207.44.43%20PM.png* | | | |
| 1. Confirm the pages that will be dynamic and interact with the database as part of the website, based on user requirements | | | * Posts * Create Post * Edit Post * Signup * Login/Logout. | | | |
| 1. Confirm all relevant technologies required to create dynamic pages, as per a typical “LAMP Stack” application | | | * Linux * Apache * PHP * MySQL | | | |
| 1. Confirm the submission includes all required documents | | |  | | | |
| 1. List at least **THREE** commercial Content Management Systems (“CMS”), provide ONE sentence describing each and confirm the tools used to build the Portfolio’s Content Management System | | | WordPress: A popular open-source Content Management System (CMS). Offers a wide range of themes and plugins to customize the website's appearance and functionality  Shopify: e-commerce platform designed for creating online stores. Offers features for product management, inventory control, secure payment processing, and marketing tools.  Wix: A website builder known for its user-friendly drag-and-drop interface and ease of use. Ideal for creating basic websites without extensive coding knowledge.  The project brief specifies building a custom CMS using a LAMP Stack.   * Linux: Operating System * Apache HTTP Server: Web Server * MySQL: Database Management System * PHP: Server-Side Scripting Language | | | |
| **8. Approval and Feedback** After you have finished the planning, obtain approval from the assessor to confirm the user requirements | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | **Yes** | **No** |
| The learner has identified, analysed, and evaluated the user requirements, workplace instructions and technical documentation, to determine all organisational requirements | | | | | ☐ | ☐ |
| The learner has considered and understood all necessary licencing policies, procedures, regulatory requirements, and organisational security/privacy policies & guidelines | | | | | ☐ | ☐ |
| The learner has determined appropriate language & technologies based on required functionality | | | | | ☐ | ☐ |
| The learner has obtained approval from the client about the pages required and the necessary dynamic functionality | | | | | ☐ | ☐ |
| **Assessor Name** | *Alex Bicknell* | **Assessor Signature** | *Signature* | **Date** | *##.##.2024* | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part 2 - Design the Database** Based on the information you have gathered from the brief complete the following: | | | | | | |
| **Question / Criteria** | | | **Complete / Not Complete** | **Evidence of Completion** | | |
| 1. The SQL database to store the dynamic website’s data has been created | | |  |  | | |
| 1. All required tables to store the dynamic website’s data have been created | | |  |  | | |
| 1. All required fields have been added to the tables | | |  |  | | |
| 1. The application directory and the entry point file that interacts with the database, has been created | | |  | index.php, connect.inc.php | | |
| 1. The register functionality has been created and the password encrypted | | |  | signup.php, signup.inc.php | | |
| 1. The login and logout functionality has been created and the user can logout of the session | | |  | login.inc.php, logout.inc.php | | |
| 1. Post functionality have been written to allow for Read, Post, Update and Delete CRUD requests via a GUI | | |  | posts.php, createpost.php, editpost.php | | |
| 1. The Create, Edit and Delete pages have been secured | | |  | createpost.inc.php, deletepost.inc.php, editpost.inc.php | | |
| 1. The learner has ensured that the website meets accessibility requirements and XHTML standards | | |  |  | | |
| **Approval and Feedback** After you have finished planning the database design, obtain approval from the assessor | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | **Yes** | **No** |
| The learner has designed the database according to user and website requirements | | | | | ☐ | ☐ |
| **Assessor Name** | *Alex Bicknell* | **Assessor Signature** | *Signature* | **Date** | *##.##.2024* | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part 3A - Develop Dynamic Server & Custom CMS** Based on the information you have gathered from the brief complete the following: | | | | | | |
| **Question / Criteria** | | | **Complete / Not Complete** | **Evidence of Completion** | | |
| 1. Confirm at least ONE web page is dynamic in accordance with user requirements and list the page file location | | |  | posts.php | | |
| 1. Confirm and list the pages which allow for user customisation, being the pages, which allow for the user to write to the server & database to change what is displayed on the CMS | | |  | * createpost.php * editpost.php | | |
| 1. Confirm that a custom stylesheet has been implemented and confirm the responsive CSS framework used. Provide a screenshot example of the styling used by the CMS | | |  | styles.css  bootstrap  *< Screenshot of CMS Theme >../../../Screenshot%202024-06-10%20at%207.34.03%20PM.png* | | |
| 1. Confirm the Custom CMS files have been backed up in a separate location to the root directory | | |  | *< Screenshot of Backup >*  *../../../Screenshot%202024-06-10%20at%208.54.50%20PM.png* | | |
| 1. Confirm the SQL database data has been exported and backed up in a separate location to the root directory | | |  | *< Screenshot of Backup >*  ../../../Screenshot%202024-06-10%20at%208.45.58%20PM.png | | |
| **Approval and Feedback** After you have finished planning the database design, obtain approval from the assessor | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | **Yes** | **No** |
| The learner has designed and built the main Custom CMS according to user requirements | | | | | ☐ | ☐ |
| **Assessor Name** | *Alex Bicknell* | **Assessor Signature** | *Signature* | **Date** | *##.##.2024* | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Part 3B - Develop Commercial Mini-CMS Exemplar** Based on the information you have gathered from the brief complete the following: | | | | | | |
| **Question / Criteria** | | | **Complete / Not Complete** | **Evidence of Completion** | | |
| 1. Configure a separate Commercial CMS solution using WordPress, as follows: | | |  |  | | |
| 1. Confirm the CMS plugins installed & configured within this mini project (**at least one integrated**) | | |  | *< Contact Form 7and Meta Slider, integrate All-in-One WP Migration in the project >* | | |
| 1. Confirm functionality provided by the WordPress Basic CMS | | |  | *< it is functioning as online art gallery to showcase artwork and details of the art and also show what is trendy right now under post sessions. >* | | |
| 1. Confirm the WordPress Mini-CMS files have been backed up in a separate location to the root directory | | |  | ../../../Screenshot%202024-05-23%20at%2011.06.52%20PM.png | | |
| **Approval and Feedback** After you have finished planning the database design, obtain approval from the assessor | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | **Yes** | **No** |
| The learner has designed and built the WordPress Mini-CMS according to user requirements | | | | | ☐ | ☐ |
| **Assessor Name** | *Alex Bicknell* | **Assessor Signature** | *Signature* | **Date** | *##.##.2024* | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Part 4 - Incorporate Additional Functionality** Based on the information you have gathered from the brief complete the following: | | | | | | | |
| **Question / Criteria** | | | **Complete / Not Complete** | **Evidence of Completion** | | | |
| 1. List and confirm the security features incorporated into the custom CMS project to ensure the server is protected and minimise potential database attacks | | |  | * Implement strong password hashing. * Validate user input to prevent SQL injection attacks * Use session management to identify authorized users. | | | |
| 1. A script that can upload and retrieve images has been created and works as required   (*Location of file AND ONE screenshot of image*) | | |  | upload.php  *< Screenshot of dynamically retrieved image >*  *../../../Screenshot%202024-06-10%20at%209.10.51%20PM.png* | | | |
| 1. The learner has ensured user friendly error messages are displayed to the user and errors gracefully handled | | |  | header.php | | | |
| **Approval and Feedback** After you have finished planning the database design, obtain approval from the assessor | | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | | **Yes** | **No** |
| The learner has incorporated the additional functionality required for the PHP CMS | | | | | | ☐ | ☐ |
| **Assessor Name** | *Alex Bicknell* | **Assessor Signature** | *Signature* | | **Date** | *##.##.2024* | |

|  |  |  |
| --- | --- | --- |
| **Part 5 - Deployment (*Optional*)** Based on the information you have gathered from the brief complete the following: | | |
| **Question / Criteria** | **Complete / Not Complete** | **Evidence of Completion** |
| 1. Deploy the application to a web server and provide a screenshot that includes the website on a custom domain | ☐ | *<Insert a screenshot of the dynamic website running in a browser on the server>* |
| 1. Implement HTTPS on the web server and provide ONE screenshot of application running in browser with HTTPS enabled | ☐ | *<Insert a screenshot of the dynamic website running in a browser on the server with HTTPs enabled>* |
| 1. Provide login details to allow the assessor to check functionality   (Include *URL, username, password*) | ☐ | *< List login details >*  *Example:*  *URL: www.alex.com Dynamic website Username: admin Dynamic website Password: password* |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Part 6 - Test and Debug the Code** Based on the information you have gathered from the brief complete the following: | | | | | | | | | | |
| **Question / Criteria** | | | | **Answer / Satisfactory Response** | | | | | | |
| 1. Validate the HTML and CSS using W3C Validation Service & HTML Checker   (*Provide TWO screenshots in total*) | | | | ../../../Screenshot%202024-06-10%20at%209.21.46%20PM.png../../../Screenshot%202024-06-10%20at%2011.45.13%20PM.png | | | | | | |
| 1. The CMS website functions as expected in at least THREE different browsers (Chrome, Firefox and Edge). List any errors encountered and how they were fixed.   (*Provide at least ONE screenshot of a page functioning in each Browser – THREE in total*) | | | | *< Screenshot ONE of page in Google Chrome >*  *../../../Screenshot%202024-06-10%20at%209.23.47%20PM.png*  *< Screenshot TWO of page in Firefox >*  *../../../Screenshot%202024-06-10%20at%209.24.58%20PM.png*  *< Screenshot THREE of page in Edge >*  *../../../Screenshot%202024-06-10%20at%209.28.12%20PM.png*  *< List of any errors >*  errors not found | | | | | | |
| 1. List at least TWO debugging tools and provide ONE advantage and ONE disadvantage of the use of each tool | | | | PHP Error Logging  Advantage: Enabling PHP error logging can help keep track of issues such as file upload errors, deletion failures, and any other runtime errors that may occur during the interaction with the file system or user inputs.  Disadvantage: If the project has a lot of errors or warnings, the log files can become large and unwieldy, making it difficult to manage and identify the most critical issues.  Xdebug  Advantage: When implementing the file upload and deletion features, Xdebug can help track down errors related to file handling, such as issues with file paths, permissions, or handling of superglobals ($\_FILES).  Disadvantage: The project may experience slower performance during development, especially when handling multiple image uploads. | | | | | | |
| 1. Test the website meets the core functionality required by the user in the brief and record the results. Below you will **complete the following SEVEN tests**, each providing the (i) test conducted, (ii) expected output of the test, (iii) success / failure and (iv) screenshot of the test result. | | | | | | | | | | |
| **Test Design** | | **Expected Output** | | **Correct Function** | | | **Screenshot of Completed Test** | | | |
| **Yes** | | **No** |
| 1. **Test:** That at least ONE CRUD GET request renders the expected output at a specified route | | *The error message "Please fill in all fields" should be displayed.*  *After opening the login page and submit the login form with both email and password fields empty.* | |  | | ☐ | < *Screenshot of GET page* >../../../Screenshot%202024-06-10%20at%2010.56.42%20PM.png | | | |
| 1. **Test:** That at least ONE CRUD POST request can be created, saves data correctly to database and can be displayed to user   (*TWO screenshots*) | | *The new post with the title "Starry Night", the provided image URL, description, artist name, and year should be displayed on the posts.php page.* | |  | | ☐ | < *Screenshot of create post page* >  ../../../Screenshot%202024-06-10%20at%2011.04.00%20PM.png  < *Screenshot of displayed posts* >  ../../../Screenshot%202024-06-10%20at%2011.05.49%20PM.png  ../../../Screenshot%202024-06-10%20at%2011.05.58%20PM.png | | | |
| 1. **Test:** That at least ONE CRUD PUT request can occur, whereby posts can be edited and displayed to user   (*TWO screenshots*) | | *Edit The post with the title "Starry Night" to be “Art Deco” displayed on the posts.php page.* | |  | | ☐ | < *Screenshot of edit post page* >  ../../../Screenshot%202024-06-10%20at%2011.10.17%20PM.png  < *Screenshot of displayed edited posts* >  ../../../Screenshot%202024-06-10%20at%2011.10.28%20PM.png  ../../../Screenshot%202024-06-10%20at%2011.10.36%20PM.png | | | |
| 1. **Test:** That the user can successfully login | | *The user is redirected to the homepage or dashboard. And a success message is displayed, “welcoming the user to their creative space” and The "Create Post" and ” Upload your artwork “ button are visible.* | |  | | ☐ | < *Screenshot of logged in user* >  ../../../Screenshot%202024-06-10%20at%2011.21.54%20PM.png | | | |
| 1. **Test:** That the outcome for an unsuccessfully login displays an error message | | *The system should display an error message indicating that the user does not exist.* | |  | | ☐ | < *Screenshot of unsuccessful login* >  ../../../Screenshot%202024-06-10%20at%2011.53.31%20PM.png | | | |
| 1. **Test:** That new users are registered and password is encrypted in the database | | *When registered success will display “You have successfully signed up!”*  *The system should register the new user successfully. Username “Rihana22Riri”*  *The password should be encrypted using a secure hashing algorithm.* | |  | | ☐ | < *Screenshot of database output for register user password* >  ../../../Screenshot%202024-06-11%20at%2012.02.33%20AM.png | | | |
| 1. **Test:** That an image can be uploaded, stored on the server and retrieve & displayed to the user   (*TWO screenshots*) | | *Image uploaded is stored on the server correctly and can be retrieved and displayed* | |  | | ☐ | < *Screenshot of image uploaded successfully* >  ../../../Screenshot%202024-06-11%20at%2012.04.49%20AM.png  < *Screenshot of retrieved image* >../../../Screenshot%202024-06-11%20at%2012.04.59%20AM.png | | | |
| **Approval and Feedback** After you have finished planning the database design, obtain approval from the assessor | | | | | | | | | | |
| **Checklist (To be completed by the learner’s facilitator)** | | | | | | | | | **Yes** | **No** |
| The learner has validated and tested the PHP CMS application as required | | | | | | | | | ☐ | ☐ |
| **Assessor Name** | *Alex Bicknell* | | **Assessor Signature** | | *Signature* | | | **Date** | *##.##.2024* | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Part 7 - Document Activities & Hand Over** Based on the information you have gathered from the brief complete the following: | | | | | | | | |
| **SUBMISSION CHECKLIST – COMPLETE PRIOR TO SIGN OFF** | | | | | | | | |
| **Complete Final Sign Off Checklist**  Prior to final sign off, please go through and tick off the following checklist **JUST PRIOR** to submission, to ensure you have:   * completed the Assessment Sheet; and * completed all required rubric and portfolio pieces. | | | | | | | | |
|  | **Requirements for Submission** | | | | | | | **Confirm** |
| **I.** | **INDIVIDUAL SUBMISSION:**  You are signing off & submitting your Assessment & Portfolio pieces (x2) individually | | | | | | | ☐ |
| **II.** | **COMPLETE ASSESSMENT SHEET:**  You have completed Parts 1 to 6 of this Assessment Sheet | | | | | | | ☐ |
| **III.** | **SIGN OFF:**  You have booked a final sign-off meeting with the Assessor prior to submission of the Assessment, to discuss the project & this checklist. | | | | | | | ☐ |
| **IV.** | **CORRECTLY EXPORTED PORTFOLIO PIECES:**  You have completed the Portfolio pieces (**x2**), and have provided all required evidence of completion by following the steps below: | | | | | | |  |
| **PHP CMS Project:**   * Go to phpMyAdmin, to your project’s database & select the “Export” tab. * Use the default “Quick” export & hit “Go”. Store the .sql file inside your PHP CMS Project folder. * Select your PHP CMS Project folder and zip the folder up. It should contain your entire CMS, the database backup and the Uploader folder (*if not integrated directly into your CMS*). | | | | | | | ☐ |
| **WordPress Mini-CMS Project [Option 2]:**   * Download & install the plugin “All-in-One WP Migration” * Go to the new “All-in-One menu” section in the Wordpress dashboard & click “export” * Clicking the green dropdown list, choose the first option “File” * This will automatically download a .wpress file. Save this into your WordPressCMS folder & zip up the folder | | | | | | | ☐ |
| **V.** | **DOCUMENTS LIST:** You have prepared the following documents to be signed off: | | | | | | |  |
| * Assessment 2 Sheet (*completed, dated & prior sections signed off*) | | | | | | | ☐ |
| * Main Custom PHP CMS Portfolio folder | | | | | | | ☐ |
| * WordPress Mini-CMS Portfolio folder | | | | | | | ☐ |
| **Ready for Submission**  You are now in a position to submit all working files along with this document having completed all necessary documentation, sign offs & checklists. | | | | | | | | |
| **Skills to be observed during this task to the required standard.** Checklist (To be completed by the learner’s facilitator)The following tasks are to be completed in relation to the brief for this project. Each of the skills must be observed on at least one occasion. | | | | | **Date 1** | | **Date 2** | |
| ##.##.2024 | |  | |
| **Satisfactory** | | **Satisfactory** | |
| **Yes** | **No** | **Yes** | **No** |
| 1. The learner’s project meets all requirements outlined in the brief to attain sign off | | | | | ☐ | ☐ | ☐ | ☐ |
| 1. The learner’s project meets all requirements outlined in the brief to attain sign off | | | | | ☐ | ☐ | ☐ | ☐ |
| 1. The learner has used the language debugging facilities of an integrated development environment (IDE) to examine and output results and variables. | | | | | ☐ | ☐ | ☐ | ☐ |
| 1. The learner has discussed their project including both the written and project working files. | | | | | ☐ | ☐ | ☐ | ☐ |
| 1. The learner used a range of strategies to establish a sense of connection and to build rapport with the client and peers | | | | | ☐ | ☐ | ☐ | ☐ |
| 1. The learner has used strategies to establish a sense of connection and build a relationship with the client | | | | | ☐ | ☐ | ☐ | ☐ |
| **Assessor Name** | | *Alex Bicknell* | **Assessor Signature** | *Signature* | **Date** | | *##.##.2024* | |