**AD006P24 - Associate Degree in Information Technology  
COSC2446 Web Programming A**

|  |  |  |  |
| --- | --- | --- | --- |
| Assessment Number: 1 Static Website & Demonstration Assignment | | | |
| **Weighting** | 30% | **Due Date** | 22nd of August 2025, COB (5:30 pm) |
| **Course coordinator** | Tanya Unterberger | **Contact details** | [tanya.unterberger@rmit.edu.au](mailto:tanya.unterberger@rmit.edu.au) (for general enquires only) |
| **Individual or Group Assessment** | Individual | | |

|  |
| --- |
| This assessment task is expected to take you a minimum of 30 hours to complete. |
| **Task** |
| This assessment includes a practical submission and a scheduled one-on-one demonstration.  Students submit a responsive, standards-compliant front-end website with interactivity using JavaScript. Assessed for design consistency, responsiveness, use of Bootstrap, JS interactivity, and code quality.  In the demonstration, you will walk through your submitted site, explain how key parts of your code work, and answer a few short questions related to the technologies used. You are allowed to use AI tools in your work, but you are expected to understand and be able to explain any code submitted.  **Understanding Your Work** During your scheduled demonstration, you will be required to explain how key parts of your submitted code work. This is a vital component of the assessment. If you are unable to demonstrate a clear understanding of your own code — regardless of whether it was written by you or with the assistance of AI tools — you will not be eligible to receive more than a **Pass (50%)** for the task. This policy supports academic integrity and ensures that all students meet the learning outcomes of the course. |
| **What is being assessed** |
| In this assessment, the following course outcomes are being assessed:  1. Define the protocols and systems used on the Web  2. Explain the functions of clients and servers on the Web.  3. Create standards-compliant web pages. |
| The following program outcomes are incorporated into this assessment task:  1. Enabling Knowledge  2. Critical Analysis  3. Problem Solving  4. Communication  5.Responsibility |
| **Assessment details** |
| This assignment aims to create SkillSwap, a peer-to-peer skill-sharing platform using HTML, CSS, JavaScript, and Bootstrap 5. The application will be a static website for stage one.  The submission must demonstrate secure client-side validation and best practices in code organisation, client-side validation, reusable components, and adherence to best practices in code structure and deployment. The project must be completed and submitted individually.  This assessment includes a practical submission and a scheduled one-on-one demonstration.  **AI TOOLS AND THEIR USE ARE NOT RESTRICTED FOR THIS ASSESSMENT TASK** |
| **Presentation requirements** |
| In the demonstration, you will walk through your submitted site, explain how key parts of your code work, and answer a few short questions related to the technologies used. You are allowed to use AI tools in your work, but you are expected to understand and be able to explain any code submitted. |
| **Assessment criteria** |
| You will be assessed on the following criteria:   |  |  | | --- | --- | | Website Structure and Design  Technical Implementation  GitHub and Deployment | Documentation and Submission  Adherence to requirements  Usability and validity | |
| Refer to the assessment rubric on Canvas for details on grading.   |  |  | | --- | --- | | **Assessment 1 Checklist** | **Completed (Y/N)** | | All pages are according to the screenshots |  | | Navbar includes logo (link to home page) and search field |  | | Carousel implemented on home page |  | | Responsive layout with *.row* and *.col-\** for every page |  | | Gallery opens images in Bootstrap modal |  | | Form on add.html has client-side validation |  | | File input checks image extension in JS |  | | Semantic HTML tags used (*header*, *nav*, etc.) |  | | Single custom CSS file is in *assets/css*. No inline styles used |  | | Single custom JS file is in *assets/js* |  | | Google Fonts and Material Icons included |  | | Ysabeau SC font used for body |  | | Libre Baskerville font used for headings |  | | Colours used correctly (#cd4f07, #6b5b95, #fff5f8, #333333, #f6978e) |  | | .htaccess file prevents unauthenticated access to Coreteaching |  | | Git history shows consistent work |  | | Teacher added as a collaborator |  | | All HTML/CSS validates via W3C tools |  | | Able to explain every part of code in demo |  | |
| **Submission requirements** |
| You are required to submit your completed project as follows: **Deployment**  * The live site must be deployed to a **Coreteaching web server** – Jupiter/Saturn/Titan (jupiter.csit.rmit.edu.au/saturn.csit.rmit.edu.au/ titan.csit.rmit.edu.au). * Ensure all links, JavaScript, CSS, and images function correctly on the deployed site.  **Canvas Submission** Submit the following as a zipped folder named COSC2446\_a1\_<yourStudentID>.zip via **Canvas** under the “AT1: Static Website & Demonstration Assignment” assignment:   * All source files and assets. * A README.md file containing:   + The **URL of the live site**. |
| **Academic integrity** |
| It is important that you submit material which is your own work. The University provides information for students on academic Integrity through the [RMIT website](https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/academic-integrity). If you are unsure about academic matters please be sure to ask your teacher for information. |
| **Assessment Policy** |
| Assessments submitted after the due date will be accepted as follows:   * Assessments submitted between 0-24 hours after the due date: 20% penalty * Assessments submitted between 24-48 hours after the due date: 50% penalty * Assessments submitted more than 48 hours after the due date will not be accepted   **Extensions**  If you will not be able to meet the due date for an assessment you may apply to your teacher for an extension of up to seven days by completing the Application of Time to Submit Assessment Work Form at <https://www.rmit.edu.au/content/dam/rmit/documents/Students/Student_forms/Application-for-extension-of-time-to-submit-work.pdf> Applications for an extension of time must be received at least one working day before the due date for an assessment.  **Special Consideration**  If unforeseen circumstances beyond your control prevent you from submitting your work on time, you may be eligible to apply for special consideration. For further information regarding special consideration, please refer to the RMIT Special Consideration page at <https://www.rmit.edu.au/students/student-essentials/assessment-and-results/special-consideration> |

## Functional Requirements

### Code and Git Usage

* All code should reside in a “wp” GitHub repository that is under your RMIT student ID username. Initially, you should clone this repository from <https://github.com/tanyarmit/wp.git>. You should add your respective teacher as a collaborator. The code should be placed in the “a1” directory. Your Git history should reflect your work process to authenticate your contributions.
* In the a1 directory, the README.md file should contain the link to your Coreteaching deployment.
* For instructions on uploading and testing files on Coreteaching web servers, please refer to the document “Developing and deploying with Git”.
* Make sure that you follow the naming convention for files and directories. Failure to do so will result in lost marks.
* Students must demonstrate consistent use of Git throughout the project timeline. A minimum of 4 commits spread over at least 3 separate days is expected. No single day can contain more than 50% of the work.

**Penalty:**

* + If there are fewer than 4 commits, or all or 50% of the commits are made on the same day, a 50% penalty will be applied to the overall marks.
  + This is to encourage regular progress, documentation of development steps, and authentic authorship.

Page structure

* For the look and feel of the site, refer to the attached screenshots. The site should look EXACTLY like that.
* Each page should use Bootstrap and be responsive.
* All pages should use semantic HTML 5 elements - header, nav, main and footer and well-structured content according to the following instructions.
* All pages should have the same navigation. Include the logo (SkillSwap\_logo.png) in the navigation bar and link it to the home page (index.html). The navigation bar should have a search form, but the search functionality is not required. The search icon is implemented with Google Material icons.
* All pages should have the same footer with a copyright that includes your name.
* Each html page should be validated against W3C standards at <https://validator.w3.org/>. You should validate all your HTML files and the CSS file. CSS should be validated via <https://jigsaw.w3.org/css-validator/>. Correct and re-test issues. Fix any errors or warnings.

### Common assets, fonts, colours, etc.

* **Fonts**: Use Google fonts for the site. “Ysabeau SC” as a default, “Libre Baskerville” for all headings, and material icons for icons. You can have a look at the examples <https://fonts.google.com/specimen/Ysabeau+SC>, <https://fonts.google.com/icons>, <https://fonts.google.com/specimen/Libre+Baskerville>.
* **CSS**: There should be one common custom CSS file for all pages - assets/css/styles.css. You must create a CSS stylesheet that will be linked to (in the <head> element). This CSS file will contain custom theme colours.
* **JS**: There should be one common custom JS file for all pages - assets/js/scripts.js. JavaScript should be used for file extension validation and modals. You will need to write JavaScript functions in a separate JavaScript file, linked to the relevant pages. You should link the JS file in the <head> or at the end of the <body> to ensure best practices.
* **Images**: Images like the logo, banner and favicon should reside in assets/images. Skills-related images should reside in assets/images/skills. The required images are supplied on Canvas.
* **Colours**: The following colours are used for the site:

1. #cd4f07
2. #6b5b95
3. #fff5f8
4. #333333
5. #f6978e (for the gradient)

### Pages & Features

|  |  |
| --- | --- |
| **File** | **Requirements** |
| **index.html** | - Home page  - Bootstrap carousel of the 4 static images - Responsive grid of skills |
| **skills.html** | - Two-column layout: left - banner image, right - table  - Table with skills offered |
| **gallery.html** | - Grid of all skill images - Images open in a Bootstrap modal on click |
| **add.html** | - A form that would add a new skill. For Assessment 1 the form will not be submitting data. - All fields should be mandatory.  - All fields should have an appropriate <label> associated with them and a placeholder (except for level)  - Form fields:   * Title (text, required) * Category (text, required) * Description (textarea, required) * Rate per hour (number, required) * Level (select, required). 1st option is “Please select” * Image (file, required) — Only jpg, jpeg, png, gif, webp   - Validation: Image file extension checked via JavaScript |

### 4. Client-Side JavaScript

* **Image-extension validation** on for the <input type="file" name="image"> field: only jpg, jpeg, png, gif, webp.
* **Modals** for full-size images on **gallery.html** (no inline JS—use assets/js/scripts.js).

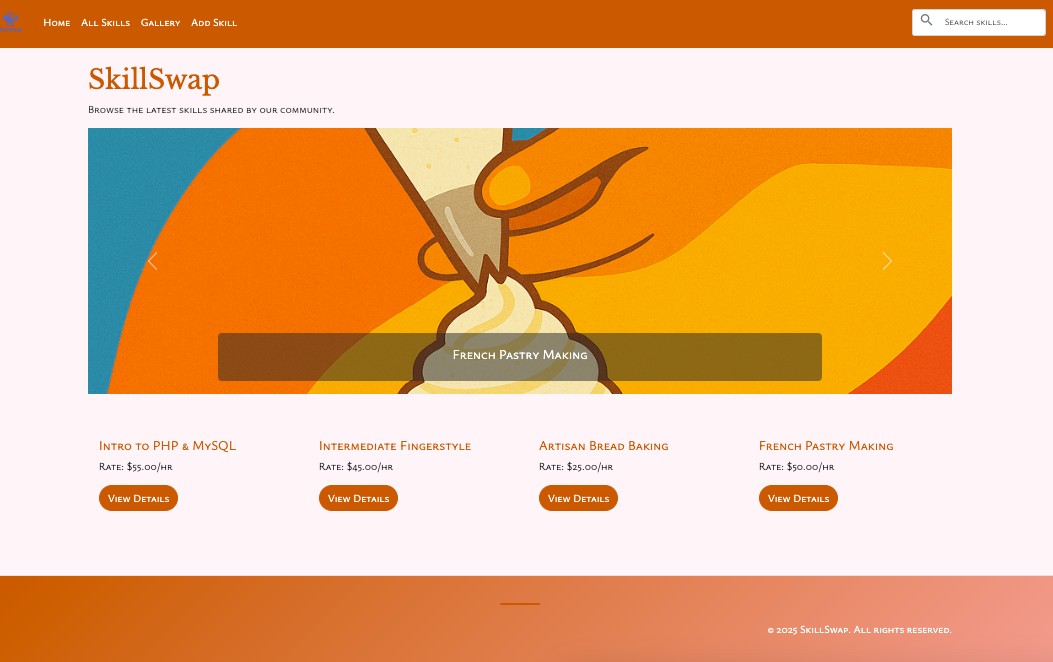
## Submission Instructions

1. Push to your private Git repository with clear commits.
2. Deploy to the Coreteaching web servers and ensure all assets load correctly.
3. Submit via Canvas a zipped folder with:

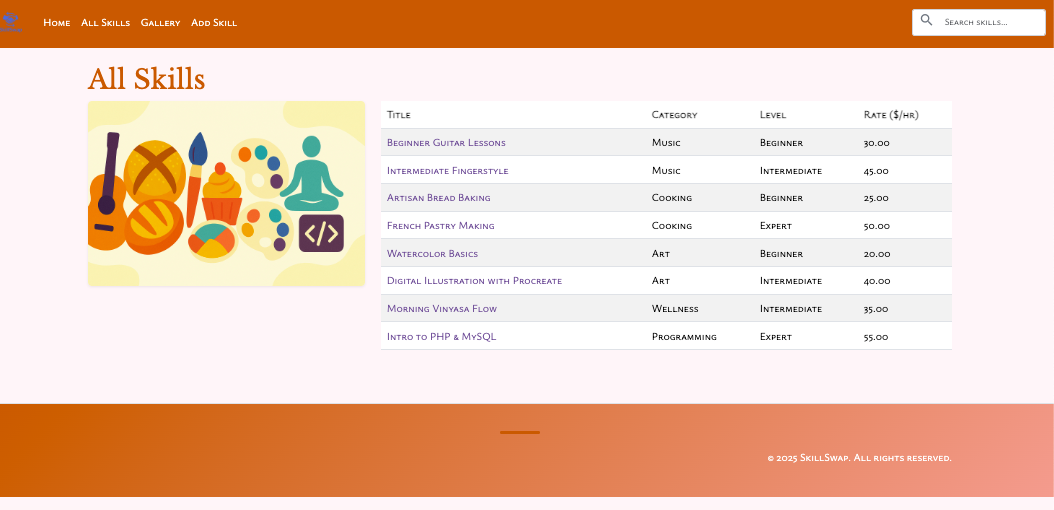
- All HTML, CSS, JS, and image files structured as required  
- README.md with the live site URL

NOTE: The following screenshots are provided as reference for the required look and feel. Students are expected to replicate these **exactly** in their implementations.

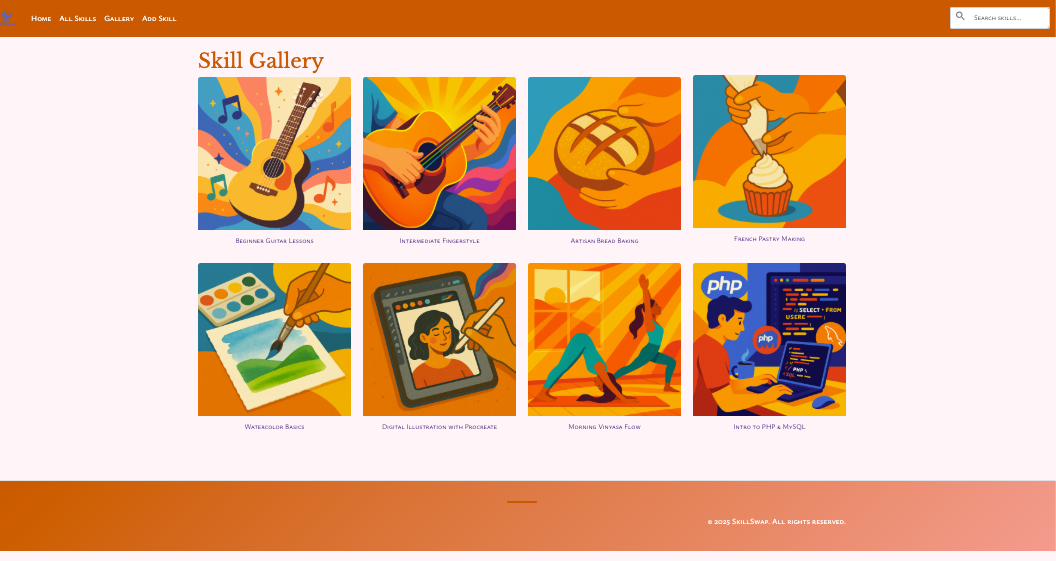
Home page:



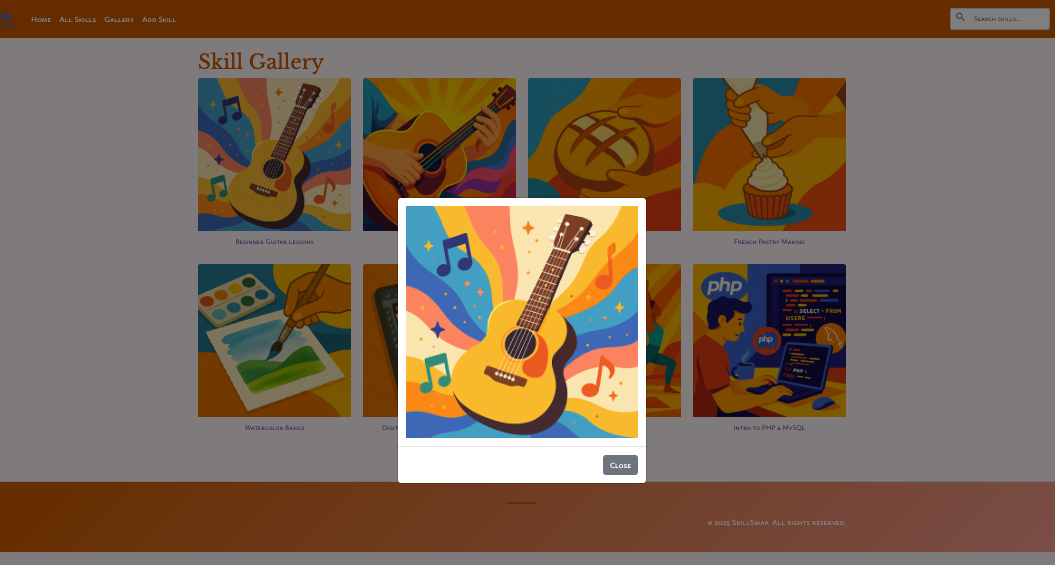
Skills page:



Gallery:



Gallery with modal:



Add skill:



JavaScript validation for adding images:

