

ECM1417
Web Development
Project Specification
Mini-game website



Haozhe Wang and Matt Collison
February 2023

This document outlines the assessment specification and submission requirements for the ECM1417 Web Development module.

1 Requirements

In this assignment you will deploy a LAMP stack web server to your Azure VM and design a website that enables a mini-game to be played in a browser. The requirements detailed below specify exactly how this website should be built.

1.1 Web server architecture

The web server will be hosted on the Microsoft Azure platform that was introduced in week 2 and the code for your submission will be sent to bart. Each student has been allocated an individual VM on which you will be expected to configure and deploy your LAMP stack. This means on your Linux VM you will need to install and run Apache httpd and php.

1.2 Website design

1.2.1 Webpages

Four webpages should be created using php to run the website. The landing (index) page, registration page, leaderboard page and the game page. These webpages must use the following names:

1. index.php
2. registration.php
3. pairs.php
4. leaderboard.php

1.2.2 Page content and layout

All pages will have a shared navbar menu at the top of the pages. The navbar menu must include the following content items with their corresponding name attribute:

1. Home - name="home"
2. Play Pairs - name="memory"
3. If the user has registered a profile display a link to leaderboard
 - Leaderboard - name="leaderboard"
4. else, if the user hasn't registered a profile display a link to register (without leaderboard)
 - Register - name="register"

The navbar items must use the Verdana font in bold with fontsize 12px. The Home item should be aligned on the left and the Play pairs and Leaderboard items on the right. The navbar background must be 'blue'. If an emoji has been selected as the user image in the registration process, this should also be shown on the navbar.

Each webpage must then contain the rest of the content for the webpage in a div with id attribute value of "main". The main div will have a background image, this must be the Arcade image (arcade-unsplash.jpg) linked below, set at the full width of the viewport and in a absolute position in the centre, vertically and horizontally.

Image source: Photo by [Carl Raw](#) on [Unsplash](#)

Advanced layout features: To make the webpage more advanced you can use bootstrap for mobile and responsive layout. You can also add animations to your game to make it more engaging for the

player. For example, when the player clicks on a card, you can add a flipping animation to simulate the card turning over. You can also add an audio track that starts when the game is playing.

All layout and content features will be assessed across all pages.

1.2.3 Landing page

The landing page (index.php) must display the welcome message “Welcome to Pairs” if the user is using a registered session, followed by a button with the content “Click here to play” that contains a hyperlink to the pairs.php page. Alternatively, if the user isn’t using a registered session the landing page should provide a paragraph with the content “You’re not using a registered session? Register now” where the Register now text has a hyperlink to registration.php.

1.2.4 Registration

The registration page, registration.php, will set session variables and cookies to generate a profile for the user. Note, there all user profile information should be stored in cookies and there is no need for a database.

The user profile registration form will include:

- Username/nickname
 - If the input contains any of the invalid characters, the program should return an error message and show this message under the <input> tag to the user to enter a new username. The invalid set includes: " ! @ # % & * () + = { } [] — ; : “ ’ < > ? /
- Avatar selector/generator
 - Simple - default image is set as avatar for all users
 - Medium - select an image from set of images
 - Complex - select and configure features to assemble an emoji avatar/image

1.2.5 Pairs webpage and emoji matching game

‘Pairs’ is a memory game that is played by selecting pairs of cards that are face down on the table. This page , pairs.php, will enable users to play this game virtually in the browser. This webpage will have a div with a grey background and 5px box-shadow where the game can be played. The game should implement the following features:

1. The game should start when the user clicks a button ‘Start the game’. The button should disappear when the game starts.
2. Simple:
 - The game should display 6 ‘cards’ (3 pairs) with preset images and allow two pairs to be turned at a time. If the cards match the cards should remain face up. The game should make a record of points scored based on how many attempts the user takes to match all the cards.
3. Medium
 - The game will display 10 ‘cards’ (5 pairs) and the emoji images on the cards will be generated by combining features (colour, smile and eyes) randomly for each run of the

game. The game should record the number of points and time taken. Points are calculated based on how many attempts and how much time the user takes to match all cards.

4. Complex:

- The user will progress through levels of the game with increasing number of cards in each level. As the levels progress, users will also have to match 3 and 4 cards rather than pairs. Again, random emojis configured from features should be used as the card images. The game should record the total number of points as well as the number of points per level for each user based on how many attempts and how much time it takes to match all cards. The background colour of the content div should change to gold (#FFD700) during gameplay if the user exceeds the previous best score for the current level.

On completion of the game if the user is in a registered session they should be shown their score and asked if they want to submit their score or 'Play Again'. If the user clicks 'submit' this should send a post request to the leaderboard page. If the user clicks 'Play again' you should restart the game and reset the score to zero.

Here is alternative plain english description of the game.

The game page: Once you have the basic layout of the game board, you need to implement the game logic using JavaScript. The game should begin by shuffling the cards and placing them randomly on the game board. Then, when the player clicks on a card, the game should reveal the image on the card. If the player clicks on a second card, the game should check if the two cards match. If they do, they should remain face up and the player should be allowed to click on another pair of cards. If they don't match, they should be flipped back over and the player should try again.

Add game logic: Once you have the basic interactivity working, you can start adding the game logic. When a user clicks on a card, you should keep track of which cards have been clicked and compare their contents. If the contents match, keep the cards flipped over. If the contents don't match, flip them back over and allow the user to try again.

Add game state management: You'll need to keep track of the state of the game, such as the number of moves the user has made, the time elapsed, and whether the game has been won or lost. You can use variables to store this information and update it as the game progresses.

Add win and lose conditions: Finally, add win and lose conditions to the game. The game should end when all the cards have been matched or when the user has made too many incorrect guesses.

Timer and score: You can add additional features to the game such as a timer and score to make it more challenging. For example, you can set a time limit for the player to match all the cards and keep track of the number of attempts it takes the player to complete the game.

1.2.6 Leaderboard

The leaderboard page, `leaderboard.php`, will have a div with a grey background and 5px box-shadow where a formatted table that contains usernames and best scores will be displayed. Complex solutions with levels will be display best scores per level as well as total best scores. The table should have border spacing at 2px and the background for table header cells set as 'blue'.

To enable multiple users in the leaderboard the php code will need to populate an appropriate datastructure to store submissions from multiple users on different browsers. Note, no additional marks will be awarded for using a database for this feature and multiple users can be augmented by creating private browser sessions so session variables don't conflict and cookies aren't stored.

2 Mark scheme

Marking will be conducted on the live websites hosted on the Azure VMs led by a README file listing the features of the website that will be submitted to bart. The marks are split by web page and half of the marks for each webpage are dedicated to layout, styling and structure of the html. The other half of the marks will be awarded for the features and content. Where there is a simple medium and complex implementation; a simple implementation can score upto 50% of the marks for that section, medium upto 75% for that section and complex 100% for that section.

- Landing page - 10 marks
- registration page - 20 marks
- Pairs game page - 60 marks
- Leaderboard page - 10 marks

2.1 Submission

As well as deploying your website to the Azure VM, you also need to submit your code via bart. You should create a zip file containing all your html, css, js code and assets. At the root of your zip file please also include a README file that links to your Azure labs VM and provides a bullet point list of the features you've completed for each webpage. This README file will guide the marking process as we will not be able to discover all features without them being mentioned here. The README should not exceed 250 words and must only contain concise bullet points.

The Azure VM will be used as the primary resource for marking. The teaching team have the ability to 'start' your VM so the VM can be turned off for submission but will be launched by us for marking. Our marking process will progress as follows:

1. Download bart submission
2. Open README file
3. Launch VM and navigate to website in a browser
4. Open each webpage in sequence and check all features listed in README
5. Enter marks and feedback per section

Note, we will validate that the relevant VM has been linked from the README and any attempt to link to another student's VM will be considered as plagiarism.

3 Summary

In summary, this project requires students to develop and combine multiple components of a full-stack web application to deploy a feature complete website that enables a user to create use sessions and play a pair matching game.

If you have any questions at any point throughout the project please do not hesitate to get in touch, h.wang3@exeter.ac.uk and m.collison@exeter.ac.uk.