Pairs Game Report

Navigation Bar

The navigation bar is implemented in all pages of the website and wrapped in <nav> element with the class “navbar”. The navbar is further split into right and left sections showing different hyperlinks texts based on the authentication status of the user. The status is determined using “$is_registered” variable that checks whether a user is logged in by verifying the $_SESSION['username'] value. Such a dynamic navigation improves the user experience by ensuring that the user only sees the relevant options based on the session status.

Regardless of whether the user is registered or not, the left part of the navbar always contains a “Home” hyperlink that redirects the user to the landing page. However, the right side of the navbar dynamically displays hyperlink texts based on the user’s authentication status. If the user is registered and logged in, the navbar will include the links to the game and the leaderboard. Additionally, the user’s personalised emoji will also be shown at the right edge of the navbar, indicating that the user is logged in. This is done by accessing the session data and accordingly layering the <img> tags with each referencing the specific facial feature chosen during registration. If the user is not logged in, the navbar will only display the links to the game and the registration page.

The text in the navbar is styled in Verdana, bold, 12px font which makes the text clear and dyslexia-friendly thanks to the Verdana's wide letter spacing and distinct character shapes.

Index Page

The “index.php” and “index.css” files make up the landing page of the website featuring a welcome message. The page displays different UI components based on the authentication status of the user. The unregistered users are displayed with a hyperlink text to the “register” page. Registered users are presented with a personalised greeting based on the username and a button that redirects to the “game” page. The button is styled with a green, rounded box that contrasts with the arcade background and a white text that enhances the readability.

Notably, all the page content is contained within a with the class name “content-container”. It is styled to be a rounded, translucent, grey box which allows the user to easily distinguish the content from the colourful arcade background. This design choice is implemented in all the website pages, resulting in a consistent, uniform look that enhances the visual coherence.

Registration Page

Registration page is represented by “registration.php” and “registration.css”. The page displays a form where the user can set their name and create their custom emoji by choosing facial features presented to them. These include the skin colour, eyes and the mouth with various options arranged in rows. This is enabled through the emoji builder under the with the ID of "custom-emoji-options".

When the user clicks on an option, they trigger the “onclick” attribute which calls the JavaScript functions (selectSkin, selectEyes, selectMouth) to update the selected option, indicated with a blue outline. Another JavaScript function “selectOption” ensures that the user can select one option at a time, preventing overlapping facial features.

The page also displays a preview of the emoji for better visualisation and real-time feedback. Additionally, the outline of the options turns blue when hovered over, improving the visual clarity. If the user does not select any custom facial features, a default emoji will be automatically set and assigned to them, ensuring that all users have an emoji. The facial features are then stored in the session to be accessed and assembled in other pages of the website.

The username is restricted to exclude special characters: “! @ # % & ^ * ( ) + = { } [ ] — ; : " ' < > ? /”. These characters are displayed under the input field in a <small> element for user guidance. Such a limitation helps to prevent injections and unwanted formatting issues. This is ensured by iterating through each character submitted in the POST data. If the username is empty or contains any of the specified characters, an error message will be returned based on the value of “$error_message”. The error message is styled to appear in a red box, clearly indicating the error to the user. In case of a successful registration, the username will be stored and accessed throughout the website.

Pairs Game Page

This main game page is implemented through “pairs.php” and “pairs.css” files. On this page, the user is presented with a green button that starts the game. It is a classic game of memory where the player clicks on the cards to can the images behind them. The goal of the game is to match all the cards with the same images before using up all 30 attempts on each level. Each correctly matched pair of cards glows green, indicating a successful match. There is also a points system where the player races against the time to match the cards with the fewest attempts. All the relevant information such as the current level and attempts left is displayed in a dashboard above the game. The game itself features 5 levels of difficulty, with each level having more cards in play and more cards to be matched. Additionally, the game’s background turns gold when the player reaches the highest score for the level, greatly increasing the engagement.

The images behind the cards are emojis made of .png files of different facial features. They are randomly generated through “generateEmojiSet” function and there are 108 possible combinations. These combinations are later layered and styled in CSS to make up a proper emoji. The function “createGameBoard” creates the DOM elements for the cards including the ID’s. Another function “flipCard” handles the logic when a card is clicked. It checks whether all cards flipped have the same ID to verify a successful match. It is worth noting that an initial implementation had an issue of repeating combinations. In rare instances, a pair of identical emojis would repeat, but they could not be matched because of different ID’s. This was later solved by incorporating a “usedCombinations” set that keeps track of already generated emojis, ensuring that duplicates are not generated.

At the end of each game session, the player is presented with their results which includes the final score, time spent, attempts made, and levels completed. In addition to the results, different buttons are displayed based on the authentication status of the user which is checked through <php> “$is_registered” variable. Both the registered and unregistered users are given an

option of “Play Again” button, but only the registered users can submit the scores to the leaderboard. Unregistered users are instead suggested to register with a hyperlink button to the registration page.

The submission starts with the “Submit Score” button. It triggers the “submitScore” function that collects all the relevant data such as score, and levels completed which is then sent as a POST request to the same page. The response is handled with <php> where it extracts and processes the data. It is then loaded into a “leaderboard_data.json” that stores the data in a standardised format. The .json file is filtered to find the highest scores for each level through “updateHighScoreBackground” function. During a game session, it constantly compares the current score to the highest score to determine whether the gold background should be displayed. The scores submitted include both the final score and the levels’ scores. However, the level scores are only saved if they are completed while the final score adds up the level scores and the score achieved in the last attempted level.

The scoring system is based on the time spent and the success of the attempt and is entirely implemented through “calculateLevelScore” function. The player starts off with 200 points and each level cleared is additional 200 points. An incorrectly guessed pair subtracts another 2 points, while a successfully matched pair scores 20 points. Additionally, 1 point is deducted for each second spent which adds pressure and discourages the player from taking too much time on memorisation. Passing all 5 levels will grant additional 200 points to signify the achievement. It is important to mention that there is a known issue with this implementation. If the player deliberately takes too long and fails to match any pairs, the displayed score on the dashboard may freeze, while the actual score goes negative. This could confuse the player. However, in a realistic scenario, this is unlikely since most players would match at least one pair before excessive time passes.

The entire game and the content are contained in a single “content-container” . It is styled in translucent, grey box to match the uniform design throughout the website. The player dashboard, that displays all the relevant information, is contained in a darker grey box which improves the readability and helps players to focus on important information. The cards are styled in blue and arranged in rows. When clicked, they visually flip, making the gameplay more intuitive and engaging. Notably, it was difficult to decide how to arrange cards. The CSS declaration “grid-template-columns: repeat(3, 1fr);” would leave empty gaps, creating visually unappealing grid. The solution was to arrange cards in the number of columns based on the number of pairs of cards in the .php file instead of .css. This resulted in consistent grids with no gaps.

Leaderboard Page

The leaderboard page is made up of “leaderboard.php” and “leaderboard.css”. This page shows all the scores submitted by the registered players and ranks them based on their scores. All the scores are parsed and extracted from the JSON and put into a table. All player submissions are filtered and ranked in a descending order based on the score using the php sort functions. Meanwhile, the UNIX timestamps stored in the JSON are converted into a human-readable format of 'Y-m-d H:i'.

There are 2 different leaderboards in 2 separate tabs: “Total Scores” and “Level Scores”. The former contains the information about the player’s rank, name, total score, levels completed and the date while the latter only includes the rank, name, level score and the date. However, because there are 5 different levels, “Level Scores” tab also contains a drop-down menu which allows the user to choose which level’s table to display.

The tables are made <table> HTML element and are styled to have border spacing of 2px while the header colour is blue. The tabs are also blue, but the active tab is of a darker shade to indicate that is it currently open. This is enabled through “switchTab” function that assigns “active” class to the tab that has been clicked and CSS styling based on it. The text in the table is in white Arial font which nicely contrasts with the background and the blue header, resulting in a visually appealing sight.

Log Report

Date	Time	Duration/hours
07/03/2025	18:00	2
07/03/2025	18:00	2
08/03/2025	16:00	2.5
08/03/2025	22:00	1
10/03/2025	11:00	1.5
10/03/2025	15:00	1
10/03/2025	17:00	1
11/03/2025	18:00	2
12/03/2025	18:30	2
13/03/2025	18:00	3
14/03/2025	20:30	4
15/03/2025	12:30	2
17/03/2025	12:00	3
17/03/2025	22:00	0.5

Declaration

AI-supported/AI-integrated use is permitted in this assessment. I acknowledge the following uses of GenAI tools in this assessment:

[ ] I have used GenAI tools for developing ideas.
[ ] I have used GenAI tools to assist with research or gathering information.
[ ] I have used GenAI tools to help me understand key theories and concepts.
[ ] I have used GenAI tools to identify trends and themes as part of my data analysis.
[ ] I have used GenAI tools to suggest a plan or structure for my assessment.
[✓] I have used GenAI tools to give me feedback on a draft.
[ ] I have used GenAI tool to generate images, figures or diagrams.
[✓] I have used GenAI tools to proofread and correct grammar or spelling errors.
[ ] I have used GenAI tools to generate citations or references.
[ ] Other [please specify].
[ ] I have not used any GenAI tools in preparing this assessment.

I declare that I have referenced use of GenAI outputs within my assessment in line with the University referencing guidelines.