*Coursework 1:   
Game Website*

*Place cover sheet here*

**Introduction**

The plan for my coursework is to create a speed running maze game, with visual effects that reward fast players and a leaderboard system to compete against.

The website will match the maze theme with solid lines and high contrasts. The mazes are not randomized. There will be a series of levels that each player will complete, the objective being to complete each level as fast as possible.

Players can compete to gain the fastest time on each level, or for a total time for all the levels. I aim to create a lowered field of view with a dark radius around the player so that memorizing the maze layout for each level is key to gaining the best times.

I also aim to have music that plays only when the player moves, so consistent movement without stopping is the most visually rewarding.

**Section 1: Website**

I have a general idea for the website planned out, but I’ll build upon it as I extend it.

The first window will be the registration / login page. Players will be prompted to register if they haven’t created an account already, with details being stored in JSON format as requested. Duplicate registrations will not be allowed.

Once an account has been created, or they already have an account, the player can login with their details. Incorrect details will be prompted with an error message.

Once successfully logged into the account, the game page will be accessible.

**The Registration / Login page** will feature input fields such as username, email, password and allow the player to create an account that will be stored in local storage. The login page will allow the player to login using these details and access the game.

**The Profile Page** will feature the accounts personal stats and the ability to input additional information. //POSSIBLY

**The Game Page** will feature a home button in the top left. The main screen will have the game in the centre, a video background of maze gameplay, and a menu above to select between different pages. (this menu will become a dropdown at certain sizes).

**The Leaderboard Page** will feature a leaderboard of individual times per level & total times for all levels. These will be listed under the associated username with the account.

**The About Page** will feature a basic description of the objectives of the game and possibly the story behind it.

**Registration / Login Page**

**A screenshot of a computer

Description automatically generated**

The registration page features a clean colour scheme and 4 input fields.

Users can enter their username, email, password and a repeat password field ensures that the password is entered correctly.

There is also a show password checkbox, a button to submit the form and a link to login if the user already has an account.

**A screenshot of a computer

Description automatically generated**

The login page only contains the email and pasword fields, the login button and a link back to registration.

Error handling is done in javascript.

**Registration Page Errors**

* Missing Username Field
* Missing Email Field
* Missing Password Field
* Username contains more than 20 characters
* Password contains more than 20 characters
* Password contains less than 8 characters
* Password does not contain a letter
* Password does not contain a number
* Password does not contain a special character
* Password does not match
* Account already exists

A screenshot of a computer screen

Description automatically generated A screenshot of a computer screen

Description automatically generatedA screenshot of a login form

Description automatically generated

**Login Page Errors**

* Missing Email
* Missing Password
* Incorrect details
* Email not found

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated

**Error Handling, Registration & Login**

All of the validation functions are completed within the same javascript file.

Variables are created for each input from the form and for the error message container.

We have an eventListener that listens for the form submittal either by button or by enter key. ‘errors’ is created as an empty array.

We check if the form has a username\_input field as only the registration form has this.   
  
**IF username\_input is present**

We call the function getRegisterFormErrors with username, email, password & repeatPassword.

This function checks all of our conditions, such as the username field not being null or the password having a minimum of 8 characters.

Each error we have, we pass into our array, and we add the incorrect class to the element where the error is found (This gives it the red styling).

We then return errors.

**ELSE**

We call getLoginFormErrors with email & password. We check this in the same way.

**if(errors.length > 0)**

We prevent form submission, this simply means if the length of errors is greater than 0, meaning if we have at least 1 error.

We list all the error messages.

**ADDITIONALLY**

When the user interacts with an input field, we check each input and if it has an incorrect class, we remove it and remove all error messages.

**ELSE**

Now that we have checked for errors and there are none, we again check

**IF username\_input is present**

We know that we are in the registration form. We turn the form data into a javascript object, turn this into a string, and create a unique key based on the email.

We then check if this key matches an existing key, meaning the account / an account with the email already exists. If it does, we prevent submission and display this error.

**ELSE**

We submit this to localstorage with it’s unique key.

**ELSE**

If username\_input is not present, we know we are in the login form. We instead check the inputted email and password against the data we have stored in localstorage.

We have appropriate errors for incorrect passwords or incorrect emails, and if all is correct then we change windows to the home page.

Ok so

We build the game

Game needs at least 5 levels that it transitions through, simple mazes, we add the timer, we add the win condition, we save the score.

Score shows at the end

Leaderboard accesses the score

Profile access the score if we’re still doing that

No problem

For example

When we go over the event tile, we stop the timer and output the timer to a variable, nothing we can figure out

YOU’LL BE FINEEEE