



CST2120 Web Applications and Databases

Coursework 1: Game Website

Final Submission

Dereck Lam Hon Wah

M00826933

Date of Submission: 09.12.22

Lab Tutor: Mrinal Sharma

Table of Contents

Introduction	4
Pages	4
Common Elements	4
Homepage (index.php).....	4
Game Page (game.php)	6
Explore Page (explore.php)	8
Leader board Page (leaderboard.php).....	9
Login Page (login.php)	10
Registration Page (register.php)	12
Support Page (support.php)	14
Conclusion	15

Table of Figures

Figure 1 Session Storage.....	4
Figure 2 Homepage displaying a log in user	5
Figure 3 Homepage displaying log out user	5
Figure 4 Homepage Youtube Video	6
Figure 5 Game Page.....	7
Figure 6Game page menu	7
Figure 7 Explorer Image Slider.....	8
Figure 8 Explore Image Slider (Hover On)	9
Figure 9 Explorer Image Slider in action.....	9
Figure 10 Leaderboard sorted table	10
Figure 11 Login Page.....	11
Figure 12 Login page Error message displayed.....	12
Figure 13 Registration Page Form	13
Figure 14 Registration Page Validation Error Message	13
Figure 15 Support Page.....	14
Figure 16 Support Page Validation Error Message	15

Introduction

Using HTML, CSS, PHP, and Vanilla Javascript, Space Frenzy is a space invader/shooter clone where the player controls a spaceship and fights against waves of aliens in space. The game currently consists of two working difficulty-working levels where the player can test his focus and determination by shooting down some aliens.

Pages

Common Elements

Common elements with similar functionality, like the navigation, footer, and play button, are present on the different web pages. The navigation consists of a list of web page links to redirect the user to a specific page, a footer consisting of a list of web page sections, and the social media links of the developer.

The play button's purpose is to redirect the player to the game page. However, it directs the user to the login page if no player is found logged in by checking if the `loggedInUser` key value is undefined in the session storage.

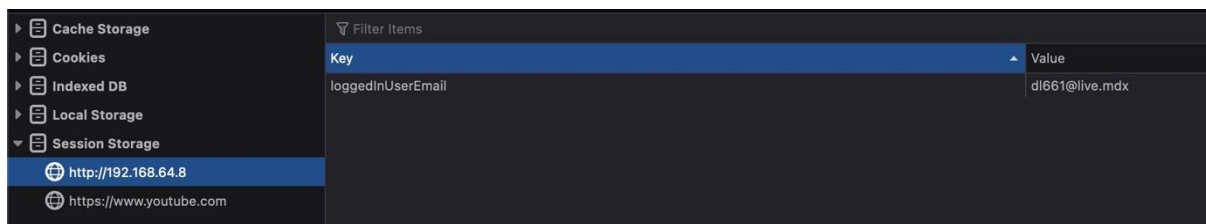


Figure 1 Session Storage

If a user logged in, the session storage `loggedInUserKey` is updated and will display its username on all web pages. Moreover, depending on this value, the login link will shift in appearance and functionality. When a user is not logged in, it will display a login link which redirects the user to login page. However, when a user is not logged in, it will display a log out link which will logged out the user and delete the `loggedInUserKey` in the session storage.

Homepage (index.php)

The homepage will introduce the player of the webpage functionality and an introduction of the game. When the user hover on the image next to "What is space frenzy", it will display a play button mentioned above in common elements. A youtube video is displayed to show where inspiration has been taken and it can be played.

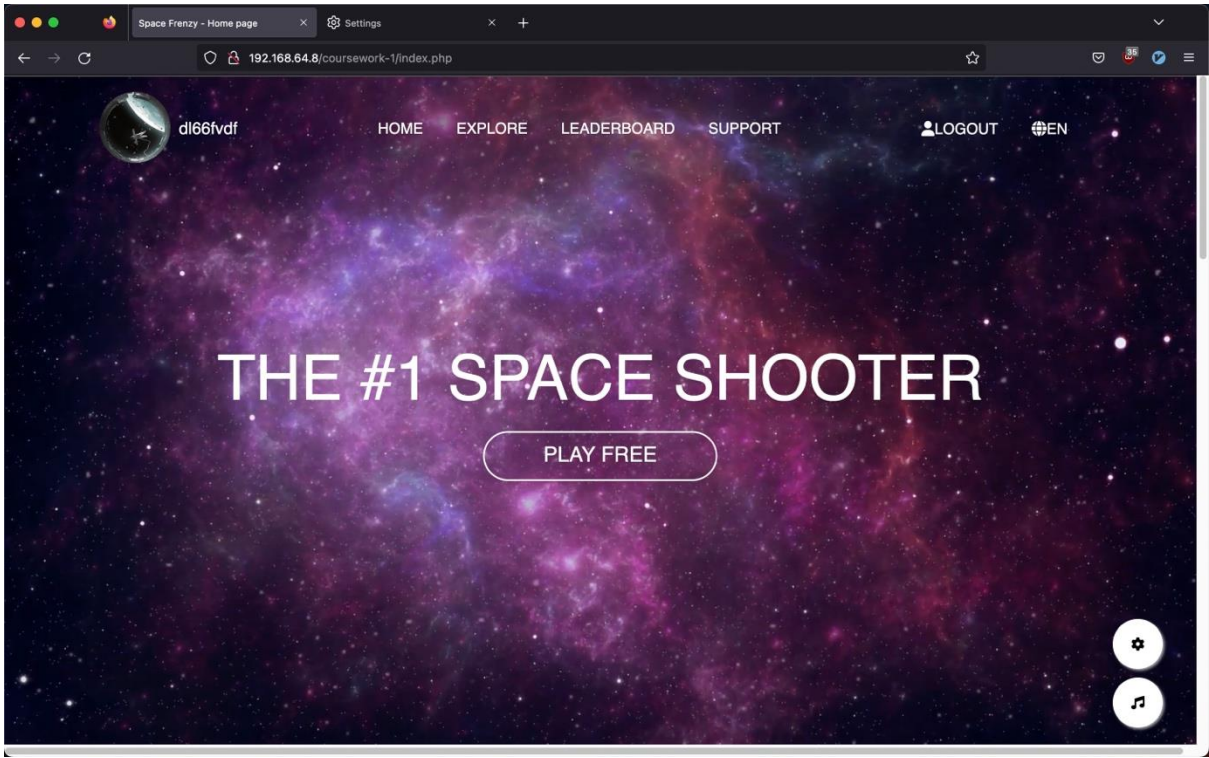


Figure 2 Homepage displaying a log in user

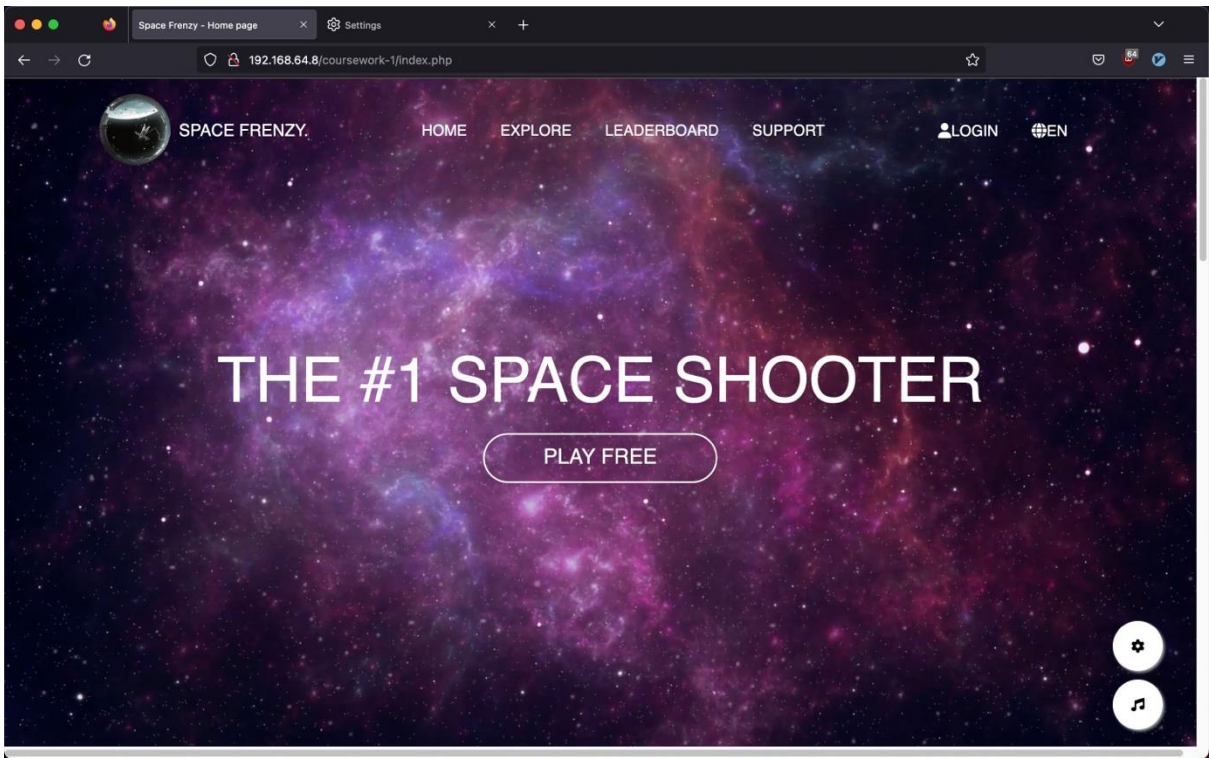


Figure 3 Homepage displaying log out user

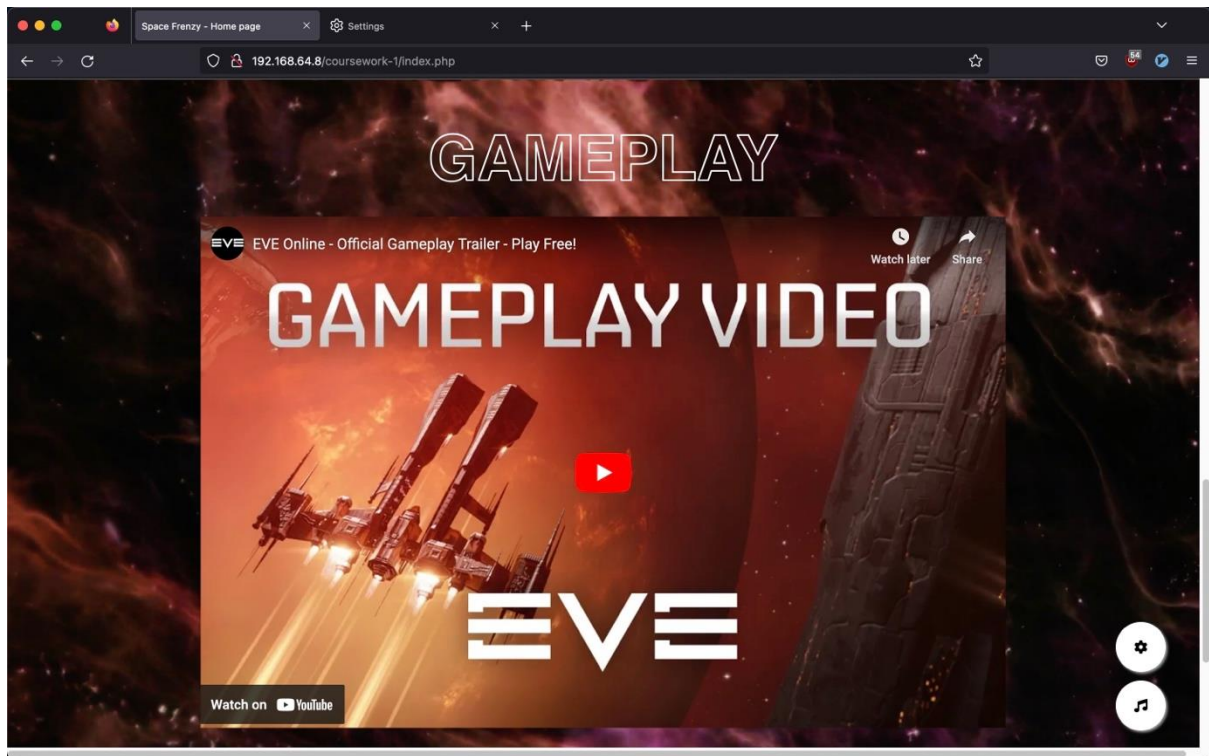


Figure 4 Homepage Youtube Video

Game Page (game.php)

These are the levels:

- Beginner difficulty - the player must eliminate every alien without coming in contact with one, or else they will perish. The aliens advance and move toward the player's spaceship each time they make contact with the game's window walls.
- The Normal level uses the same controls as the Beginner level. A large alien boss has entered the waves and needs five bullets to be eliminated.

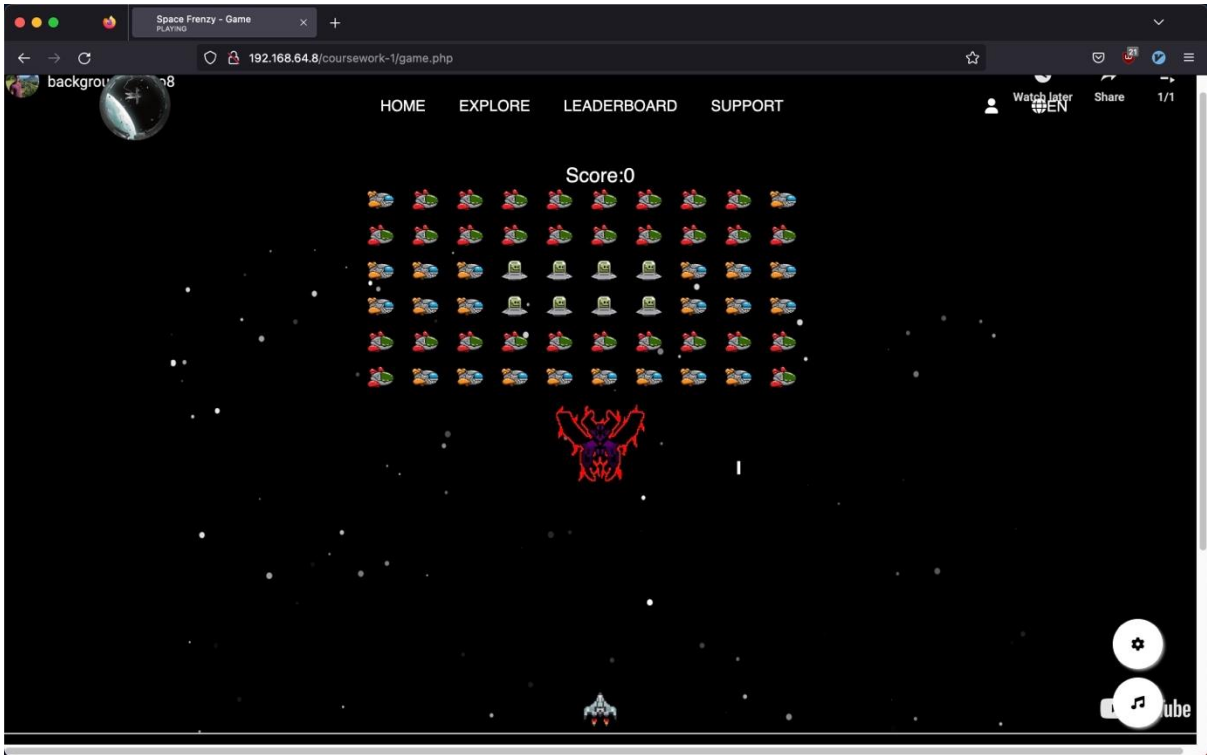


Figure 5 Game Page

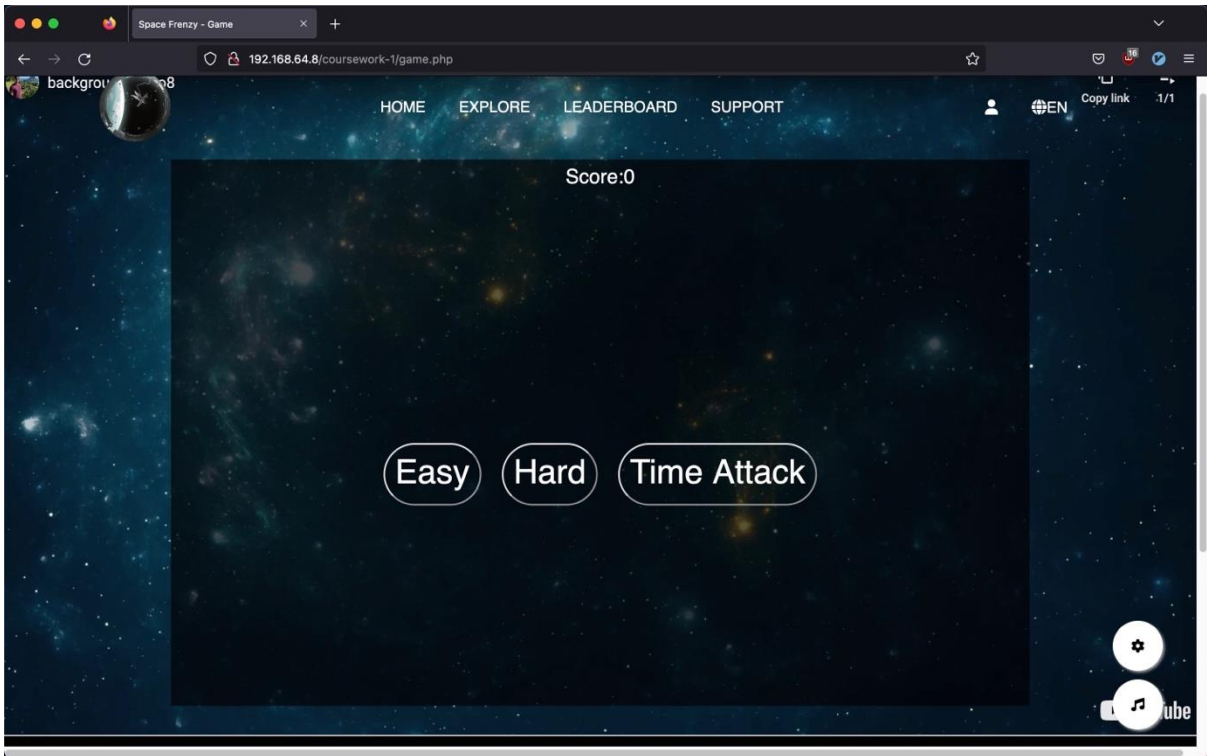


Figure 6Game page menu

Explore Page (explore.php)

The explorer page contains an overview of the different modes and a how-to-play section of the game. The modes' overview is displayed in an image slider using CSS and javascript to slide automatically among the images. When the mouse hovers over the image, a description of the specific mode renders. It follows with a how-to-play section where the user is provided with an instruction list of the available controls.

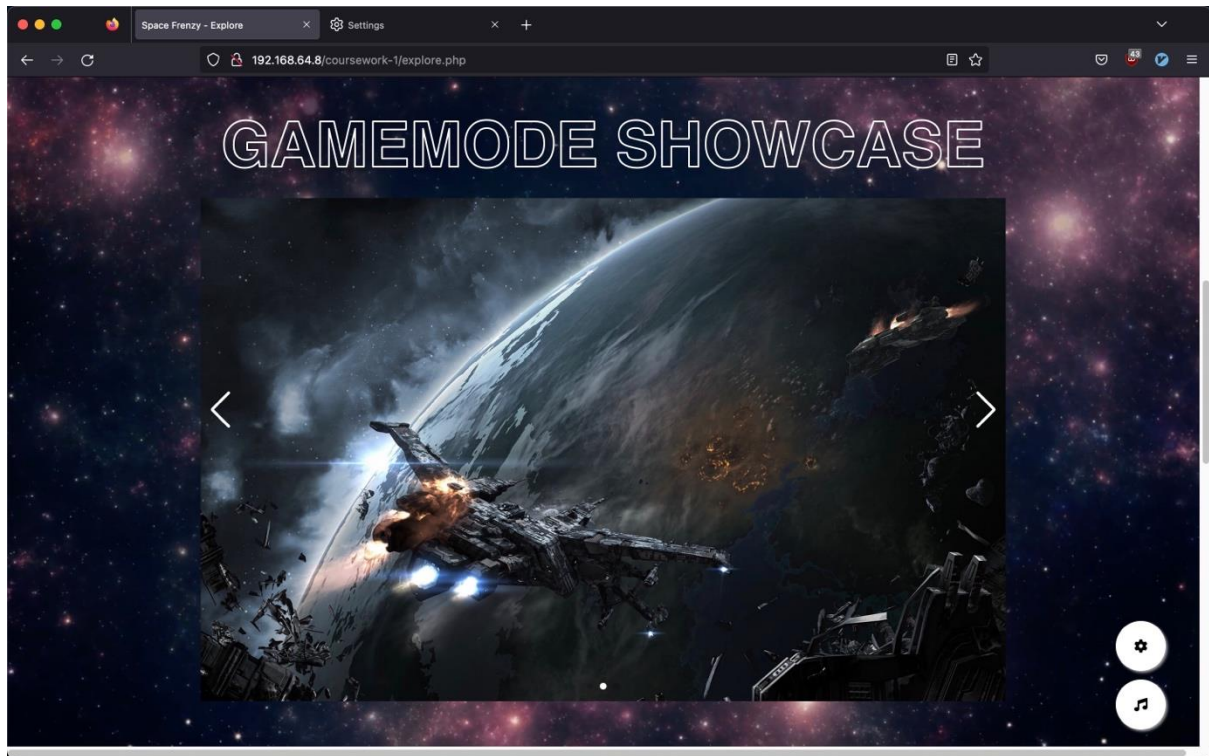


Figure 7 Explorer Image Slider

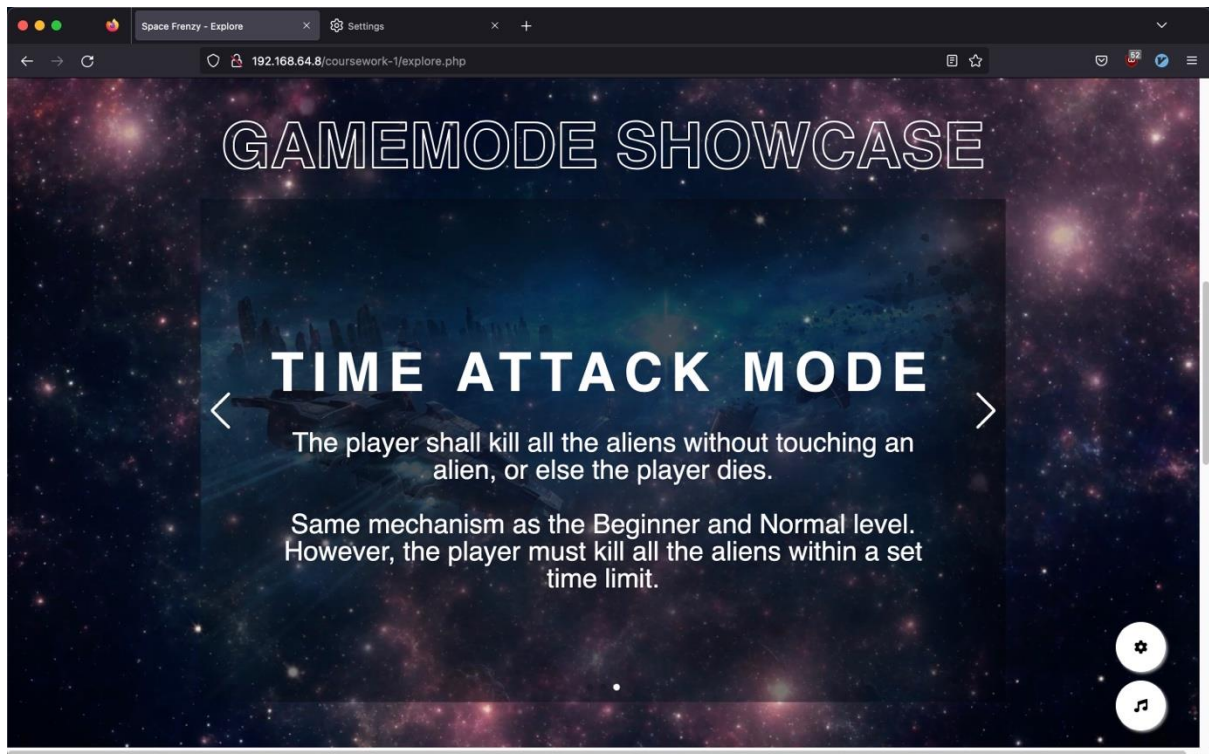


Figure 8 Explore Image Slider (Hover On)

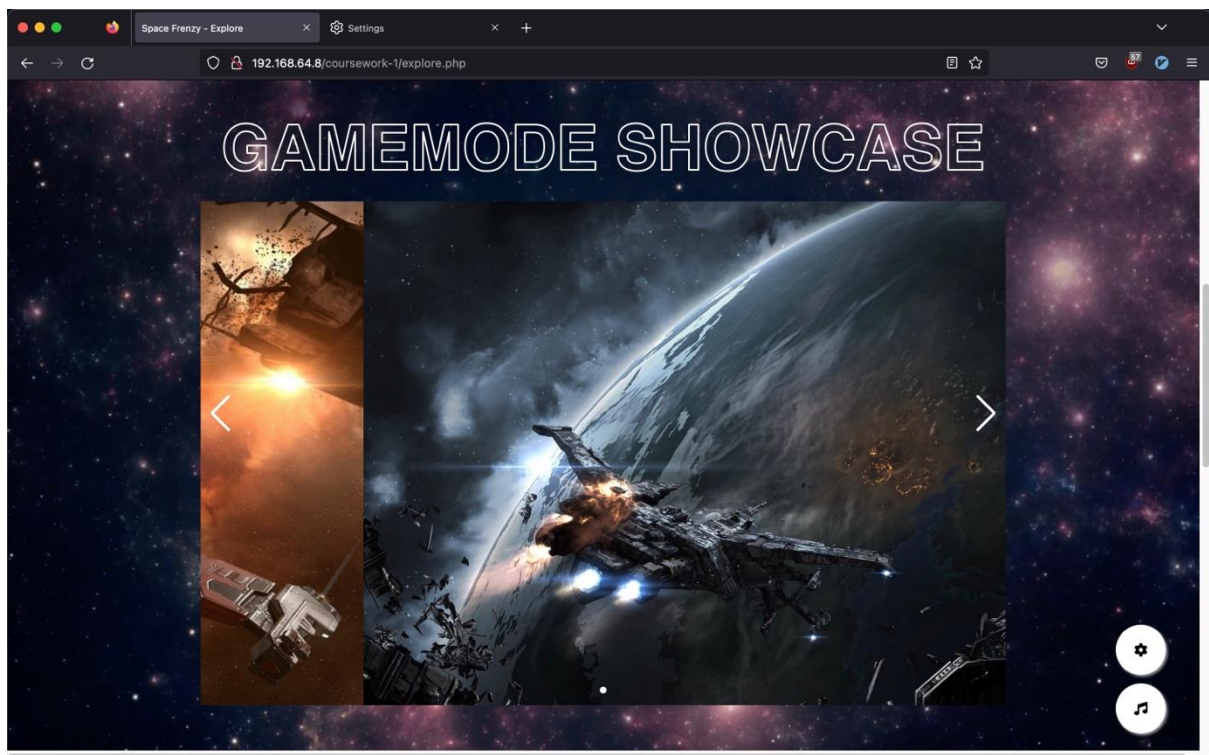


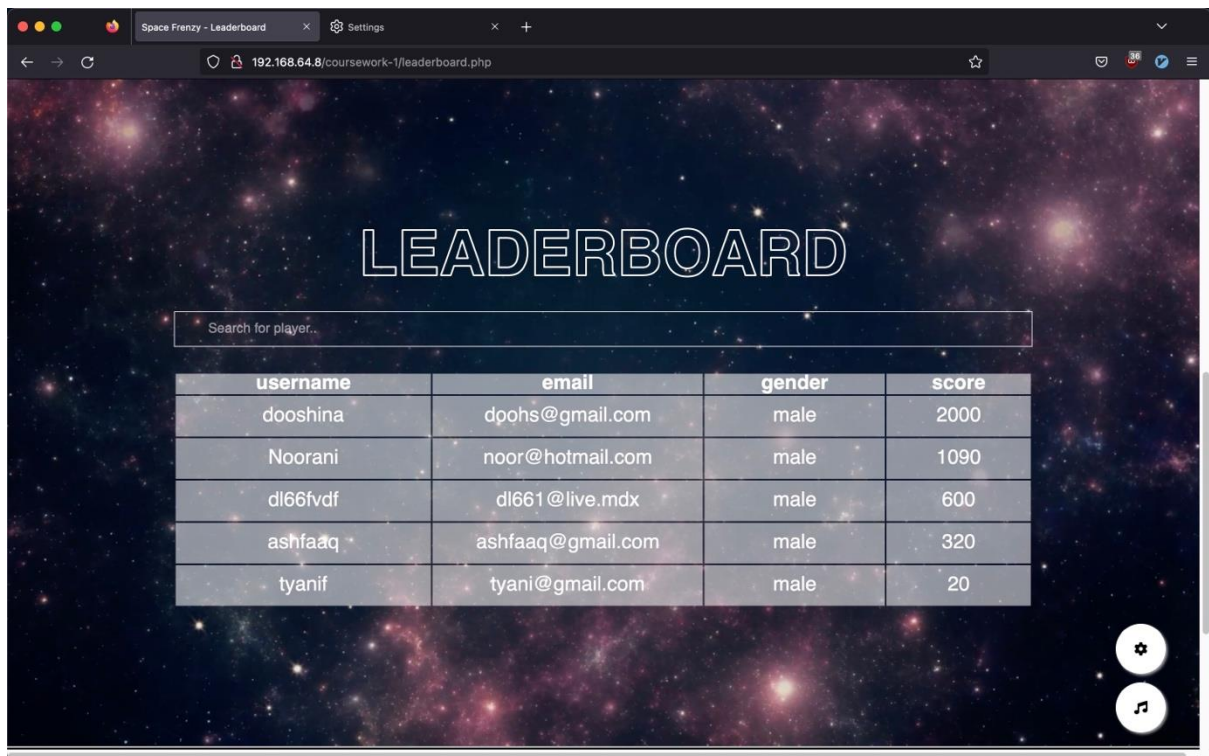
Figure 9 Explorer Image Slider in action

Leader board Page (leaderboard.php)

The leaderboard page consists of a sorted table in descending order from the highest scorer player to the lowest. When the leaderboard page finishes loading, a script executes to retrieve an array of user objects using an array of keys returned from the local storage. Local

storage JSON data is parsed to an array of javascript objects and executes a merge sort. A merge sort is basically dividing one large array, dividing and sorting it into smaller paired arrays, and combining them into a large, sorted array. When the merge sort is over, the DOM interface for table `HTMLTableElement` will render a leaderboard dynamically for the user.

Each time the user visits the leaderboard page, it updates the table following any game played. The local storage updates the player's scores each time the user wins or loses. The scoring system is cumulative and when the user wins a game, his score increments in proportion to the number of aliens killed. However, when it is game over, the score will increment by half the number of points obtained.



username	email	gender	score
dooshina	dpohs@gmail.com	male	2000
Noorani	noor@hotmail.com	male	1090
dl66fvd	dl661@live.mdx	male	600
ashfaaq	ashfaaq@gmail.com	male	320
tyanif	tyani@gmail.com	male	20

Figure 10 Leaderboard sorted table

Login Page (login.php)

The Login Page is essential to play the game, only log in user is allowed to access game page. When a user has successfully log in, his email address is stored in the session storage as the value of the `loggedInKey`. This key will be checked multiple times in different aspects of the website to check if any user is logged in. Some validations have been implemented in JavaScript to check the data submitted. If any doesn't match the validation, an error message is popped onclick of the login button.

1. Checks if the email address and password fields are empty.
2. Checks if any email address or password fields are empty.
3. Checks if email address inputted matches any email address found in the local storage, if email address not found the local storage display "undefined" value.

4. Checks if matched email address's password in local storage matches with the password inputted.

If the user has successfully login, he is redirected to explore.php and its username and logoff link are displayed in the navigation.

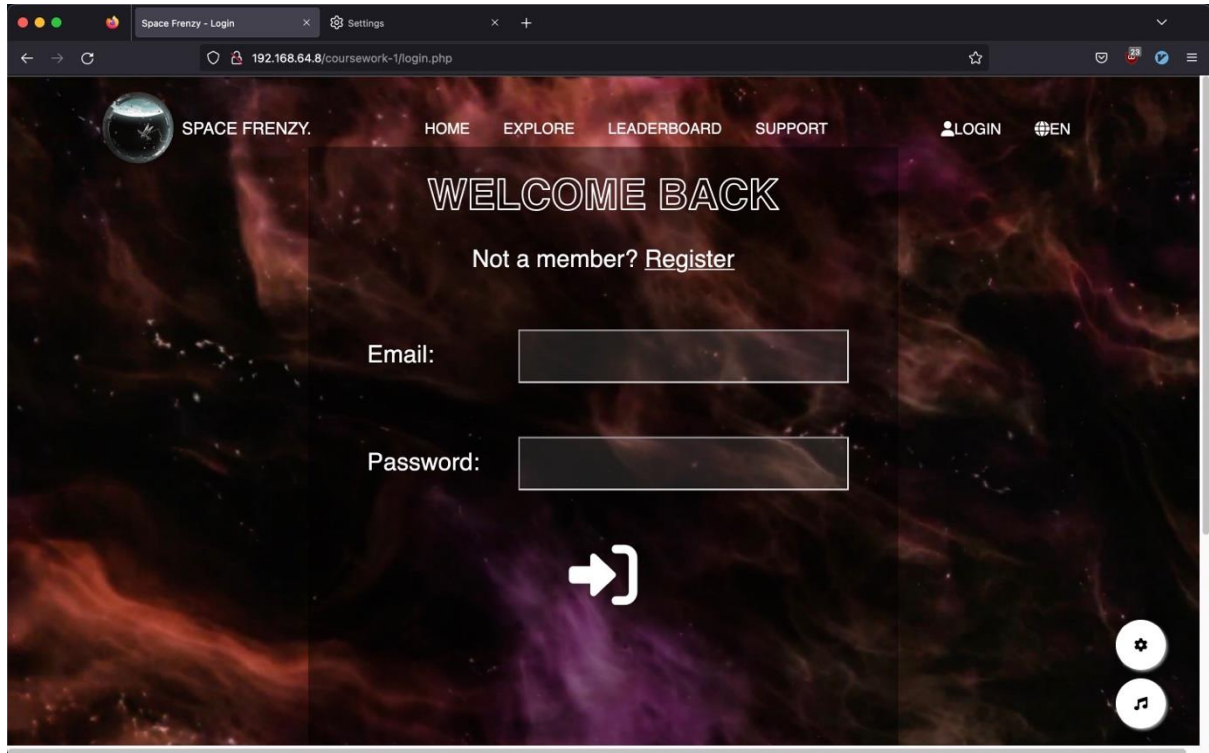


Figure 11 Login Page

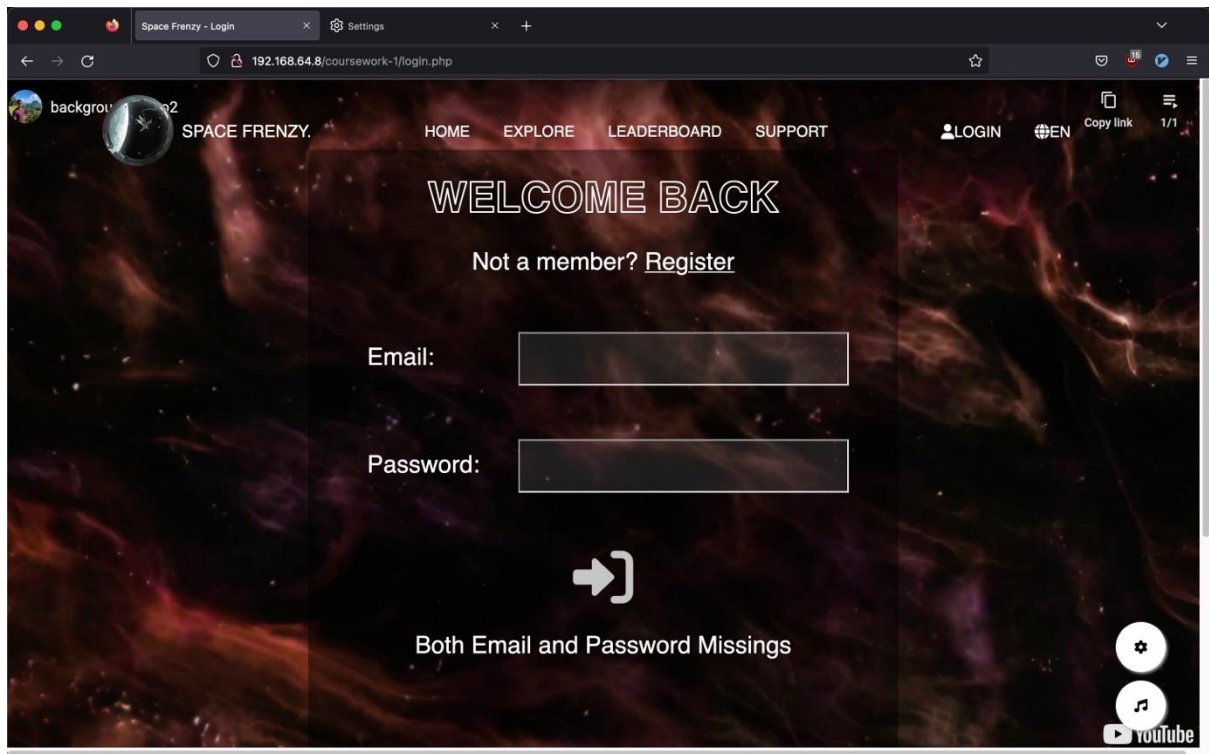


Figure 12 Login page Error message displayed

Registration Page (register.php)

The registration form validation is performed using javascript and regular expressions. Upon validation, the script will return an userObject, stringify it and add it to the local storage by keeping the email address as the key.

The validations are as follows:

1. Check if any fields are empty.
2. Check username consists of 6-10 characters and contain only alphanumeric characters and underscore.
3. Check email address is a string seperated into two parts by @ symbol, a "personal_info" and a domain.
4. Check password should have at least one number, one uppercase and lowercase character, one special symbol and be between 6-20 characters long.
5. Check Retype Password should match Password entered.
6. Check if checkbox has been ticked.

After the form validation, it is stored in an object, converted into JSON and stored in the HTML local storage.

The screenshot shows a web browser window with the URL `192.168.64.8/coursework-1/register.php`. The page has a dark space-themed background with a nebula. The navigation bar includes 'SPACE FRENZY.' with a logo, and links for 'HOME', 'EXPLORE', 'LEADERBOARD', 'SUPPORT', 'LOGIN', and 'EN'. The main heading is 'GETTING STARTED'. Below it is a link 'Already have an account? [Log In](#)'. The registration form contains the following fields: 'Username:' (text input), 'Email:' (text input), 'Gender:' (dropdown menu with 'Male' selected), 'Password:' (text input), and 'Re-type Password:' (text input). Below the form is a list of four validation error messages, each preceded by a red 'X' icon: 'A username consists of 6-10 characters and contain only alphanumeric characters and underscore.', 'A valid email address is a string seperated into two parts by @ symbol, a "personal_info" and a domain.', 'A valid password should have at least one number, one uppercase and lowercase character, one special symbol and be between 6-20 characters long.', and 'Retype Password should match Password entered.' At the bottom of the form is a checkbox labeled 'I agree with the terms and conditions.' and two buttons: 'SAVE' and 'CANCEL'. On the right side of the page, there are two circular icons: a gear for settings and a music note for audio.

Figure 13 Registration Page Form

The screenshot shows the same web browser window as Figure 13, but with some fields filled out. The 'Username:' field contains 'bgfbfqb'. The 'Password:' field contains '*****'. The 'Re-type Password:' field is empty. The validation error messages are now: 'Valid username entered.' (preceded by a green checkmark), 'A valid email address is a string seperated into two parts by @ symbol, a "personal_info" and a domain.' (preceded by a red X), 'Valid password entered.' (preceded by a green checkmark), and 'Retype Password should match Password entered.' (preceded by a red X). The 'SAVE' button is now highlighted with a blue border.

Figure 14 Registration Page Validation Error Message

Support Page (support.php)

It will allow the current registered user to change his current password to a new one. If a user desires to change his password, he will have to pass some validation in pure javascript and if validation fails, a pop up message will displayed on the screen.

1. Checks if a user is currently logged in the session storage.
2. Checks if the current, new and retype fields are empty.
3. Checks if either current, new or type fields are empty.
4. Checks if current password matches the user's password in local storage.
5. Checks if new password does not match the current user's password in local storage.
6. Checks if new password is have at least one number, one uppercase and lowercase character, one special symbol and be between 6-20 characters long through a reg exp check.
7. Checks if new password matches retype password to confirm the user has crosschecked his updated password.

If the new password has passed all validation, it will be updated in the local storage and user redirect to explore.php.

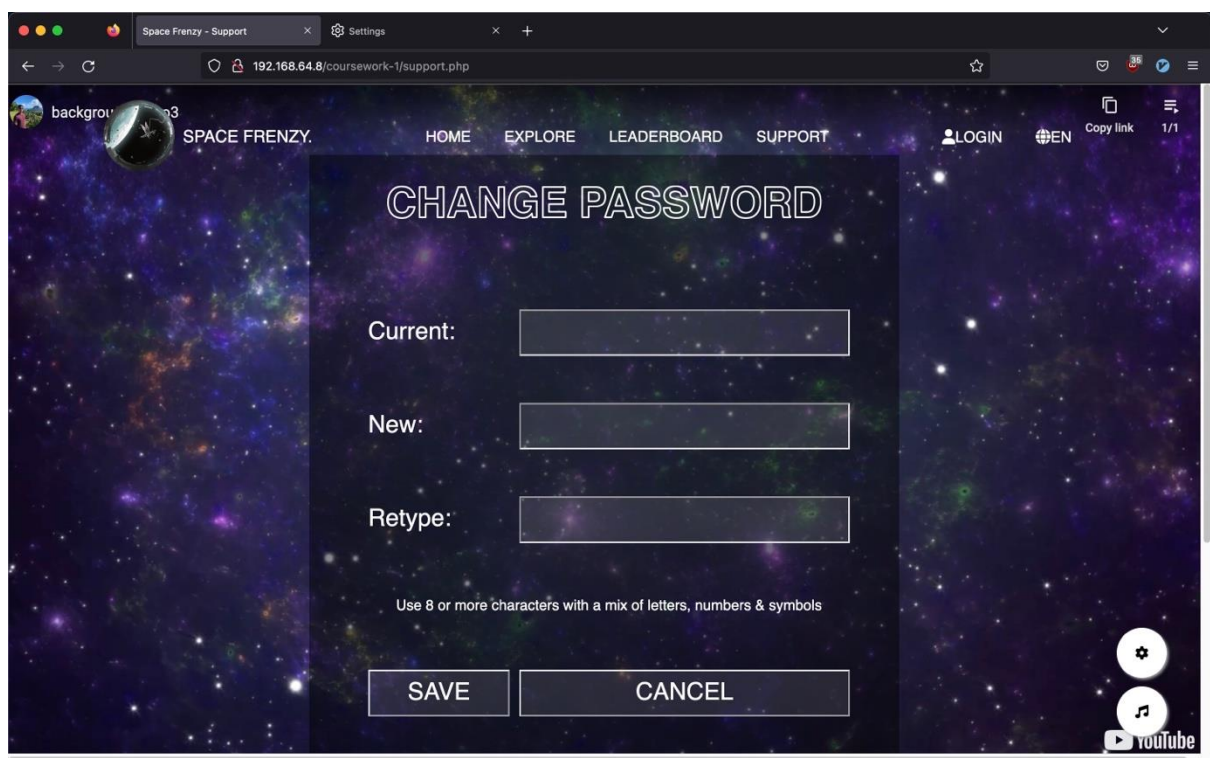


Figure 15 Support Page

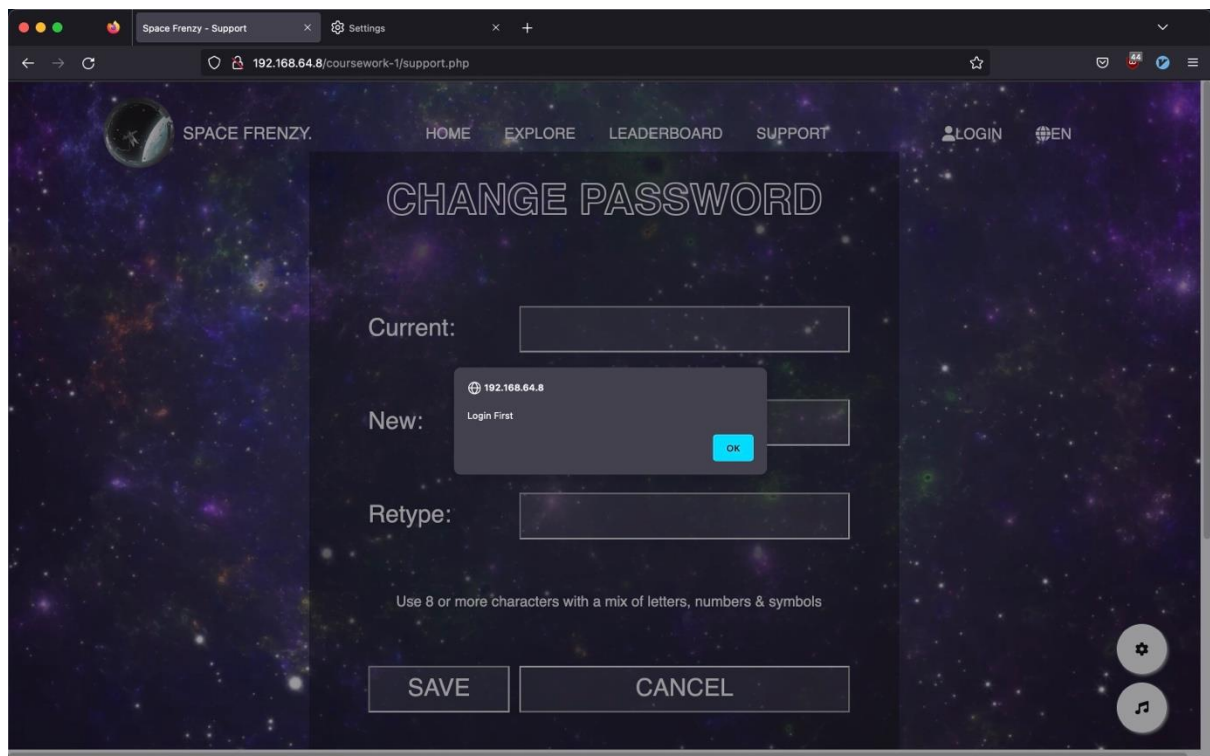


Figure 16 Support Page Validation Error Message

Conclusion

Implementation of javascript using technologies like classes, inheritance, canvas, local storage and session storage was really amazing. And hope that I will learn more in the future.