PROJECT REPORT

Hamza Bari

Contents

Introduction 2
Screenshots of all webpages 3
Home Page 3
Registration Page 4
Login Page 5
Game Page 6
Score list page 7
How the game website works 8
Registration webpage 8
User successfully registers their account 8
User fails register their account 9
Login webpage 11
User successfully logins 11
User fails to login 12
Game webpage 14
How the game is played 14
Score List webpage 18
Storage of user scores in Local Storage 18
User scores loaded & displayed on the score list webpage 19
The sources used 20
Source 1 20
Source 2 20
Source 3 20
Challenges I found in duration of the project 21

Introduction

In this report I'm going to talk about the game website I have developed. This includes the following: how the user needs to register, how the user needs to login, how the user needs to play the game, and how the user can check their individual score in the scores list webpage, how the login handles the error messages and how the registration form uses input validation to check if all the user information is entered correctly or not.

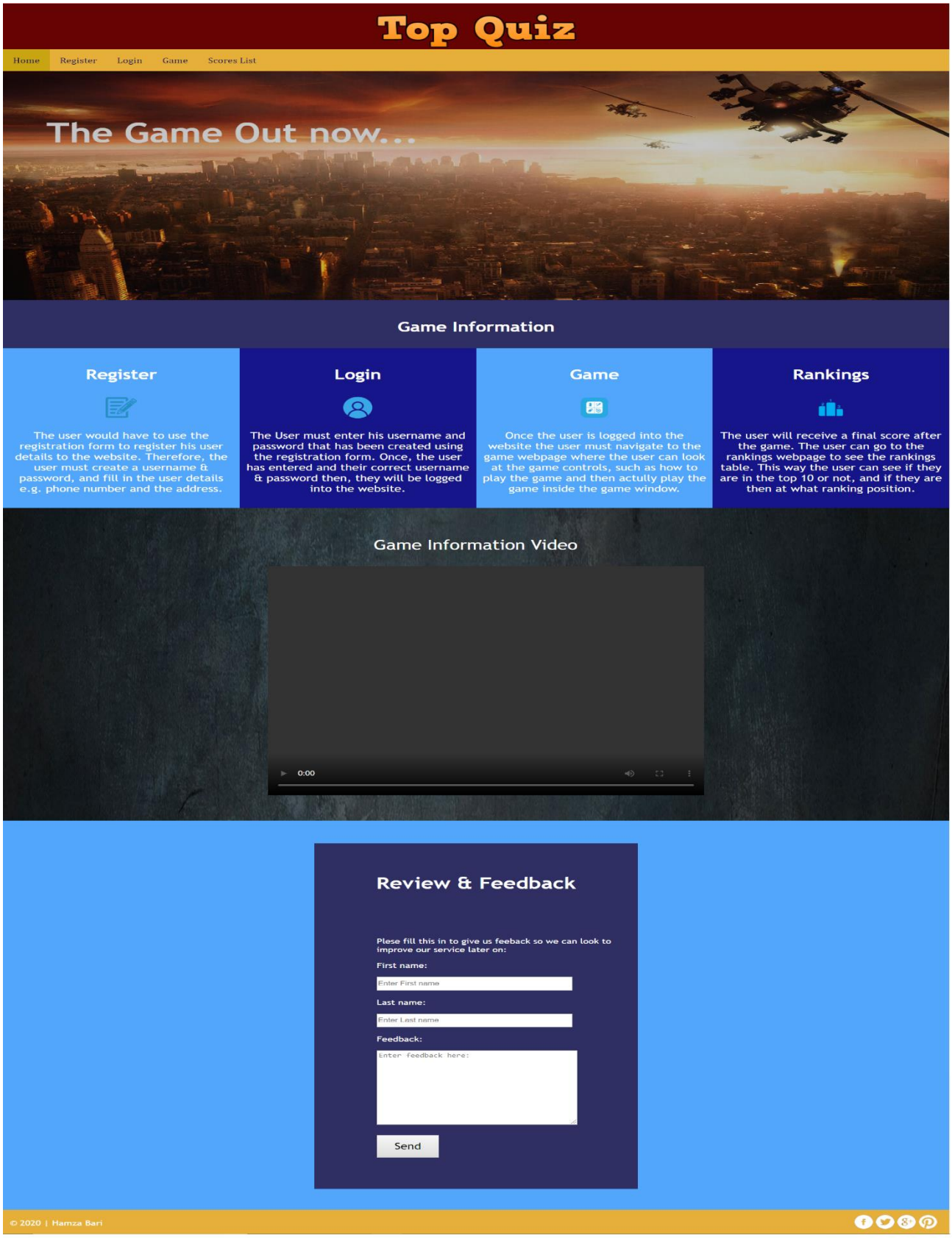
This project has been implemented using: HTML, CSS, PHP and JavaScript. I used HTML to add the contents to my webpage e.g. navigation bar, footer bar, forms etc. I used CSS to style those contents and added colours and changed the font types. I used PHP to eliminate duplicates which was the navigation bar, footer bar, webpage header with the title, html tags at the top, and html tags at the bottom. After that, I used the PHP functions to display those contents dynamically on every webpage. I used JavaScript to create the following functionalities: user accounts including registration and login, game implementation the quiz game, and finally displaying the scores in the scores list page and in the local storage.

I have used the Google Chrome web browser during my implementation. I used Local Storage to save all the user data and retrieve it onto the website. I have used session storage to store the user who will currently be logged in the website.

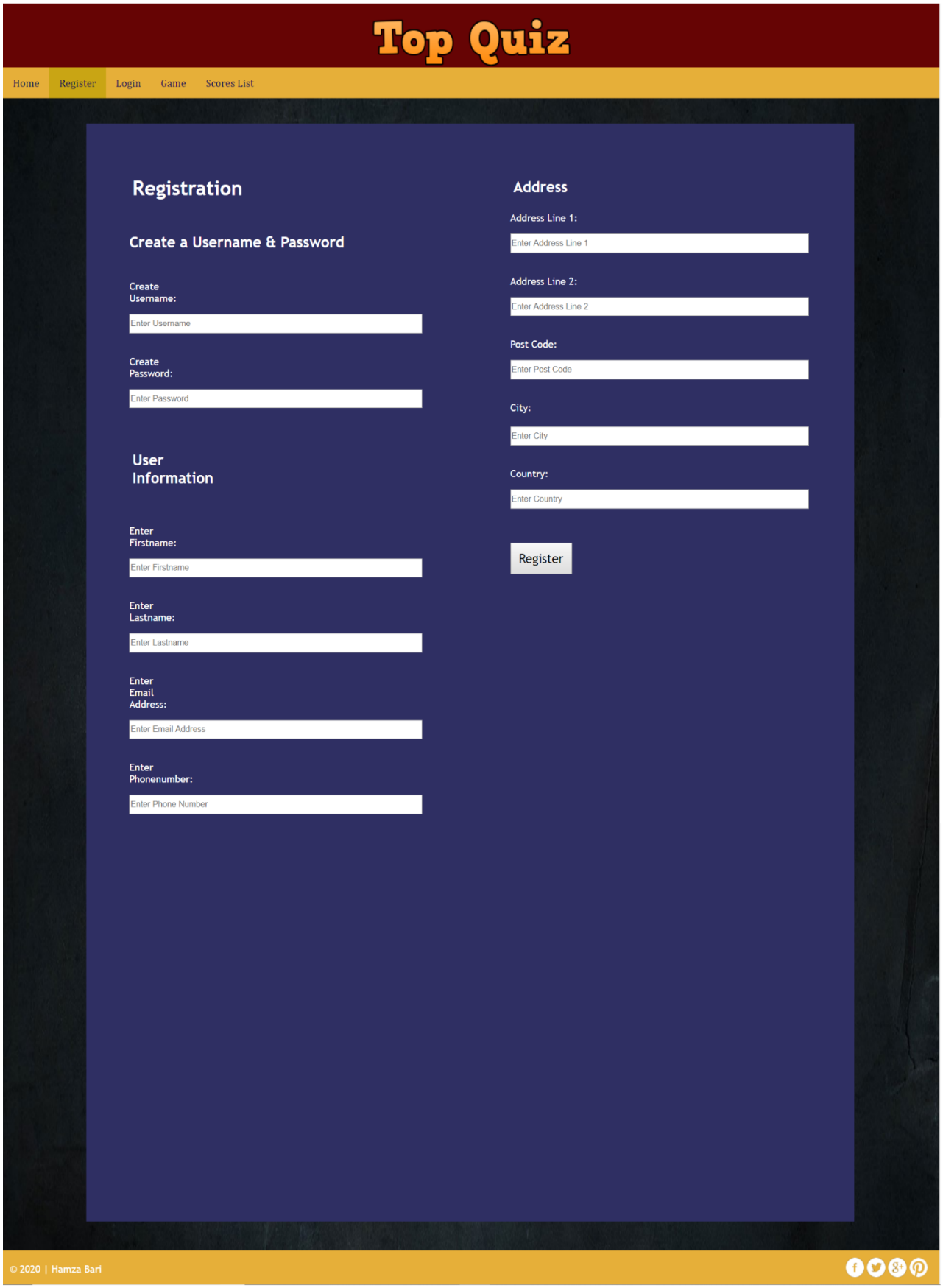
I have also provided full screenshots of all the webpages that are included in my website. You can see that in the section below.

Screenshots of all webpages

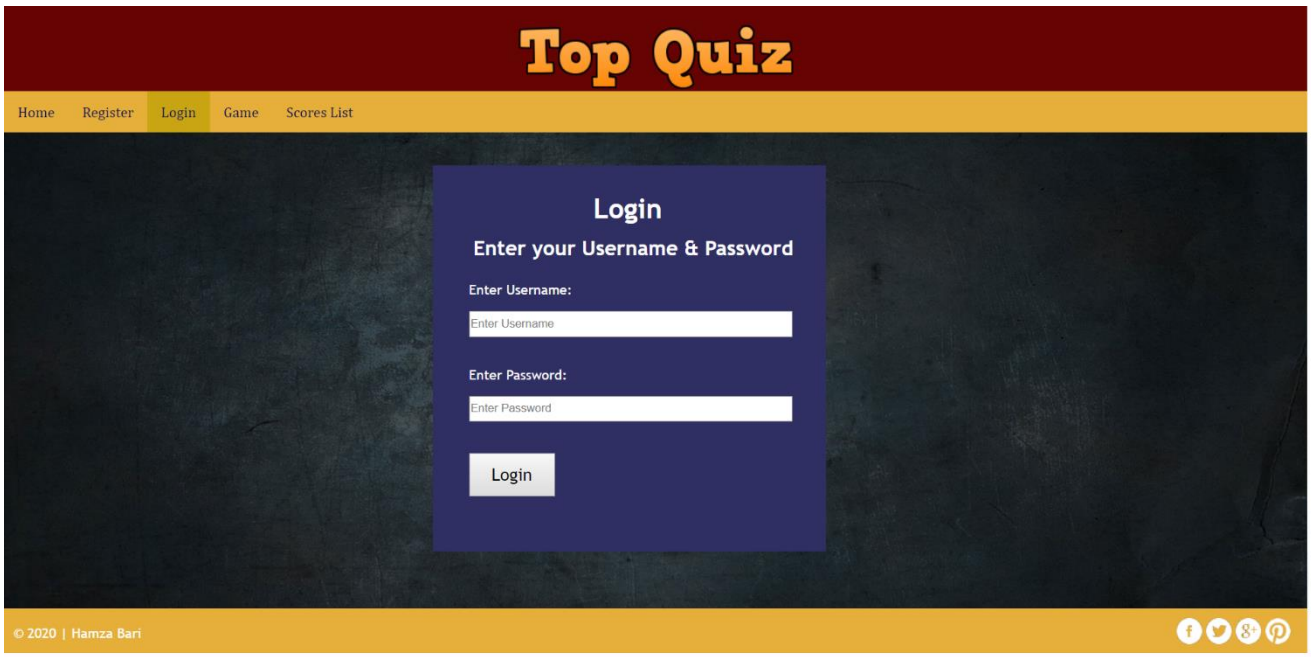
Home Page



Registration Page



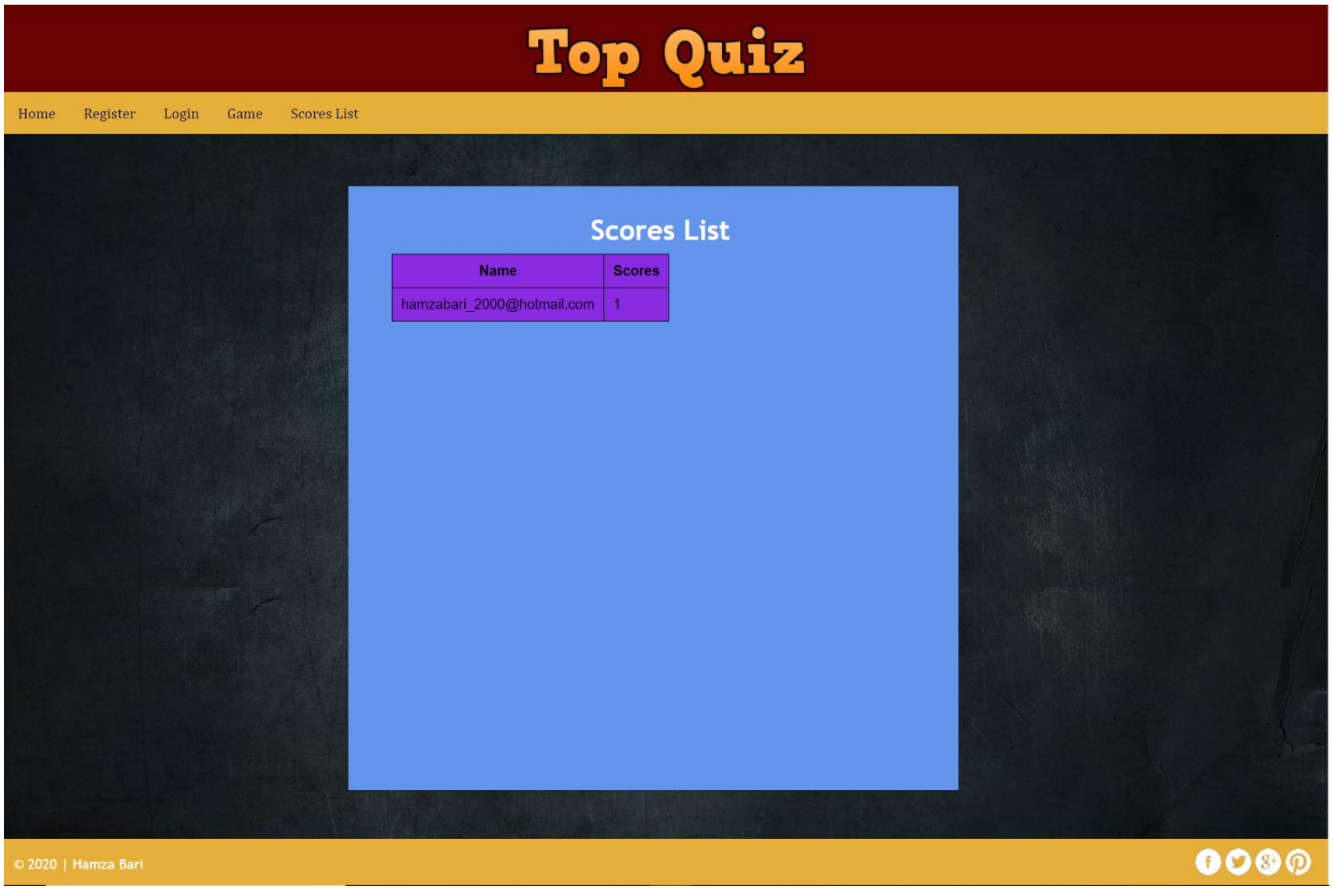
Login Page


Top Quiz
Home
Register
Login
Game
Scores List
Login
Enter your Username & Password
Enter Username:
Enter Username
Enter Password:
Enter Password
Login
© 2020 | Hamza Bari

Game Page


Top Quiz
Home
Register
Login
Game
Scores List
How to play the game?
Follow these instructions below to see how to play the game. Use game tips given below:
1.) Click on a textbox to enter your answers.
2.) Look at the given images provided for some questions.
3.) Try spending 1 minute per question.
4.) To restart refresh your page.
You will recive a final score once the game ends.
Scores:
Game - Top Quiz
Q1: What is the month after November?
Q2: What is 11 x 15 equal to?
Q3: What is the capital city of USA?
Q4: How many days does Febuary have in a leap year?
Q5: What is half of 50?
Q6: What is the double of the number 8?
Q7: What is the number of total days in a year?
Q8: Looking at this picture below, how many letters does in contain?
What is your name?
Q9: Looking at this picture below, What is it?
Q10: How many dots does this picture contain below?
Submit
© 2020 | Hamza Bari

Score list page



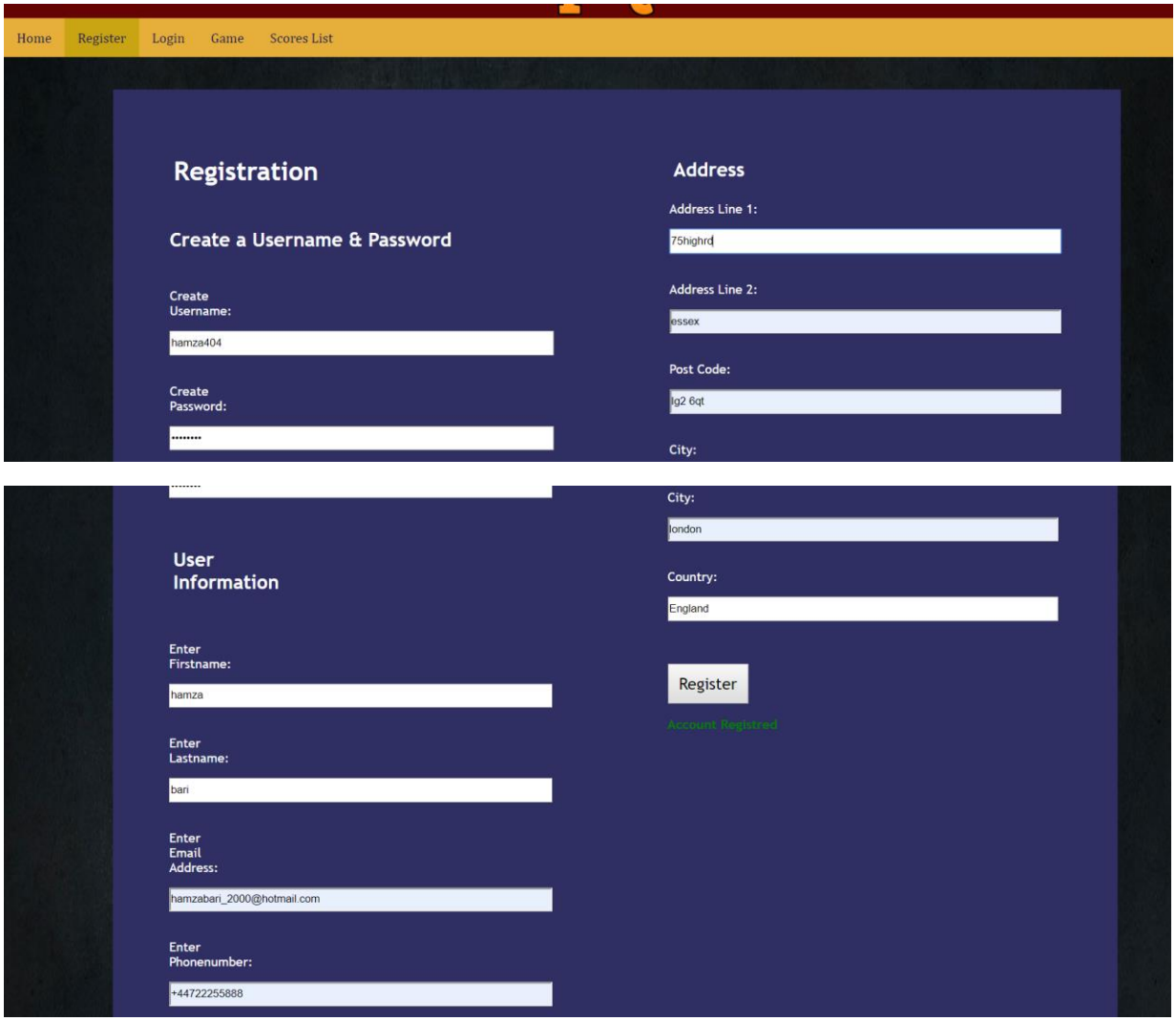
How the game website works

In this section I'm going to show how the user can use the website to register, login, play the game, and see their scores on the score list webpage. Below, I am going to explain all of this is detail.

Registration webpage

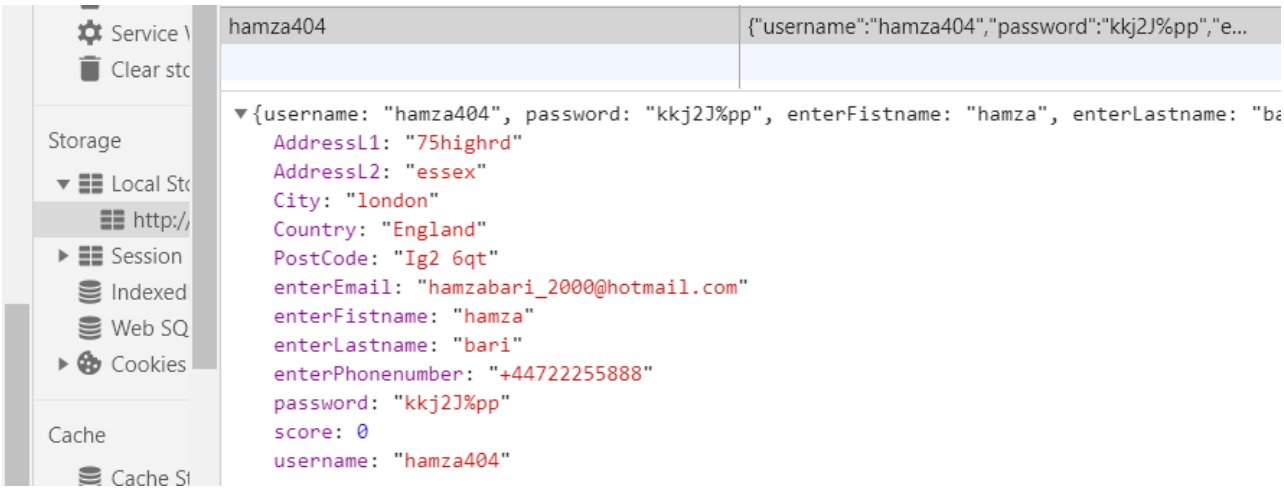
This is where I'm am going to show the following things: the way the user successfully registers to the game website, and the user information gets stored onto the local storage. If the user fails to register an account, then they will receive error messages to inform them which input they have entered incorrectly.

User successfully registers their account



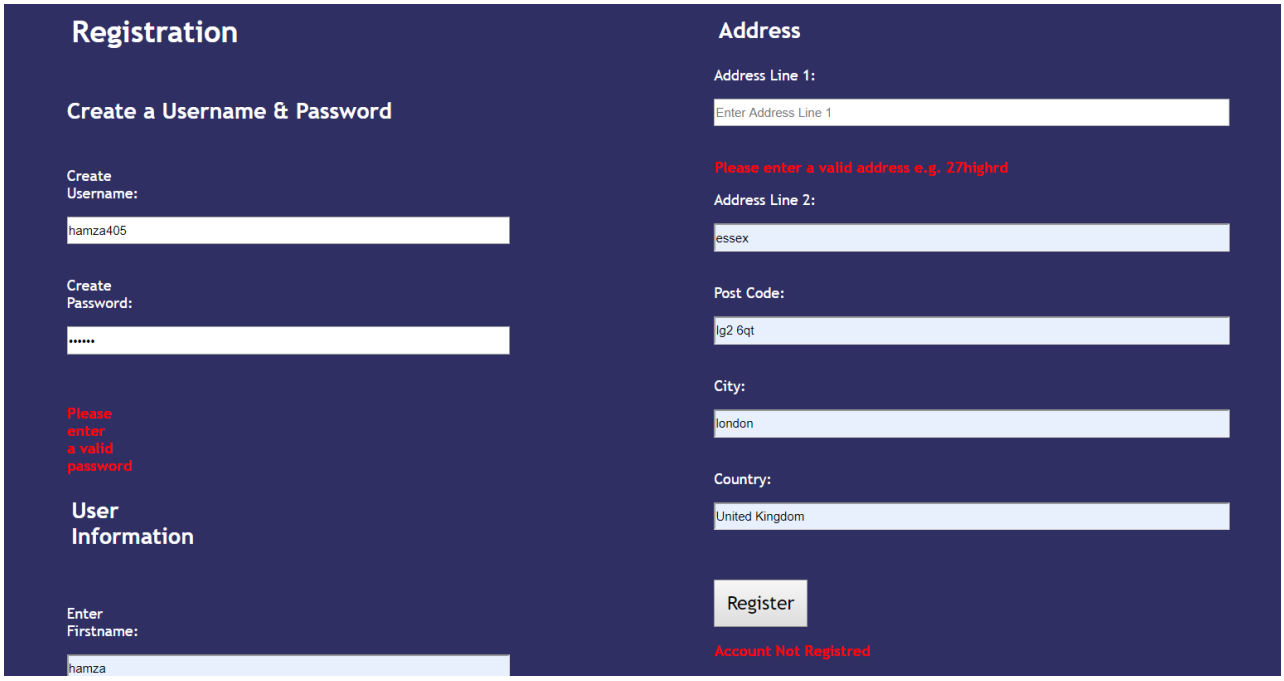
This is where the user is on the registration page to register his account onto the website. The user must input all the information into the input text fields according to the correct information. After that, the user needs to click onto the submit button where he will receive a message at the bottom.

This message is to inform the user that their account has been successfully been stored onto the website.

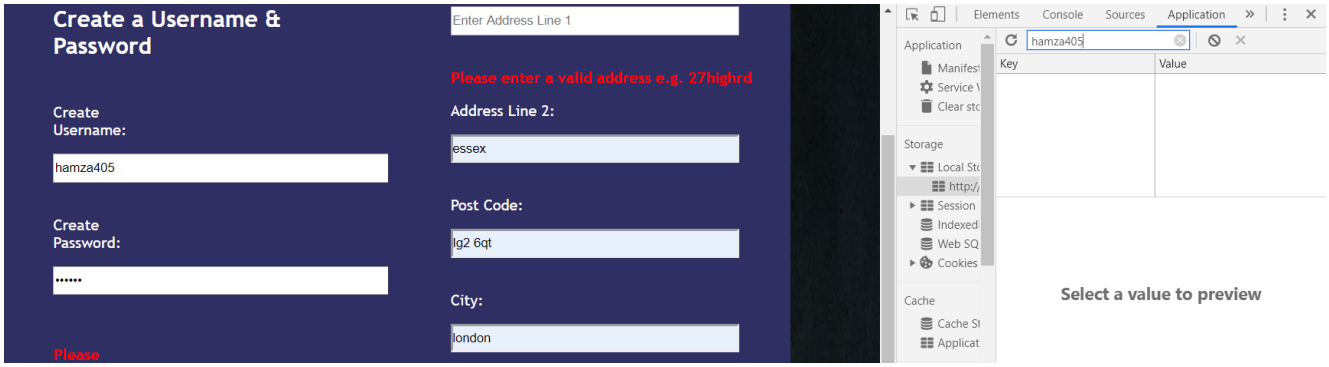


If the user registration is successful and their account is registered then, all their user information is stored onto the local storage in the JSON file format, as you can see that from the screenshot above.

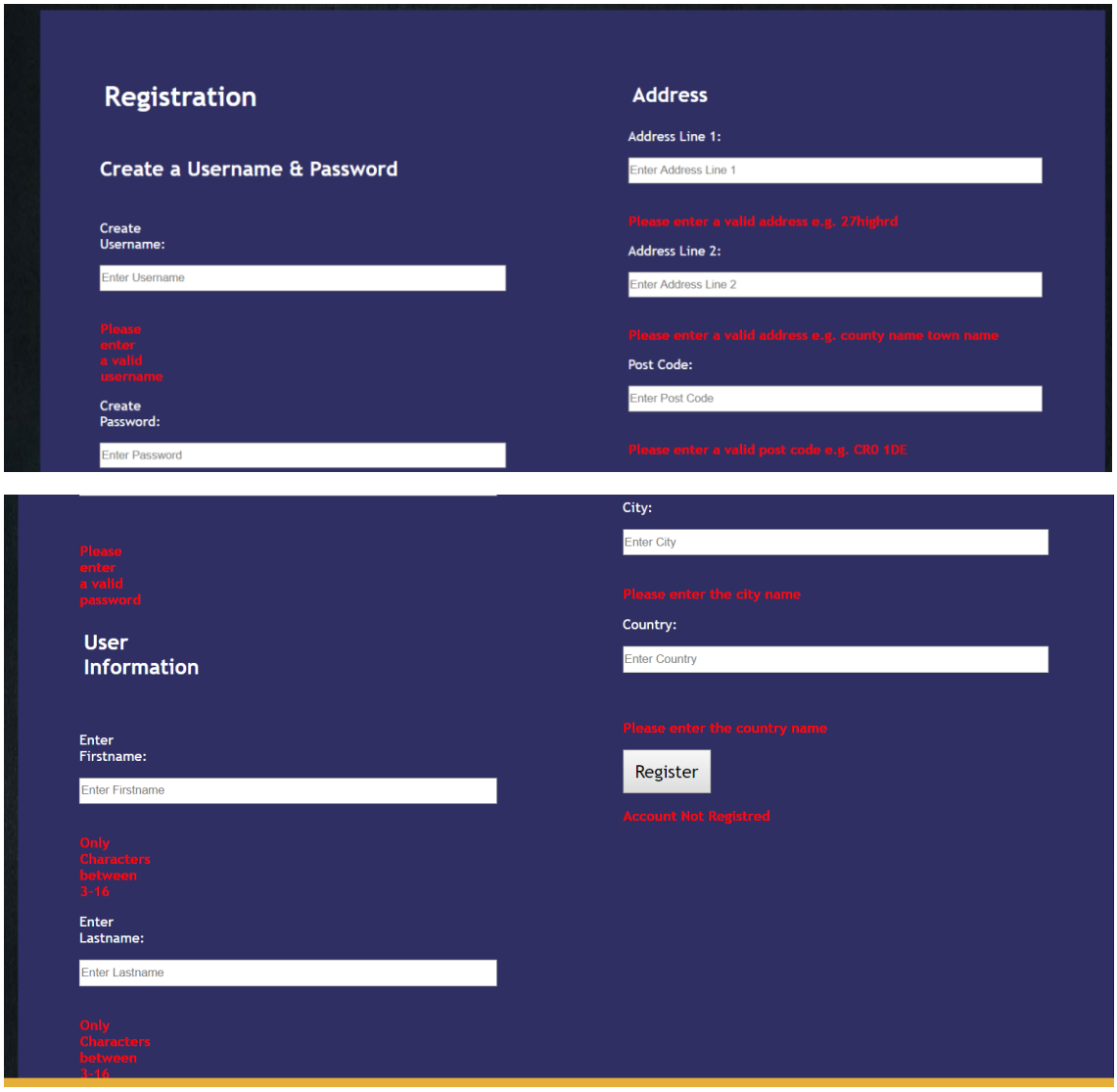
User fails register their account



While the user is registering their account, they might leave a text field empty, or they might provide an invalid information which doesn't match the user validation. Then their account will not be registered onto the website as a result, there will be an error message to inform the user that their account has not been registered. In addition to that, wherever the user has left an empty text field or entered an invalid information which does not match the validation then, underneath that text field the user will receive an error message.



As, a result of user failing to register their account, you can see that their information isn't getting saved onto the local storage. You can compare this by looking at the screenshot when the username isn't appearing up on the local storage.



If the user just tries to submit an empty form which has no information entered then, underneath all the input text fields there will be an error message. There will also be an error message to inform the user that their account hasn't been registered to the website.

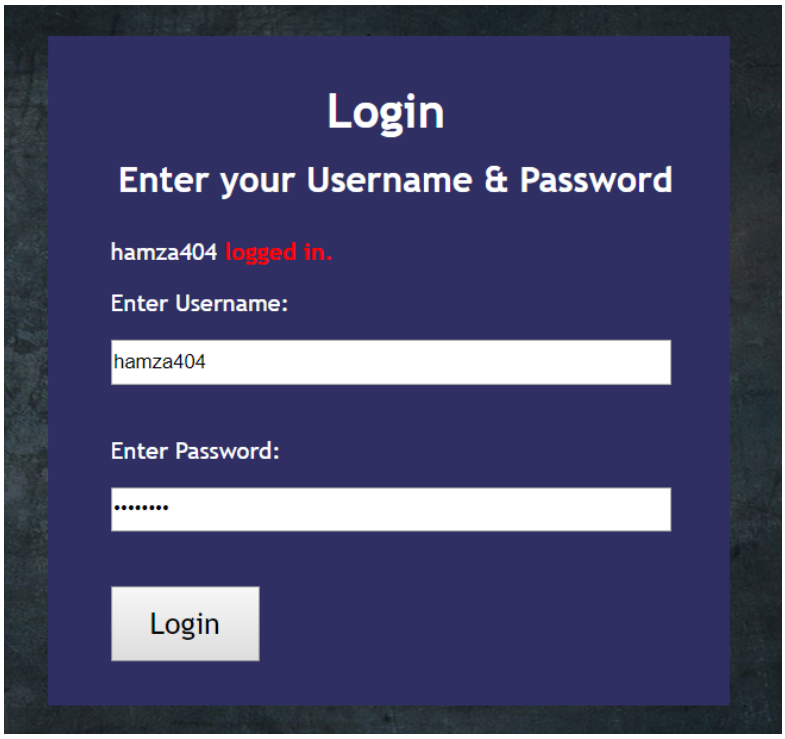
Login webpage

This is where I'm going to show how the login functionality works. Once, the user has registered their account onto the website. After that, the users would have to use their username and password to login to the website from their registration information. If the user successfully enters their username and password, then they receive a message with their username saying, "logged in".

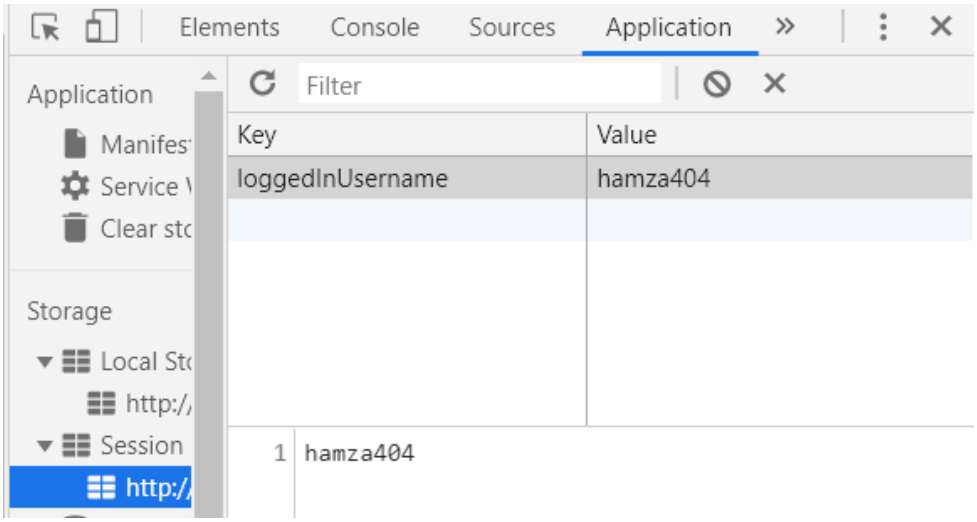
If the user enters an incorrect password then, they will receive an error message, which alerts the user that they entered an invalid password.

If a user tries entering when they aren't registered then, they receive an error message which again alerts the user that they aren't registered to this website therefore, they need to register and in addition to that, if the user enters an incorrect username then the same error message will appear.

User successfully logins



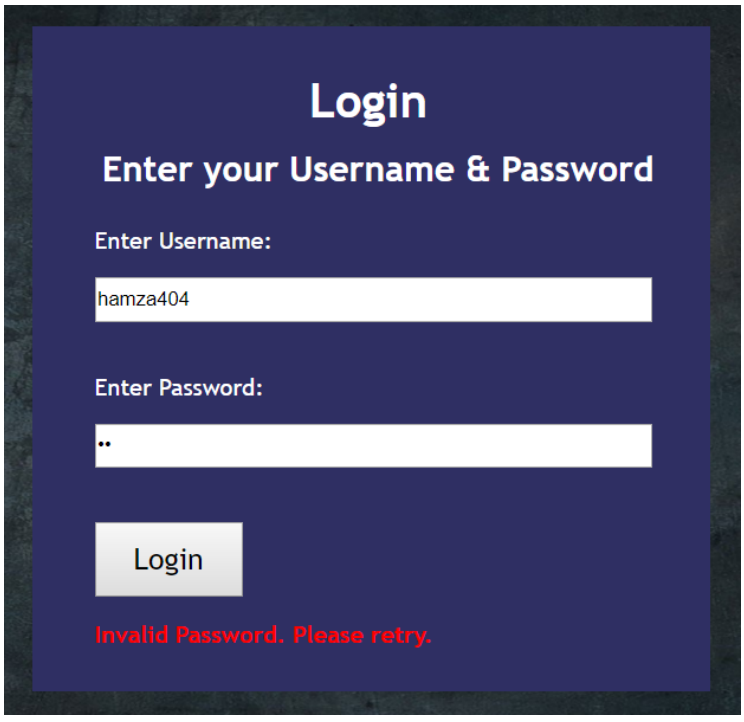
After the user has registered then the user must use his correct username and enter it in the username text field and enter the correct password into the password text field. After that, the user must click on the login button to login to the website. As result, on top of the username label it will display as message saying "username logged in" which, makes the user realise that they have successfully been logged into the website.



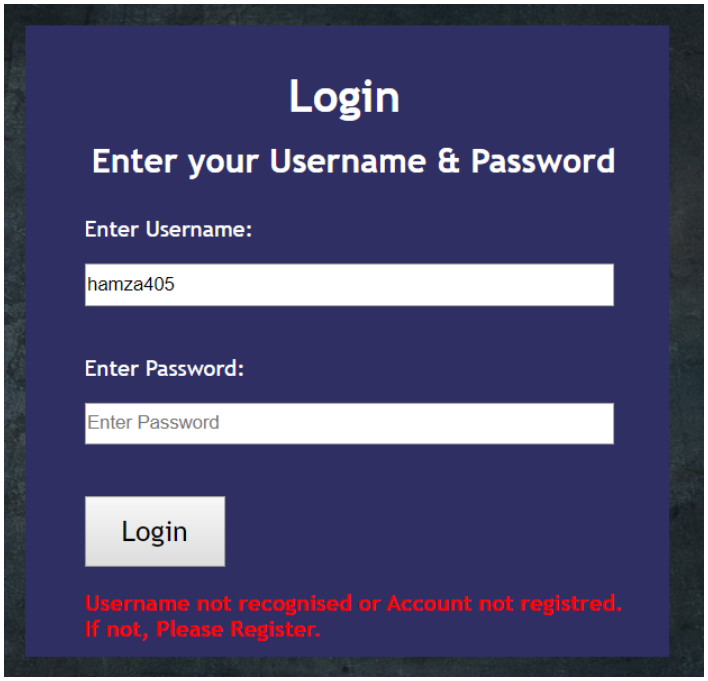
When the user successfully login to the website then, in the session storage it will show the username value which, is going to be the username. That username is going to be next to the **loggedInUsername key**. All the logged in users are getting saved onto the session storage, if they have successfully logged in to the website.

The reason why Session Storage was used for the login is because, it will disappear once the user closes the website. If the user opens their website another time then, they would have to login to the website again which makes more sense. If the local storage was getting used then, the user would stay signed in, even if they close the website and open it later which wasn't making sense. That's why the session storage was used for the login of the user.

User fails to login



If the users account is registered and while they are on the login page they might of entered the correct username, but they entered an invalid password therefore, underneath the login button the user will receive an error message saying that their password is not correct and they need to re-enter their password to login to the website again. This message will keep on appearing, until the user enters their password correctly.



If the user enters their wrong username or if a user tries to login when they aren't registered to the website, then they will receive an error message saying that their username is not recognised, or this account isn't registered.

Game webpage

In this section I'm going to explain about the following: how the game is played, what the game is about, and how the scores are displayed on the scores list webpage, and how the score is updated inside the local storage.

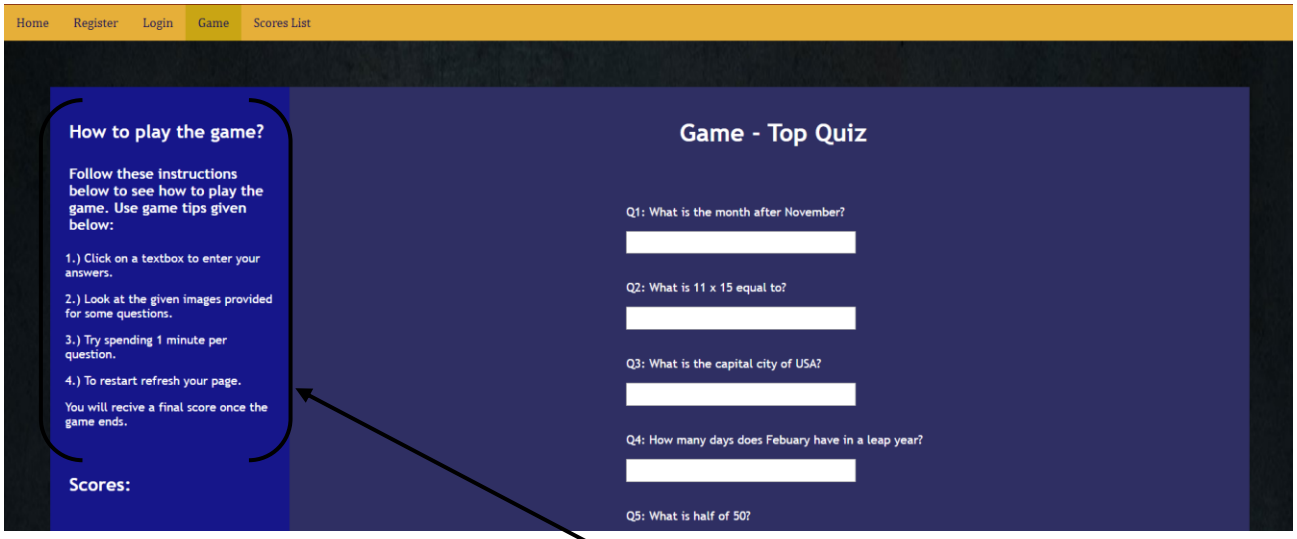
How the game is played

The way the user must play the game is, first register and login, or if the user is already registered then they just have to login. Then the user must navigate onto the game webpage after that, there are two columns, on the left column there are a list of all the game instructions which the user must follow to play the game. After that, on the right-hand side column is where the user would play the game. The game is called **"Top-Quiz"** and **it's a quiz game which has ten questions about general knowledge**.

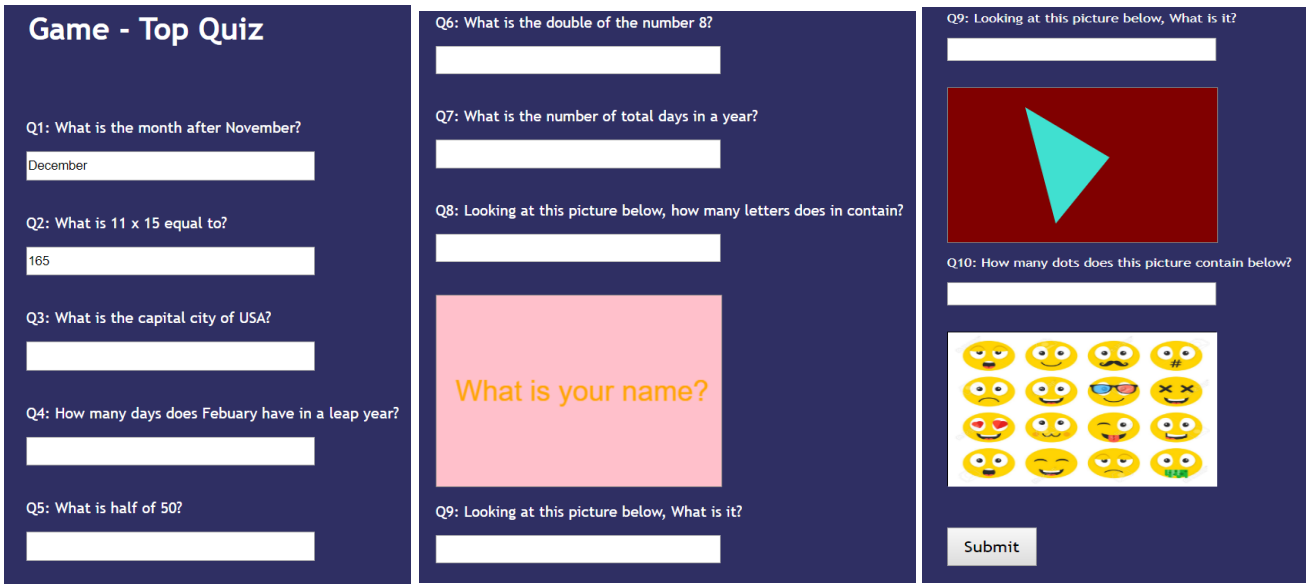
There is one point per one question correct. There will be a question and underneath the question there will be an answer-box, inside the answer-box **(answer-box is a text-field)** is where the user would need to enter his answers. The last three questions are related to the canvas drawings and an image, which are underneath the answer-box therefore, for those last three questions the user would have to look at those canvas drawings and the image to answer those last three questions successfully.

Once, the user has inputted all his answers in the answer-box then, the user needs to click on the submit button, which is below all the questions. Once, the user has pressed the submit button then, underneath each of the correct questions answered it will show a green coloured message which will say **"correct answer"**, and the questions which have been left blank or incorrectly answered will receive a red message saying **"Wrong answer"**. Next to the red message it will display a white coloured sentence which, shows what the correct answer should have been for example, this is what it will show **"wrong answer the answer should have been 29"**.

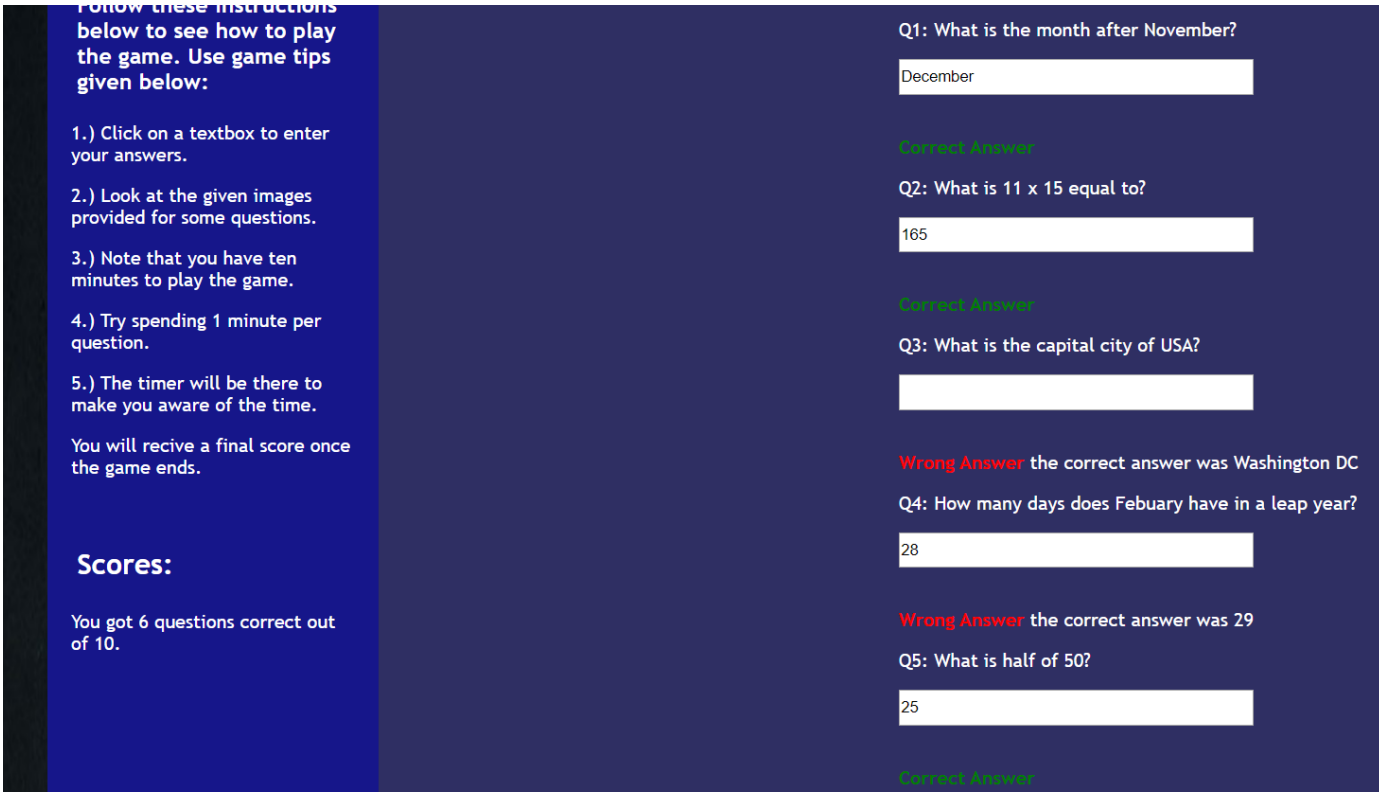
After that, the user should look back at the left-hand column. Underneath the game instructions there is a score label, and underneath the score label is where the user score is going to be listed in a sentence e.g. **"You got 1 question correct out of 10"**. Also, this score will be updated inside the local storage. The player can refresh the game webpage and keep on having as many go's as they want. Each time they play, the game score will be getting updated.

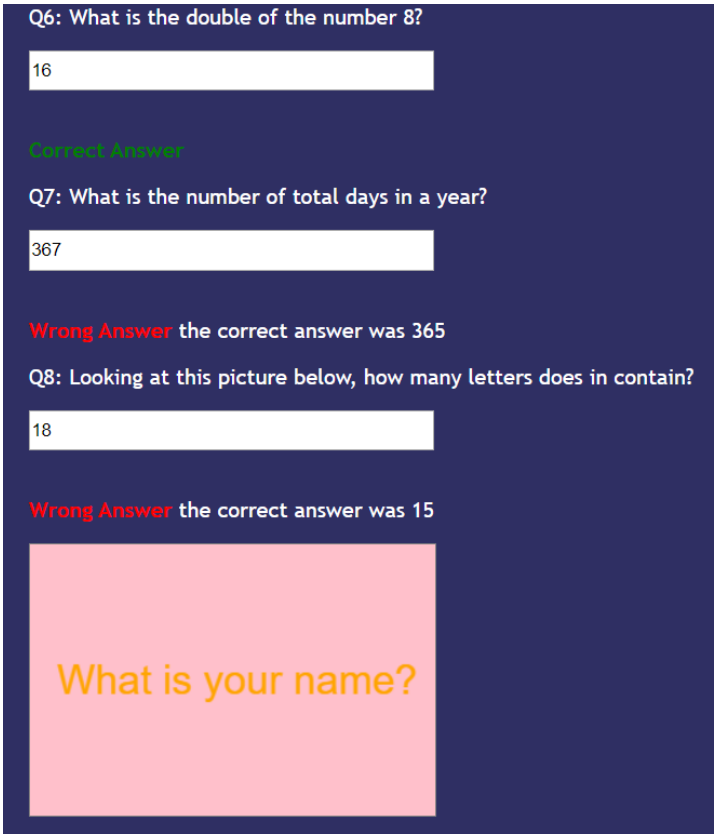


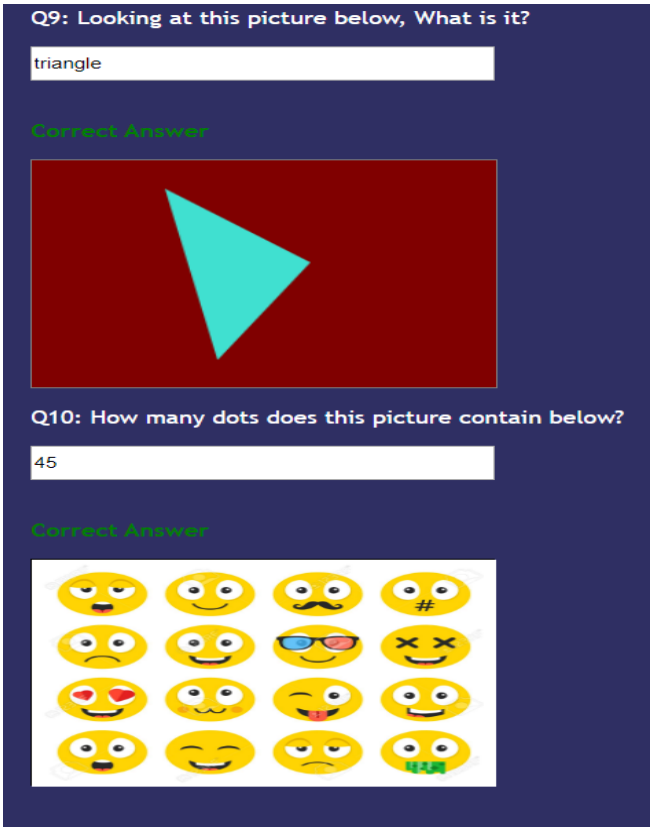
On the game webpage left-hand side of the column, there are a list of instructions which would help the user on how to play the game. it's a sort of a help guide for the users, to play the game.



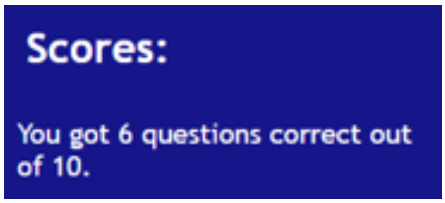
On the right-hand side of the column is where the user would play the quiz game. The way this game works is that there are ten questions in total. The user will get one point per one question correctly answered. For question eight and nine there is a canvas drawing which is related to the question, as the user must use it for answering those two questions. Also, there is a picture for question ten which, again the user must use for answering question number ten. Below each question there is a text field which, the user must use for entering in their answers. Once, they have entered their answers then, the user needs to click on the button below submit which, will complete the game.


below to see how to play the game. Use game tips given below:
1.) Click on a textbox to enter your answers.
2.) Look at the given images provided for some questions.
3.) Note that you have ten minutes to play the game.
4.) Try spending 1 minute per question.
5.) The timer will be there to make you aware of the time.
You will recive a final score once the game ends.
Scores:
You got 6 questions correct out of 10.
Q1: What is the month after November?
December
Correct Answer
Q2: What is 11 x 15 equal to?
165
Correct Answer
Q3: What is the capital city of USA?
Wrong Answer the correct answer was Washington DC
Q4: How many days does Febuary have in a leap year?
28
Wrong Answer the correct answer was 29
Q5: What is half of 50?
25
Correct Answer

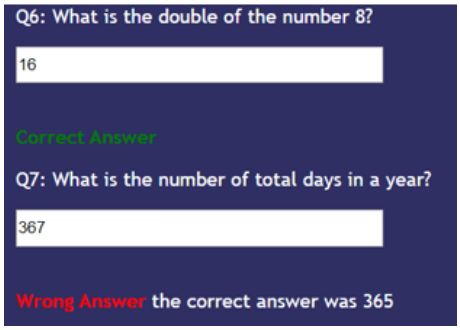

Q6: What is the double of the number 8?
16
Correct Answer
Q7: What is the number of total days in a year?
367
Wrong Answer the correct answer was 365
Q8: Looking at this picture below, how many letters does in contain?
18
Wrong Answer the correct answer was 15
What is your name?

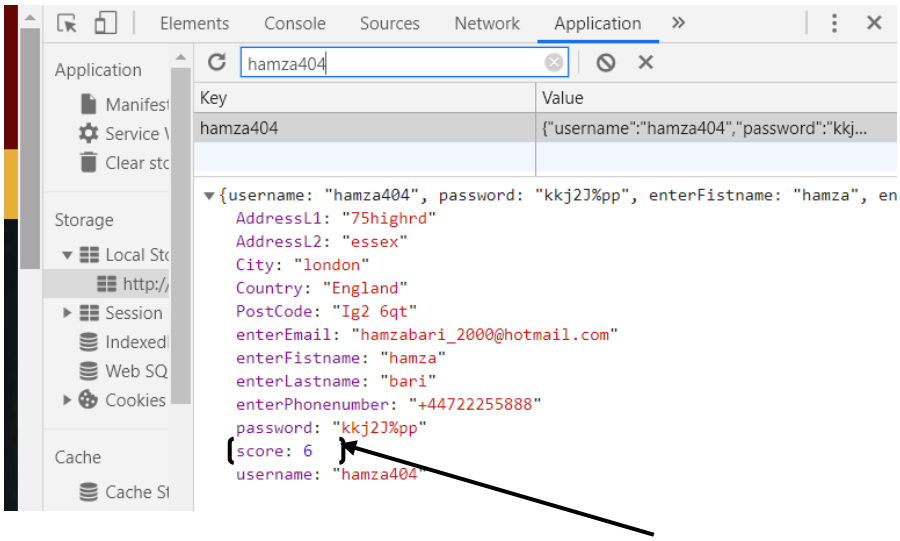

Q9: Looking at this picture below, What is it?
triangle
Correct Answer
Q10: How many dots does this picture contain below?
45
Correct Answer

As, the user completes his game by pressing the submit button then, on the left-hand side of the column underneath the game guidance instruction there is a score label, and underneath the score label is where it displays a message saying the total number of score which, the user has achieved after playing the game. You can see this in the screenshot below.



In addition to that, underneath each of the question text field the user will receive a feedback message for example, In the screenshot above you can see wherever the user got the question correct. After that, underneath that text field it will display a message saying **"correct answer"** in a green colour which shows the users what answers they got correct. If the user gets the question incorrect then a message is displayed underneath the text field saying, **"wrong answer"** and in addition to that there will be a message next to that which, will be displaying the correct answer. You can see the example of this in the screenshot below.





Once the user has played the game then, the user will receive a score, and in addition to that inside the local storage the user score gets updated. When the user account is created the score is set to zero as, the user plays the game then, their scores get updated in the local storage. Also, if the user keeps on playing the game, then the user score will continue to get updated inside the local storage.

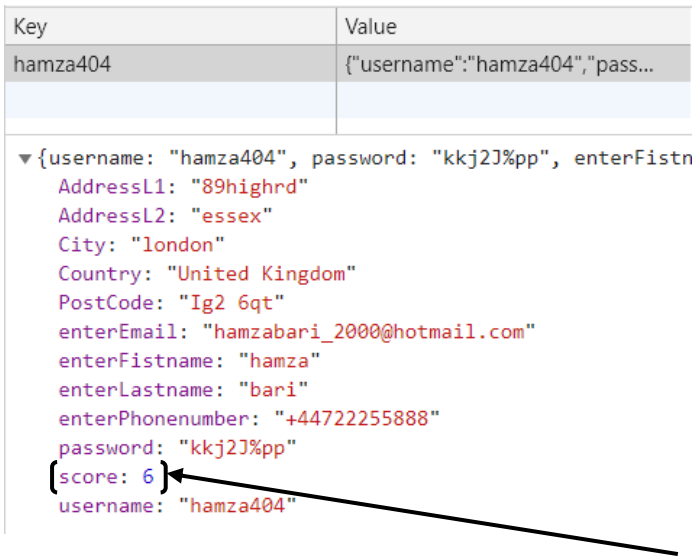
Score List webpage

In this section I'm going to explain how the scores are saved and updated inside the local storage and, in a JSON format. I will talk about how the scores are loaded up on the scores list webpage where, it will display the user email with the user score within a table.

Storage of user scores in Local Storage

Once, the user has completed playing the game then, the user would press the submit button. After that, on the submit button click, the users current score will get updated inside the local storage in JSON format.

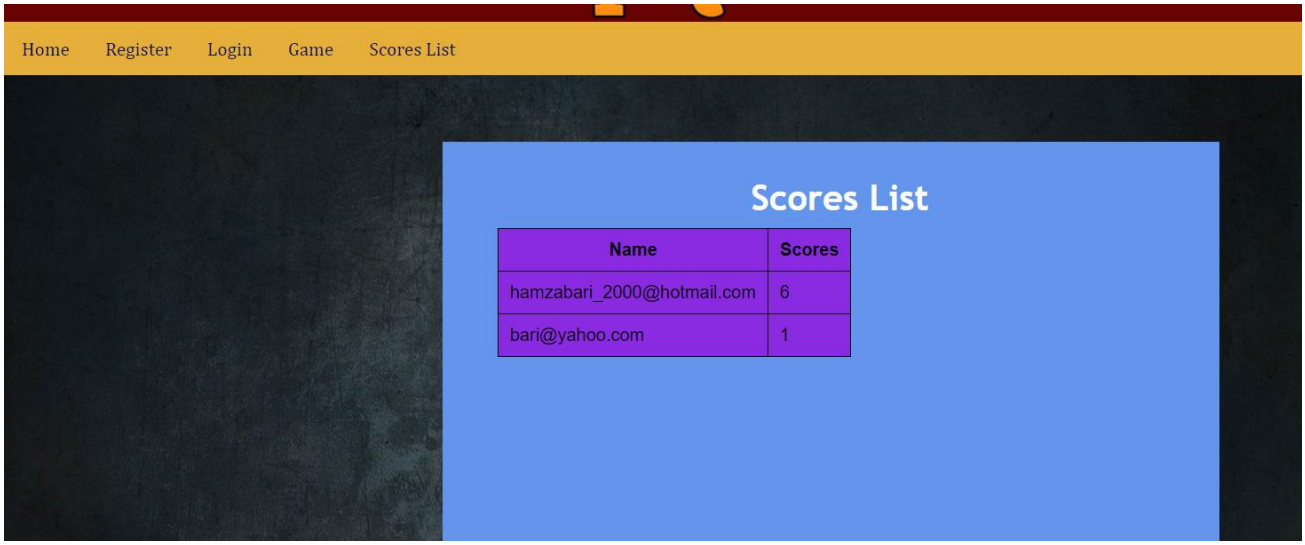
When the user registers on the website for the very first time then, their score is set to 0.



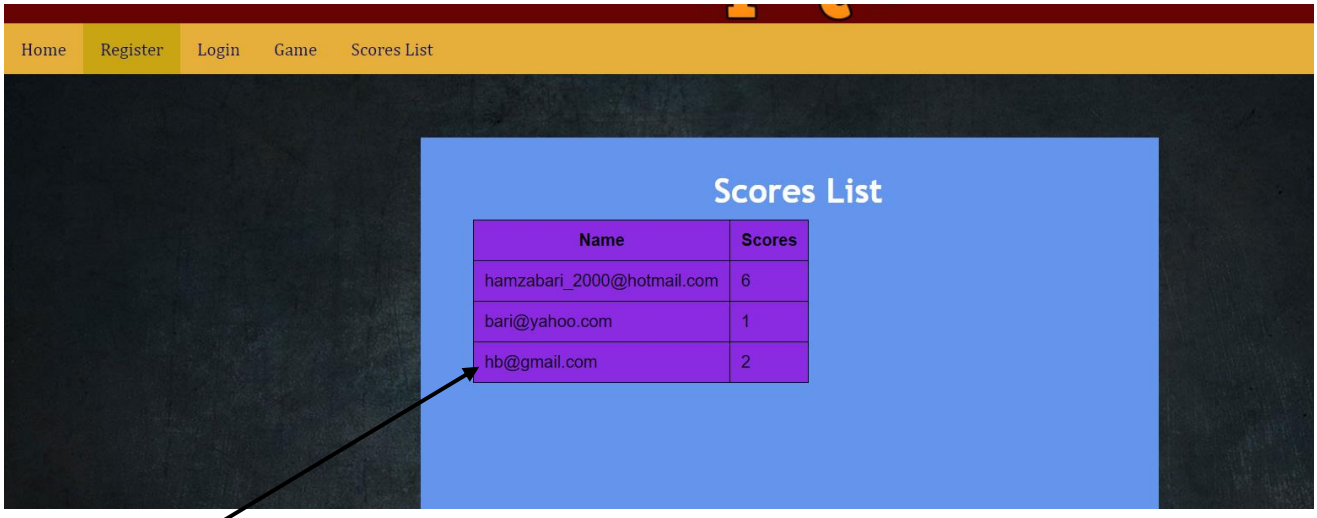
Once the user plays the game and clicks on the submit button then, you can see from the screenshot above that the user score is now updated. In addition to that, every time the user plays the game the score will be updated, if there are any changes in the score. If the user keeps on playing and the score they are getting is equal each time then, their score will remain the same on local storage.

User scores loaded & displayed on the score list webpage

When the game is played and the submit button is pressed then, all the scores are saved in JSON format inside the local storage. In addition to that, the user email and the user score is going to be loaded and displayed inside the scores list webpage, inside the table. This is where the users can navigate to the scores list webpage for checking their scores after they play the game.



Once the user has played their game then, this is where they can check their score. This table will remain even if the user is logged out. The users can always navigate to this webpage and check their individual scores.



In the screenshot above, you can see that a new user is added. So, each time a new user login and plays the game then, the table will add their email and score in a new row below the previous one.

The data going inside the scores list table is being loaded from the local storage, and it's displaying the data in the table.

The sources used

This is where I'm going to explain about the sources, I used for developing this project. This includes which links I used from the Internet for getting ideas about how to implement the project, and in addition to that, what IDE and software's I used while I was developing this game website project.

Source 1

This is the link I used https://www.w3schools.com/ during the development of my project.

I used this website to research about, how to create a CSS box model and add content inside it which includes, adding spaces, moving the content in the correct position. This helped me when I added the contents of the registration, login, game and the scores list inside a box. As, you notice that all my content is inside boxes on the webpages.

On this website I looked at examples of how to use the html columns therefore, this helped me create two columns on my register and game form. As, you will notice that both contents are next to each other.

I also used this to learn a bit about JavaScript especially on how to use the arrays, as this was useful when I had to create an array to push the user email and score inside the array. This helped me load and display all the user email and score on the scores list webpage as, I used that array with a for loop to display all the users email and scores on the scores list webpage.

Apart from this website there were no other third-party libraries, or other websites used.

Source 2

I developed my game website project on the Visual Studio Code IDE.

Source 3

I used the XAMPP Server to run PHP on the webpages.

Challenges I found in duration of the project

In this section of the report I'm going to explain, what I found difficult and what is not working in this project e.g. the problems I had.

The things I found difficult when I was developing my game project was displaying the user score in the scores list webpage inside the table. Therefore, I created a separate object in JavaScript which only has the user email and the current score which was called score object. I managed to display that onto the scores list webpage but, the problem was that it was only displaying it for the user who was logged in, as it wasn't displaying for all the other users who were registered. Therefore, I researched online about using arrays, and using for loops with an array to display things more than once.

So, once I learnt that concept, I created an array, and what that array does is once the game is played, and the submit button is pressed then, it adds the score object into the array. I make that array into a JSON format and then, using a for loop I display all the users in the array onto the scores list webpage. After that, it started to work but, **there is only one problem with it and that is when the user refreshes the webpage and plays the game again, then in the scores list webpage instead of the score getting updated inside the table, it adds a new row with the email and the score on the table.** This is one of the limitations I have in this project.

This was the task that I found quite challenging, when I was developing my project.