

A Report On
“GAMING WEBSITE”

**Submitted to the
Department of Computer Applications**

In Partial Fulfilment of the course

Master of Computer Applications

Under the guidance of

Ms.TINTU P B

BY

**ATHUL K KUMAR
(REG NO: SGI18MCA-I020)**



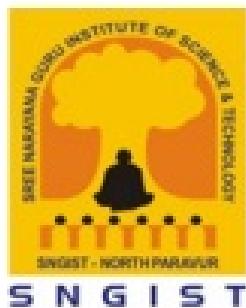
DEPARTMENT OF COMPUTER APPLICATION

**SNGIST GROUP OF INSTITUTIONS
North Paravur - 683520**

2018-2023

SNGIST GROUP OF INSTITUTIONS

North Paravur-683520



BONAFIDE CERTIFICATE

Certified that the Project Work entitled

”GAMING WEBSITE”

is a bonafide work done by

ATHUL K KUMAR

*In partial fulfillment of the requirement for the
Award of*

MASTER OF COMPUTER APPLICATIONS

Degree from

APJ Abdul Kalam Technological University, Thiruvananthapuram

(2018-2023)

Prof. KAVITHA C R

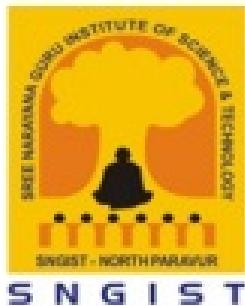
Ms.TINTU P B

HEAD OF DEPARTMENT

INTERNAL GUIDE

SNGIST GROUP OF INSTITUTIONS

North Paravur-683520



CERTIFICATE

This is to certify that the project entitled "**GAMING WEBSITE**" has been successfully carried out by **ATHUL K KUMAR** (Reg no: SGI18MCA-I020) in partial fulfillment of the Course **Master of Computer Applications**.

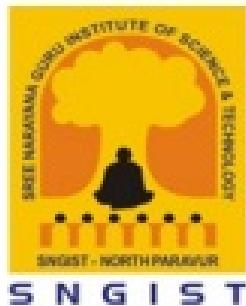
Date:26-October-2022

Prof. KAVITHA C R

HEAD OF DEPARTMENT

SNGIST GROUP OF INSTITUTIONS

North Paravur-683520



CERTIFICATE

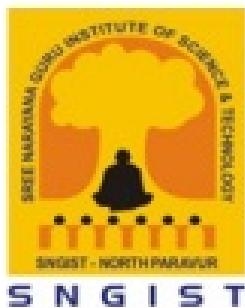
This is to certify that the project entitled “GAMING WEBSITE” has been successfully carried out by ATHUL K KUMAR (Reg no: SGI18MCA-I020) in partial fulfillment of the course Master of Computer Applications under my guidance.

Date:26-October-2022

**Ms.TINTU P B
INTERNAL GUIDE**

SNGIST GROUP OF INSTITUTIONS

North Paravur-683520



DECLARATION

I, ATHUL K KUMAR, hereby declare that the project work entitled “GAMING WEBSITE” is an authenticated work carried out by me under the guidance of Ms. TINTU P B for the partial fulfillment of the course **MASTER OF COMPUTER APPLICATIONS**. This work has not been submitted for similar purpose anywhere else except to **SNGIST GROUP OF INSTITUTIONS, North Paravur**, affiliated to **APJ ABDUL KALAM UNIVERSITY, THIRUVANANTHAPURAM**. I understand that detection of any such copying is liable to be punished in any way the college deems fit.

Date:26-October-2022

Name:Athul K Kumar

Place:Paravoor

ACKNOWLEDGEMENT

In the name of almighty **GOD**, I express my sincere thanks to him keeping me fit for successful completion of the project.

I am thankful to **Prof. Dr. SAGINI THOMAS MATHAI, Principal, SNGIST GROUP OF INSTITUTIONS** for his kind support in all respect during our study.

I sincerely thank **Prof. KAVITHA C.R, HOD, Department of Computer Applications, SNGIST GROUP OF INSTITUTIONS**, for her encouragement to carry out this project.

I especially thank my Guide **Ms. TINTU P B, Department of Computer Applications, SNGIST GROUP OF INSTITUTIONS**, in examining the draft of this seminar and for valuable suggestions.

I want to thank the Department of Computer Applications for giving me the permission to prepare the project on the topic “**GAMING WEBSITE**”.

ATHUL K KUMAR

Contents

1	EXECUTIVE SUMMARY	1
2	INTRODUCTION	1
2.1	EXISTING SYSTEM	2
2.2	PROBLEM DEFINITION	3
2.3	PROPOSED SYSTEM	3
2.4	OBJECTIVE OF THE PROJECT	3
2.5	SCOPE OF THE PROJECT	3
2.6	HARDWARE REQUIREMENTS	4
2.6.1	Operating system Windows 7 and above	4
2.6.2	ETHERNET	4
2.6.3	SSD (Solid State Drive)	4
2.7	SOFTWARE REQUIREMENTS	5
2.7.1	HTML (Hyper Text Markup Language)	5
2.7.2	Javascript	5
2.7.3	VS Code	6
3	METHODOLOGY	6
3.1	SCRUM	6
3.2	SCRUM ROLES	6
3.2.1	Product Owner	6

3.2.2	Scrum Master	7
3.2.3	Scrum Team	7
3.3	SPRINT PLANNING MEETING	7
3.4	DAILY SCRUM MEETING	8
3.5	SPRINT REVIEW MEETING	8
3.6	PRODUCT BACKLOG	8
3.6.1	USER STORIES	8
4	MILESTONES	9
4.1	Sprint 1	9
4.2	Sprint 2	9
4.3	Sprint 3	10
4.4	Sprint 4	10
5	MODULE DESCRIPTION	10
5.1	Admin Module	10
5.2	User Module	10
6	SYSTEM DESIGN	11
6.1	UI DESIGN	11
6.2	UML DIAGRAM	11
6.2.1	USE CASE DIAGRAM	11
6.2.2	CLASS DIAGRAM	12

6.3 DATABASE DESIGN	12
7 TESTING	13
7.1 TEST CASE	14
7.1.1 Login	14
7.1.2 REGISTRATION	15
7.1.3 TEST REPORT	15
8 SYSTEM IMPLEMENTATION	16
8.1 SCREENSHOTS	16
8.1.1 Landing Page	16
8.1.2 Login Page	16
8.1.3 Game Page	17
8.1.4 Scramble game	17
8.1.5 Memory game	18
9 CONCLUSION AND FUTURE ENHANCEMENT	18
10 APPENDIX A	19
10.1 SAMPLE SOURCE CODE	19
11 APPENDIX B	21
11.1 WEBLIOGRAPHY	21
11.2 BIBLIOGRAPHY	21

List of Figures

1 EXECUTIVE SUMMARY

The online website for gamers. Users can play different games after registering their account. We can interact with the website. When the game finished the score will be displayed to the user.

2 INTRODUCTION

Smartphone usage computer technology is on rise. As of 2012, almost half of the users are using this new technology. Every day we see major technological enhancements in mobile phones and tablets such as faster processors with more RAM and enhanced GPU capabilities. If we do a comparison amongst Samsung's flagship phones over the last three years, we can see that processing power has almost doubled with each model . Today, we can effortlessly run graphic intensive applications on smartphones and other computer devices it has become a very large market for games. Mobile games constitute a major share of the applications in these various operating systems. They are amongst the most successful, and highest grossing apps in app stores today. Namely, iOS apps are written in Objective-C, Java drives Android application development, and Windows Phone development is done in C. In order to develop an application which runs on all these platforms include other devices we need to develop an web application using javascript. The investment in terms of learning, designing, development and testing games on all of these plat-

forms can be hard for smaller companies and start-ups. The question is, can we use some other technology such as HTML5, which is also rapidly growing? The games created using JS engine can be run on different platforms with a minimum of modifications. A lot of developers today develop games using game engines, as it simplifies the design process and reduces the time to design radically. There are numerous frameworks present in different languages such as Box2D in C++, Unity3D in flash, Crafty in JavaScript, etc., that are created to develop desktop or console based video games. However, a game engine designed specifically to develop games is of great interest and also is an upcoming field in today's market. We experimented with Crafty, a JavaScript game engine, in the process of designing game. We created some games to better understand the working of the framework. In order to create javascript games, we leveraged JavaScript and HTML5 functionalities. These include canvas for drawing and local storage for data persistence. Four test games were created with each testing a specific set of functionalities provided by the engine.

2.1 EXISTING SYSTEM

In the existing system the games that we found on the website will be paid or the site will contain of lots of advertisement, it will provide an hard time for the users or gamer who are interacting with that website. Some website provides monthly subscription to achieve more advantages over the website.

2.2 PROBLEM DEFINITION

The sites which provides payment options which cant be afforded by every one and also the advertisement that are displayed during the game causes the very bad time.

2.3 PROPOSED SYSTEM

This gaming site will provide data security and easy of accessing the site. Also we will not provide any payment or charges for playing the game. We will not provide any advertisement to the user or gamer if they are highly interacting with the website. There will not be any monthly subscription to acquire more advantages over the website, every body can access it in free of cost.

2.4 OBJECTIVE OF THE PROJECT

The main objective behind this website is to provide an user friendly environment and have providing fun time while playing with the games. It feels relaxed to the users. And provides an best user interface experience to everyone. No advertisement will be provided to the user while they are engaging with the site.

2.5 SCOPE OF THE PROJECT

Gaming gives relaxation and enjoyment to every user. In this busy world, gaming is the solution to release the depression and tension. Social networking with gaming is a nice combo for any user who was addicted to the world of gaming. The re-

uirements specified in this document will be used for designing all the aspects and components of the game. The document will be updated as the requirements grow and changeover the design and development process

2.6 HARDWARE REQUIREMENTS

2.6.1 Operating system Windows 7 and above

Microsoft Windows, also called Windows and Windows OS, computer operating system (OS) developed by Microsoft Corporation to run personal computers (PCs). Featuring the first graphical user interface (GUI) for IBM-compatible PCs, the Windows OS soon dominated the PC market. Approximately 90 percent of PCs run some version of Windows.

2.6.2 ETHERNET

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications sensors detect general movement, but do not give information on who or what moved.

2.6.3 SSD (Solid State Drive)

An SSD, or solid-state drive, is a type of storage device used in computers. This non-volatile storage media stores persistent data on solid-state flash memory. SSDs

replace traditional hard disk drives (HDDs) in computers and perform the same basic functions as a hard drive. But SSDs are significantly faster in comparison. With an SSD, the device's operating system will boot up more rapidly, programs will load quicker and files can be saved faster.

2.7 SOFTWARE REQUIREMENTS

2.7.1 HTML (Hyper Text Markup Language)

HTML is a markup language that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on.

2.7.2 Javascript

JavaScript is a lightweight programming language that web developers commonly use to create more dynamic interactions when developing web pages, applications, servers, and or even games. Developers generally use JavaScript alongside HTML and CSS. The scripting language works well with CSS in formatting HTML elements. However, JavaScript still maintains user interaction, something that CSS cannot do by itself.

2.7.3 VS Code

Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

3 METHODOLOGY

3.1 SCRUM

Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is a lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.

3.2 SCRUM ROLES

3.2.1 Product Owner

Mr.Shameer K S ,Associate professor,senior faculty,was the product owner for this project ,and acted as spokesman for the customer and defines features of the product based on each backlog items or each specific request of the customer.He would pri-

oritize these features according to the market values, decide on a release date for the product, and is responsible for the profitability of the product . The product owner should also adjust the contents of the features and their priority after every sprint and decide if what has been produced is acceptable.

3.2.2 Scrum Master

Dr.C.R.Kavitha, HOD MCA was the Scrum master for this project. The Scrum master is responsible for making sure a Scrum team lives by the values and practices of Scrum, and for removing any impediments to the progress of the team. As such, she should shield the team from external interference's, and ensure that the Scrum process is followed, including issuing invitations to the daily Scrum meetings.

3.2.3 Scrum Team

The Scrum team consists of a group of people developing the software product. In this project, the scrum team consists of Mr . Shameer, the product owner, Dr.C.R.Kavitha, who acted as Scrum master as well as the project supervisor Ms.Tintu P B and Athul K Kumar, Developer. There is no personal responsibility in Scrum, the whole team fails or succeeds as a single entity.

3.3 SPRINT PLANNING MEETING

Most of the time our sprint planning meetings went as planned, though sometimes the product owner was unavailable . In these cases, the meeting simply needed to be

scheduled one or two days later . These extra days would come in handy for cleaning up what we had produced the earlier sprint.

3.4 DAILY SCRUM MEETING

Our daily Scrums took place at 10am. People could arrive as early as 9.00am and work until then, but as long as they did arrive before the meeting started it did not matter(formally).

3.5 SPRINT REVIEW MEETING

Our review meetings were always held on fridays . The product owner would visit the team project room along with any other interested parties, and the team would demonstrate new features on a live system, and answer any questions that might arise during the demo . Usually, we would spend one or two days before the demo checking id everything was working, and run test demonstrations internally.

3.6 PRODUCT BACKLOG

3.6.1 USER STORIES

1. As an admin I can enter into the site by adding email and password.
2. As an admin I can add games and view number of users who logedin to the site.
3. As an user I can register my account by adding email , password and other details.

4. As an user I can login to the site by adding email and password.
5. As an user I can view the list of games.
6. As an user I can play the games.
7. As an user I can see my score.
8. As an user I can view my account details.

4 MILESTONES

4.1 Sprint 1

Conducted the first meeting with the college authority on 10th August 2022 and gathered requirements for the development of the system. The requirement collection was mainly done in this sprint which took nearly 1 week and the sprint ended on 18th August 2022.

4.2 Sprint 2

Second sprint started on 22nd August 2022 in which the database design was started and the user interface of the system was also designed. Started form designing. Development of the admin module was also initiated at this time. information.

4.3 Sprint 3

Third sprint started on 5th September 2022 in which the other module's development was started.

4.4 Sprint 4

Fifth sprint started on 7th October 2022 in which testing and validation are done for each modules.

5 MODULE DESCRIPTION

5.1 Admin Module

Where admin can login and see the users who are created the account.

5.2 User Module

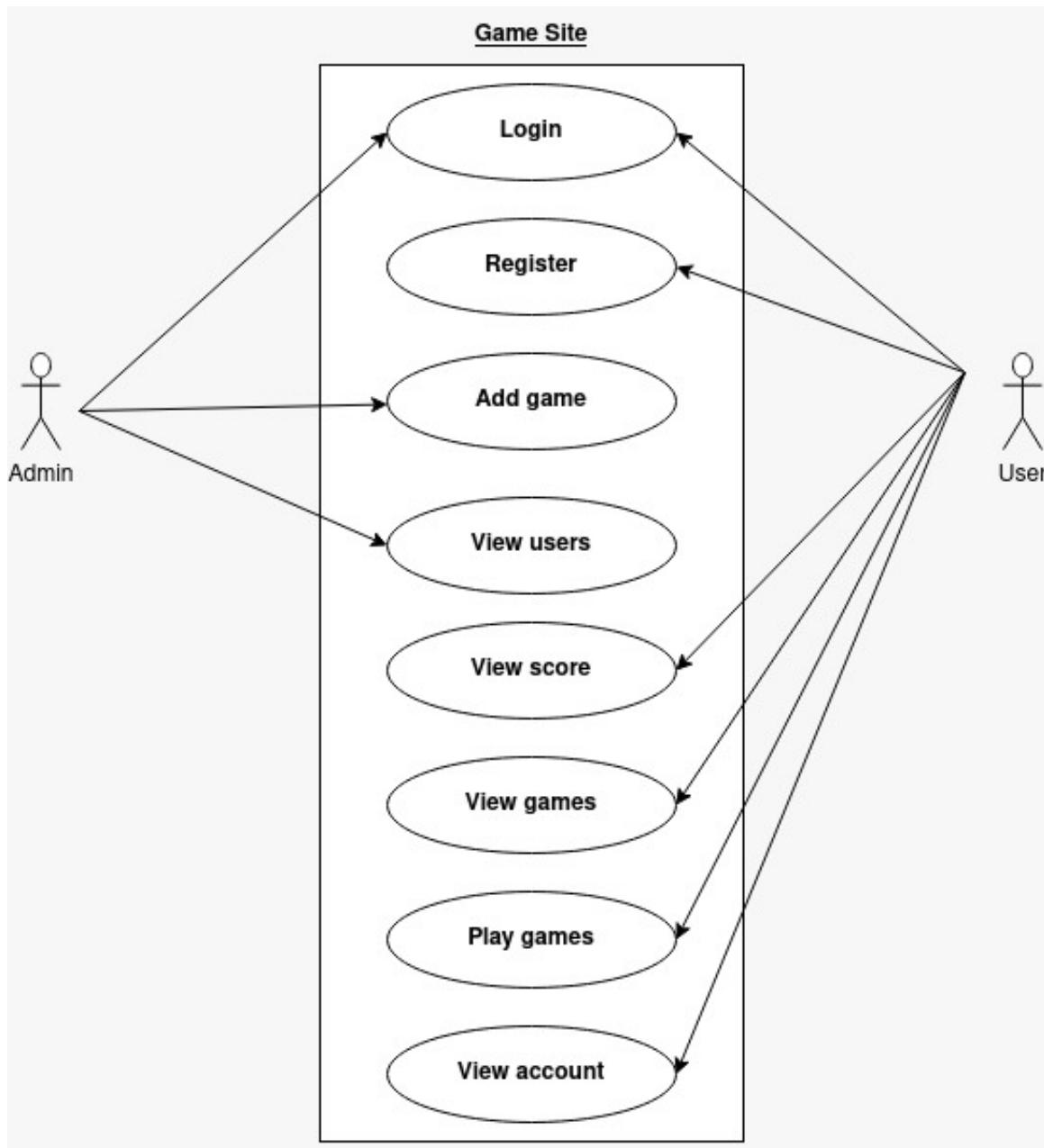
We can see an list of games where user can interact and play it.

6 SYSTEM DESIGN

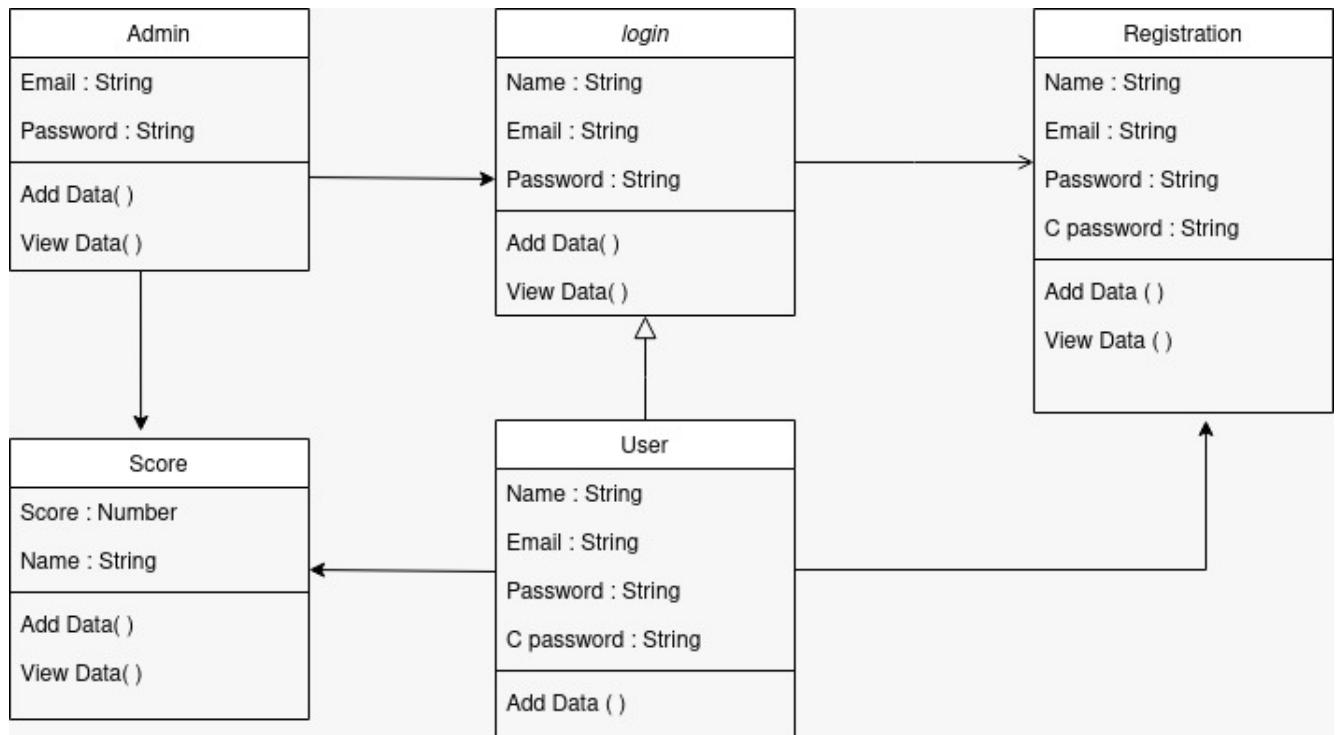
6.1 UI DESIGN

6.2 UML DIAGRAM

6.2.1 USE CASE DIAGRAM



6.2.2 CLASS DIAGRAM



6.3 DATABASE DESIGN

The screenshot shows a MongoDB database interface with three collections listed:

- admins**: Storage size: 20.48 kB, Documents: 1, Avg. document size: 61.00 B, Indexes: 1, Total index size: 20.48 kB.
- scoreboards**: Storage size: 20.48 kB, Documents: 4, Avg. document size: 88.00 B, Indexes: 1, Total index size: 36.86 kB.
- users**: Storage size: 20.48 kB, Documents: 7, Avg. document size: 228.00 B, Indexes: 1, Total index size: 36.86 kB.

At the top, there are buttons for Create collection, View, Sort by (Collection Name), and a search bar.

```

_id: ObjectId('63303457f5124b120d02df33')
username: "123"
email: "123@gmail.com"
password: "$2b$10$GvH.MG61rrsl92IPjoSx5.D3wEYOCXrvI4prtHnmW2Tq25zQlzs9C"
cpassword: "$2b$10$r9cLvrzhSpTA00EB5NrK.74OQgO2Gbh3Zcs3RKgRbrMxPSP4XX/o"
_v: 0

_id: ObjectId('6330355133d19d43b869bcc7')
username: "hello"
email: "hello@gmail.com"
password: "$2b$10$DAR2efhFTThD4Hxh.AyYbFO2ToEeboH4zsCvsWF3GIJexg2nKXrazG"
cpassword: "$2b$10$WVpsAzMCLz3YBE0XymRnuHQuYiwXPUXLzMWhMu18GqiOtSvhZSO"
_v: 0

_id: ObjectId('633277e93ed5157db7158645')
username: "vyshakh"
email: "vys@gmail.com"
password: "$2b$10$tFDij2WN25xai3pJe8zYPe/5KRVD1bFSe8NtGqYaQ8iKbNbe4DCTq"
cpassword: "$2b$10$IXdS/YCQ.2LUvla09jJjwOHGvbW9pVloX6Mc5KnFQvQ4VOSeWDB06"
_v: 0

_id: ObjectId('634cd60230a34409cc832c71')

```



```

_id: ObjectId('634b8e894b1bb440c5909896')
user: "63303457f5124b120d02df33"
> wordsrambler: Object

_id: ObjectId('634b90c67503c12eedb65d9c')
user: "6330355133d19d43b869bcc7"
> wordsrambler: Object

_id: ObjectId('634cdf9d2b6a965a99ca19b2')
user: "634cdf782b6a965a99ca19ac"
> wordsrambler: Object

_id: ObjectId('634d0947a95e1b3e553e85a5')
user: "634cd60230a34409cc832c71"
> wordsrambler: Object

```

7 TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques

include the process of executing a program or application with the intent of finding software bugs . Testing can be stated as the process of validating and verifying that the software product such as meets the requirements that guided its design and development, responds correctly to all kinds of inputs, performs its functions within an acceptable time.

7.1 TEST CASE

7.1.1 Login

SL.NO	INPUT	EXPECTED RESULT	OBSERVED RESULT
1	Click Login button with invalid username and password.	Display “invalid username and password”.	Display “invalid username and password”.
2	Click Login button with valid username and password.	Display “valid password”.	Display “valid password”.
3	Click Login button with each field as blank.	Display “data required”.	Display “data required”.

7.1.2 REGISTRATION

SL.NO	INPUT	EXPECTED RESULT	OBSERVED RESULT
1	Click sign up button without giving values to some fields.	Display “data required on blank field”.	Display “data required on blank field”.
2	Click sign up button with invalid registration details.	Display “invalid credentials”.	Display “invalid credentials”.
3	Click sign up button with valid registration details.	Display “successfully registered”.	Display “successfully registered”.
4	Click sign up button with invalid email.	Display “invalid email address”.	Display “invalid email address”.
5	Click sign up button with already registered email.	Display “email already exist. Try another”.	Display “email already exist. Try another”.

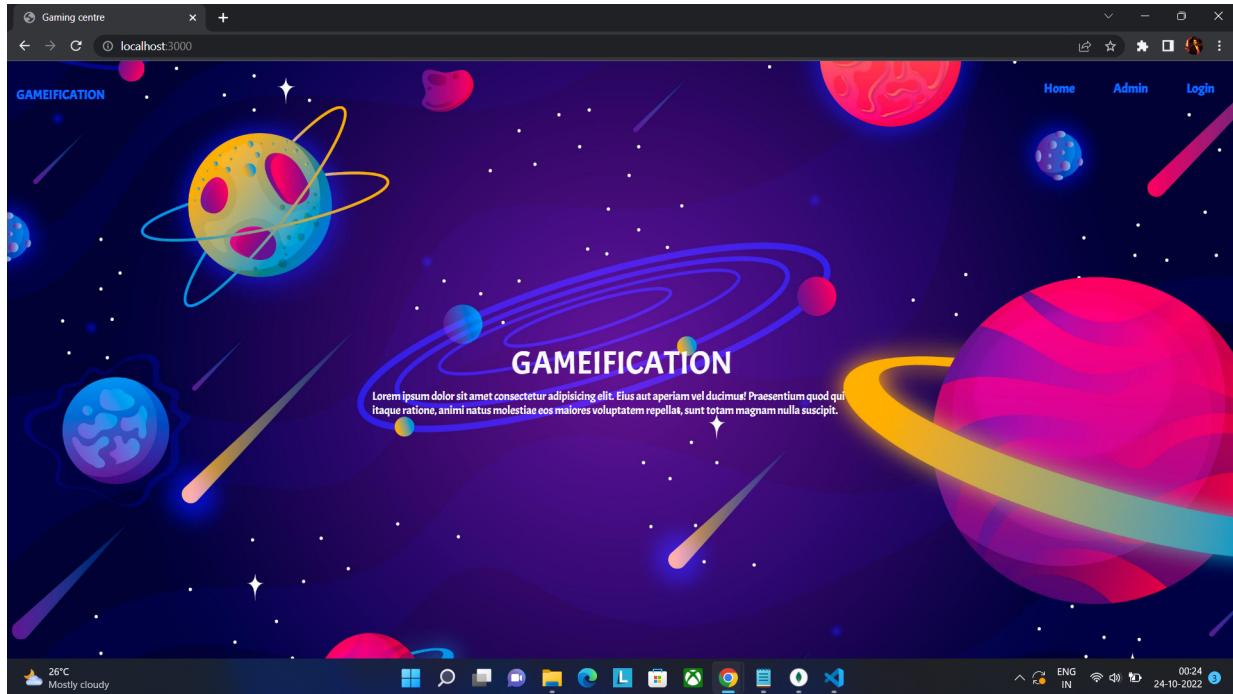
7.1.3 TEST REPORT

TEST CASE NO	PASS/FAIL	EXPECTED RESULT	ACTUAL RESULT
TC-01	Pass	The details should be stored in the DB.	The details are stored in the DB.
TC-02	Pass	Output should be displayed and tables are updated.	Output is displayed and tables are updated.
TC-03	Pass	Output should be displayed and tables are updated.	Output is displayed and tables are updated.
TC-04	Pass	Output should be displayed and tables are updated.	Output is displayed and tables are updated.

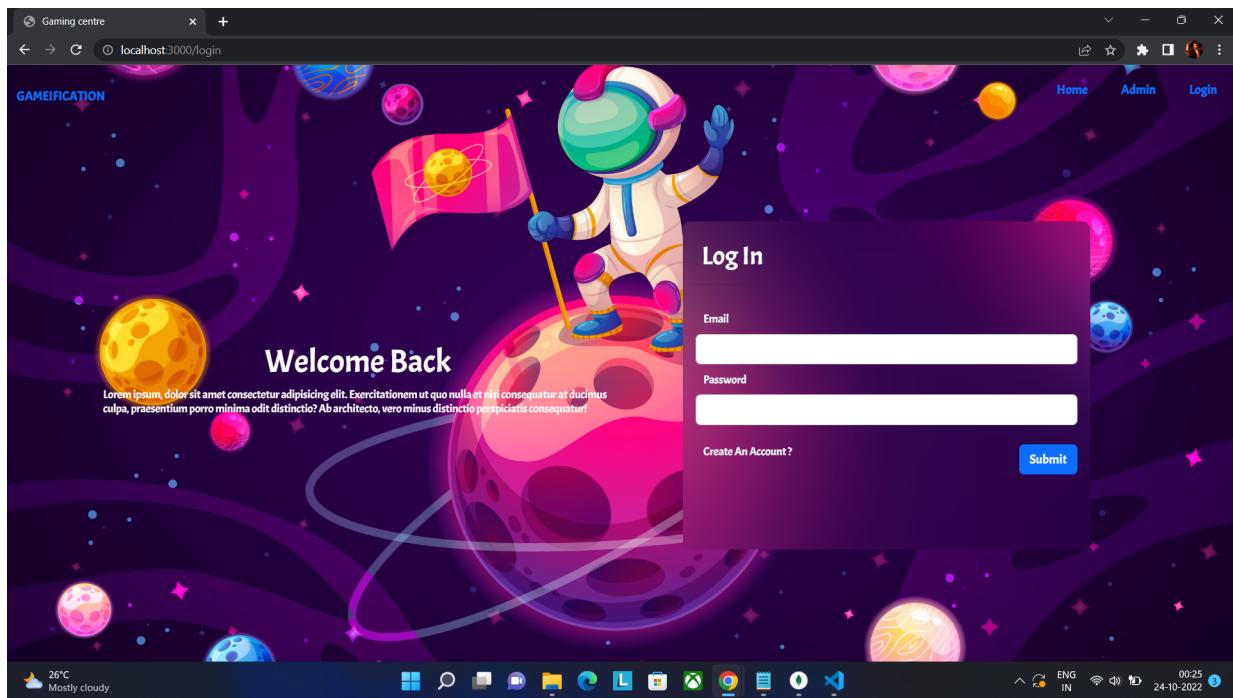
8 SYSTEM IMPLEMENTATION

8.1 SCREENSHOTS

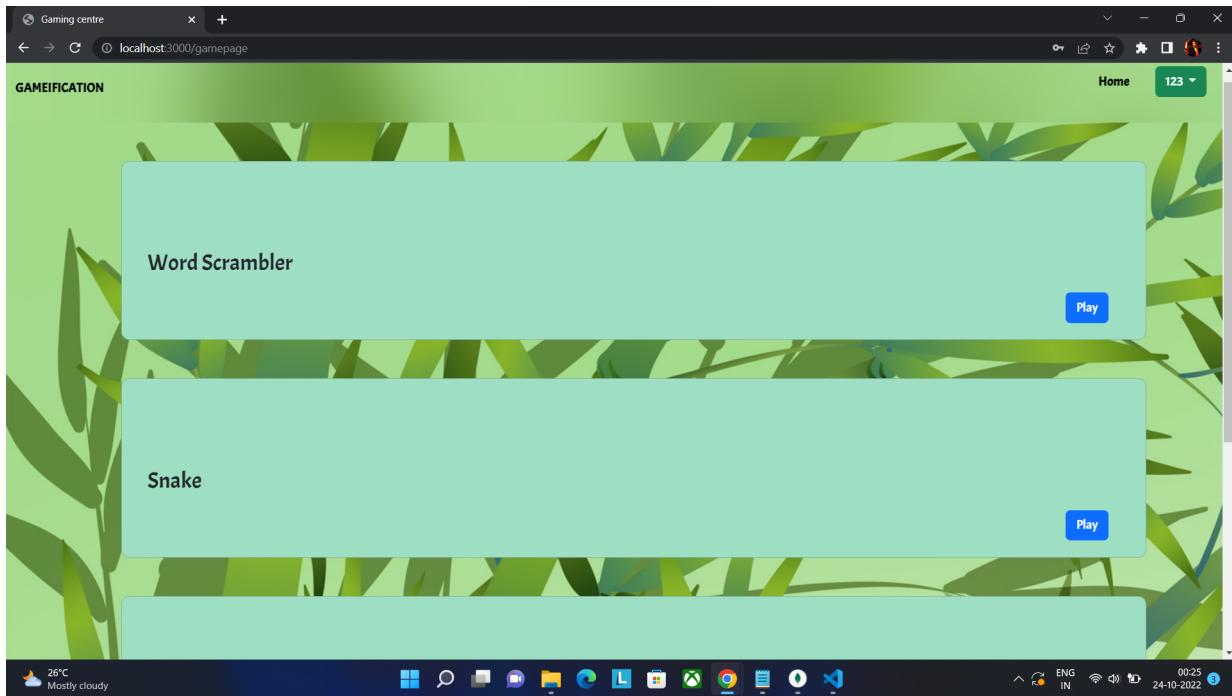
8.1.1 Landing Page



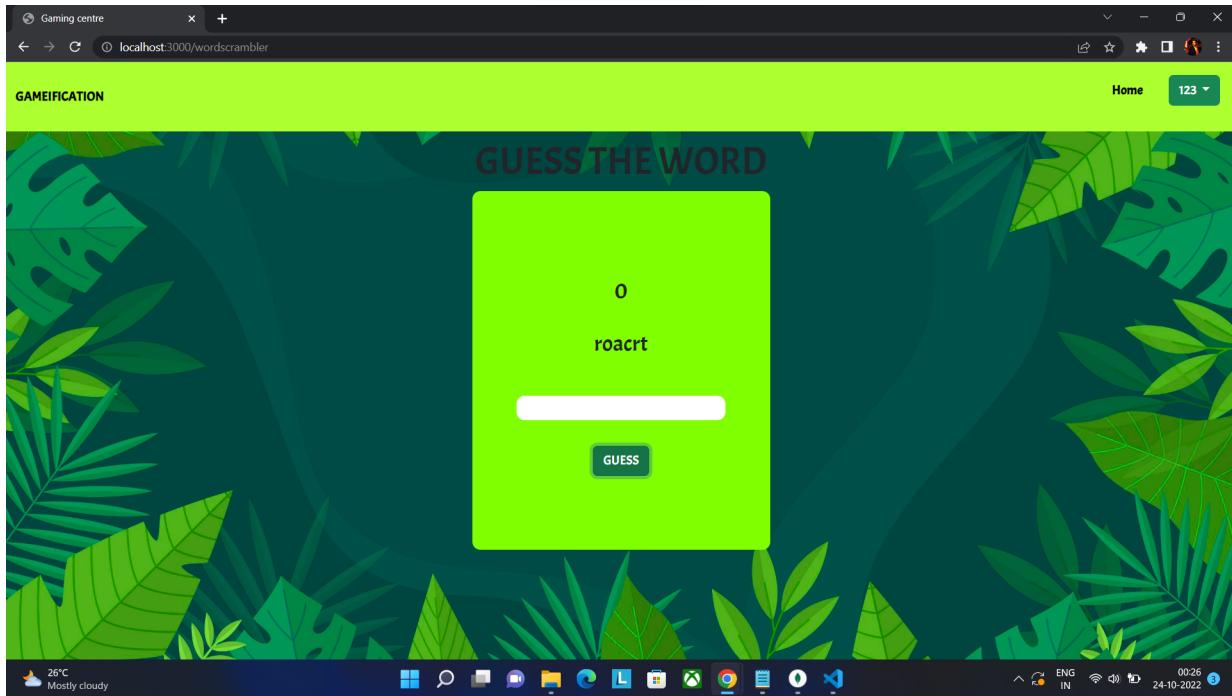
8.1.2 Login Page



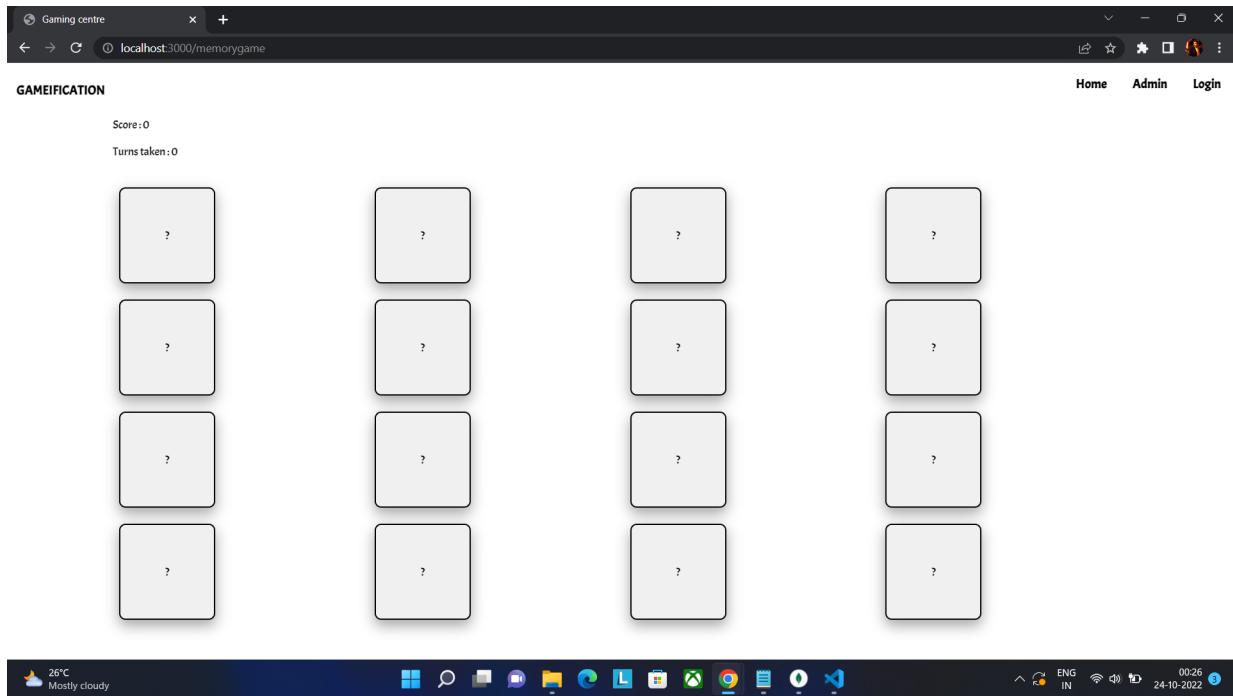
8.1.3 Game Page



8.1.4 Scarmble game



8.1.5 Memory game



9 CONCLUSION AND FUTURE ENHANCEMENT

Gaming is on the rise. Although a lot of users are not hard-core gamers, a significant lot of them try the games on smartphones and other devices in their leisure time. Therefore, a lot of developers are looking for an opportunity to get into this market of game. There are some roadblocks in order to start this development, which include the platform to choose, the different types of hardware, and most importantly, the learning curve. The idea of this project was to specifically address these issues. A game site called gamification has been created with the help of web development technologies such as HTML5, Css, Javascript, MongoDb, local storage and hardware related JavaScript events and APIs. As shown in the test games, this site is simpler to use. Future improvements to the system could include improvements in

the user's experience of installing and launching games, more robust data persistence technique. Appropriate monetization processes can also be deployed, so that the developers can sell their work comfortably.

10 APPENDIX A

10.1 SAMPLE SOURCE CODE

The screenshot shows a code editor interface with two tabs open:

- admin.js** (Top Tab):


```

1 <!doctype html>
2 <html lang="en">
3
4   <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>Gaming centre</title>
8
9     {{!-- css links --}}
10    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/css/bootstrap.min.css" rel="stylesheet"
11      integrity="sha384-Xe+8c90Ja6thvch5P7q+mSPa58Cu9mPX5F5x1GE00V1taq75lOrf0E17Vk"
12      crossorigin="anonymous">
13    <link rel="stylesheet" href="/stylesheets/{{style}}>
14  </head>
15
16  <body>
17    {{!-- header section --}}
18    {{if admin}}
19      {{>admin-header}}
20    {{else}}
21      {{>user-header}}
22    {{/if}}
23
24    {{!-- body section --}}
25    {{>body}}
26
27    {{!-- scripts link section --}}
28    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.5/dist/umd/popper.min.js"
29      integrity="sha384-Xe+8c90Ja6thvch5P7q+mSPa58Cu9mPX5F5x1GE00V1taq75lOrf0E17Vk"
30      crossorigin="anonymous"></script>
31    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.min.js"
32      integrity="sha384-0mDivhPfdexHEHF8QH3/9/vQ9uori45z4JhnFSRydbmQbmL5titQeculuzyk"
33      crossorigin="anonymous"></script>
      
```
- users.js** (Bottom Tab):


```

1 var express = require('express');
2 var router = express.Router();
3 const userHelpers = require('../helpers/user-helpers')
4 const verifyLogin = require('../middleware/verify-user')
5 require('../db/connection')
6
7 /* GET users listing. */
8 router.get('/', function (req, res, next) {
9   res.render('user/index', { style: 'style.css' })
10 });
11 //Login page
12 router.get('/login', (req, res) => {
13   if (req.session.loggedin) {
14     res.redirect('/gamespace')
15   } else {
16     res.render('user/user-login', { style: 'login.css', 'loginErr': req.session.loginErr })
17     req.session.loginErr = false
18   }
19 }
20 )
21
22 router.get('/signup', (req, res) => {
23   res.render('user/user-signup', { style: 'signup.css' })
24 })
      
```

Both tabs show a list of files in the left sidebar, including views, routes, and stylesheets. The bottom tab also includes a terminal window at the bottom showing the execution of node commands and the resulting network traffic logs.

EXPLORER ... JS admin.js ↗ user-login.hbs × # adminlogin.css M # gamepage.css 1.M ↗ admin-login.hbs M

views > user > ↗ user-login.hbs > ↗ section.login-section > ↗ div.container.mt-4 > ↗ div.row.login-left-right > ↗ div.col-md-6.login-right > ↗ div.container.login-part.p-2

```

1  <section class="login-section">
2  	<div class="container mt-4">
3   	<div class="row login-left-right">
4    	<div class="col-md-6 login-left">
5     	<h1>Welcome Back</h1>
6
7     	<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Exercitationem ut quo nulla et nisi
8      	consequatur at ducimus culpa, praesentium porro minima odit distinctio? Ab architecto, vero minus
9      	distinctio perspiciatibus consequatur!</p>
10
11    	<!-- Login part -->
12
13    	<div class="col-md-6 login-right">
14     	<div class="container login-part p-2">
15       	<h2 class="m-3" style="color: white;">Log In</h2>
16       	<hr>
17
18       	{{!-- checks if the user enters valid data or not --}}
19
20       	{{#if loginErr}}
21          <div class="alert alert-danger position-relative" role="alert">
22            Invalid credentials
23          </div>
24        {{/if}}
25
26        {{!-- form body --}}
27
28        <form action="/login" method="post" class="p-2">
29          <label for="">Email</label>
30          <input type="email" name="email" class="form-control">
31
32          <label for="">Password</label>
33          <input type="password" name="password" class="form-control">

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

GET /stylesheets/memorygame.css 304 11.271 ms - -
GET /javascripts/memorygame.js 304 6.417 ms - -

EXPLORER ... JS admin.js ↗ user-signup.hbs × # adminlogin.css M # gamepage.css 1.M ↗ admin-login.hbs M

views > user > ↗ user-signup.hbs > ↗ section > ↗ div.container.mt-4 > ↗ div.row.signup-left-right > ↗ div.col-md-6.signup-right

```

1  <section>
2  	<div class="container mt-4">
3   	<div class="row signup-left-right">
4
5    	<div class="col-md-6 signup-left">
6      	<h3>Create An Account</h3>
7      	<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Harum, cumque. Sed excepturi libero
8       	repudiandae et ratione illo neque architecto nisi ullam! Eaque, quam animi.</p>
9
10
11   	{{!-- Sign up form part --}}
12   	<div class="col-md-6 signup-right">
13     	<div class="container signup-part">
14       	<h2 class="pt-3">Sign UP</h2>
15       	<hr>
16
17        <form action="/signup" method="post" class="mt-2">
18          <label for="">User Name</label>
19          <input type="text" name="username" class="form-control">
20
21          <label for="">Email</label>
22          <input type="email" name="email" class="form-control">
23
24          <label for="">Password</label>
25          <input type="password" name="password" class="form-control">
26
27          <label for="">Confirm Password</label>
28          <input type="password" name="cpassword" class="form-control">
29
30          <a href="/login" class="mt-4">Already have account?</a>
31
32          <button type="submit" class="btn btn-primary mt-4">Submit</button>
33

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

GET /stylesheets/memorygame.css 304 11.271 ms - -
GET /javascripts/memorygame.js 304 6.417 ms - -

EXPLORER ... JS admin.js # adminlogin.css M # gamepage.css 1.M ↗ admin-login.hbs M JS wordscraper.js

public > JavaScripts > JS wordscraper.js ↗ btn.addEventListener('click') callback

```

1  const msg = document.querySelector('.msg');
2  const guess = document.querySelector('input');
3  const btn = document.getElementById('btn');
4  const score = document.querySelector('.score');
5
6  let play = false;
7  let newword = '';
8  let randomWord = '';
9  let swords = ['apple', 'orange', 'mango', 'pineapple', 'grapes', 'banana', 'carrot', 'potato', 'watermelon', 'strawberry'];
10 let scoreCount = 0;
11 //Function to pick random words from the swords
12 const createNewWord = () => {
13 	let randomNumber = Math.floor(Math.random() * swords.length); //To get an random number and pass in the array index
14 	let newTempWords = swords[randomNumber]; //It will choose an random element from the sword array
15 	return newTempWords;
16 }
17 //Function to scramble the word
18 const scrambleWord = (arr) => {
19 	//Loop to shuffle the selected word
20 	for (let i = arr.length - 1; i >= 0; i--) {
21 		let temp = arr[i];
22 		let j = Math.floor(Math.random() * (i + 1));
23
24 		arr[i] = arr[j];
25 		arr[j] = temp;
26 	}
27 	return arr;
28 }
29 //If the user press the click start button
30 btn.addEventListener('click', function () {
31 	if (!play) {
32
33

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

GET /stylesheets/memorygame.css 304 11.271 ms - -
GET /javascripts/memorygame.js 304 6.417 ms - -

```

public > javascripts > memorygame.js > # adminlogin.css M # gamepage.css 1, M ~ admin-login.hbs M
1 //Pascalcase followed
2 var lastbutton = null; //Last button that was clicked.
3 var score = 0; //Score, i.e. how many Cards are opened
4 var turns = 0; //Number of turns taken.
5
6 const Cards = []; //Array to store the Cards.
7 //Setting the Cards.
8 for (let j = 0; j < 8; j++) {
9   r = Math.floor(Math.random() * (90 - 64 + 1)) + 64; //Character set from unicode.
10  Cards.push(String.fromCharCode(r)); //Push twice.
11  Cards.push(String.fromCharCode(r));
12 }
13 Cards.sort(() => 0.5 - Math.random()); //Random shuffling of Cards.
14 Cards.push(Cards[0]); //Ignore the Card at 0th positon.
15 var style =
16 "box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0 rgba(0,0,0,0.19);transition-duration: 0.2s;cursor: pointer; border-radius: 8px;";
17 for (let f = 1; f < 16; f++) {
18   document.getElementById("button" + f).style = style;
19   document.getElementById("button" + f).hover = "(color: green;)";
20   //document.getElementById("button" + f).hover="background-color: white; color: black; border: 2px solid #4CAF50;""
21   //Button formatting
22 }
23 function Buttonclicked(a) {
24   //Function to be executed once button is clicked.
25   //a is the parameter which tells which button is clicked.
26   document.getElementById("button" + a).innerHTML = Cards[a]; //Flip the Card over.
27   document.getElementById("button" + a).disabled = "disabled"; //Disable clicking the same Card.
28   document.getElementById("button" + a).style =
29   | "opacity: 0.7; cursor: not-allowed; color: Blue";
30   checkscore(a); //Check the Cards.
31 }
32
33 function checkscore(a) {
PROBLEMS ① OUTPUT DEBUG CONSOLE TERMINAL
GET /stylesheets/memorygame.css 304 11,271 ms - -
GET /javascripts/memorygame.js 304 6,417 ms - -
}

```



```

views > user > memorygame.hbs > div.container > div.container > p#score
1 <div class="container">
2
3   <div class="container">
4     <p id="score">Score : 0</p>
5     <p id="Turns">Turns taken : 0</p>
6   </div>
7
8   <div class="keys">
9     <button id="button1" class="button" onclick="Buttonclicked(1)">
10    ?
11    </button>
12    <button id="button2" class="button" onclick="Buttonclicked(2)">
13    ?
14    </button>
15    <button id="button3" class="button" onclick="Buttonclicked(3)">
16    ?
17    </button>
18    <button id="button4" class="button" onclick="Buttonclicked(4)">
19    ?
20    </button>
21    <button id="button5" class="button" onclick="Buttonclicked(5)">
22    ?
23    </button>
24    <button id="button6" class="button" onclick="Buttonclicked(6)">
25    ?
26    </button>
27    <button id="button7" class="button" onclick="Buttonclicked(7)">
28    ?
29    </button>
30    <button id="button8" class="button" onclick="Buttonclicked(8)">
31    ?
32    </button>
33    <button id="button9" class="button" onclick="Buttonclicked(9)">

```

11 APPENDIX B

11.1 WEBLIOGRAPHY

11.2 BIBLIOGRAPHY