MEMORY MATCHING GAME

A project report submitted to

Rajiv Gandhi University of Knowledge & Technologies

SRIKAKULAM

In partial fulfillment of the requirements for the Award of the degree of

BACHELOR OF TECHNOLOGY IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

G. Bharat Chandra (S170711)

as *Summer Internship*

Under the Guidance of
Mrs. S. Lakshmi Sri mam
Department of CSE,
Srikakulam.

Rajiv Gandhi University of Knowledge Technologies – SKLM

CERTIFICATE

This is to certify that the thesis work entitled "**Memory Matching Game**" was successfully completed by Gedala Bharat Chandra (S170711) in partial fulfillment of the requirements for the Summer Internship in Computer Science and Engineering of Rajiv Gandhi University of Knowledge and Technologies under my guidance and output of the work carried out is satisfactory.

Mrs. S. Lakshmi Sri mam
M. Tech (Assistant professor)
Dept of CSE,RGUKT-SKLM
Project Guide

Mrs. M. Roopa mam M. Tech
M. Tech (Assistant professor)
Dept of CSE,RGUKT-SKLM
Head Of Department, CSE

DECLARATION

I hereby declare that this thesis entitled " ***Memory Matching Game*** " a web application which is carried out by me during the academic year 2021-2022 in partial fulfillment of the requirements for the summer internship in Computer Science and Engineering.

I further declare that this dissertation has not been submitted elsewhere for any Degree. The matter embodied in this dissertation report has not been submitted elsewhere for any other degree. Furthermore, the technical details furnished in various chapters of this thesis are purely relevant to the mentioned project and there is no deviation from the theoretical point of view for design, development, and implementation.

G. Bharat Chandra (S170711)

ACKNOWLEDGEMENT

I would like to articulate my profound gratitude and indebtedness to my project guide Prof. Lakshmi Sri madam(Asst. Prof),who has always been a source of inspiration, constant motivation, and guiding factor throughout the project time. It has been a great pleasure for me to get an opportunity to work under her guidance and complete the thesis work successfully.

I would like to extend my sincere thanks to Prof. M Roopa madam(Asst. Prof), Head Of Computer Science and Engineering Department, For her constant encouragement throughout the project.

I am also thankful to other members of the department with their support my work was carried out successfully.

I thank one and all who have rendered help to me directly or indirectly in the completion of my thesis work.

Project Associate

G. Bharat Chandra (S170711)

ABSTRACT

Universities and colleges are the places where students get jam-packed with loads of work like classes, campus jobs or internships, ...etc in a way of attaining future goals. so the days become stressful and restless. On such stressful days students search for stress busters for mind relaxation. we obviously know those are exercising, sleeping and games like BGMI and Call of Duty. Playing such kind of games may lead to adverse things like addiction. But games like memory games may help them sharpen their brains by providing fun and relaxation.

Match them is a plain online fun game which makes the player enthusiastic. It calculates and displays the number of moves and time taken by the player to complete a level. It can be developed by using technologies like HTML, CSS, JavaScript. Playing memory games creates a positive impact on students in a way to enhance their quick response to problems, stronger attention and concentration. These memory games not only provide fun but helps to improve students critical thinking, attention to detail, and visual recognition.

1. INTRODUCTION

1.1. Introduction:

The Memory Matching Game is a classic game that tests a player's memory and concentration. The game consists of a gird of cards placed face down, and the player must flip over pairs of cards in an attempt to match them. The game is won when all pairs have been successfully matched.

1.2. Statement of the Problem:

After a stressful day, most of us prefer to play video games. But playing games like PUBG or Call of Duty may lead us to addiction. In order to have fun and sharpen our brain at the same time we must choose Memory games.

1.3. Objectives

- The main objective of memory matching game is to for the player to flip over cards and match pairs of identical images or symbols.
- The goal is to remember the location of the cards and match them as quickly and accurately as possible, usually in the least number of moves or in the shortest amount of time.

1.4. Goals

- Basic GUI.
- Basic Controls.
- User-friendly interface.

1.5. Scope

- This application can be used by anyone who want to play simple memory games.
- This application helps us to sharpen our brains and improve memory power.

1.6. Limitations

- Limited to certain number of cards and images.
- May not be suitable for players with certain cognitive or visual impairments.
- The player's memory capacity may not be sufficient to remember all the cards or images.

2. LITERATURE SURVEY

2.1. Collecting Information:

We have taken the information from various memory games like cards, and other online games. This include looking at the history and development of memory games and their variations, as well as their cognitive and educational benefits.

2.2. Benefits:

- Improve memory power.
- Sharpen our brains.

2.3. Summary

In our system we build a simple web application using HTML, CSS and jQuery. This application consists of a grid of images or symbols faced down, and the player must flip over pairs of cards in an attempt to match them.

3. SYSTEM ANALYSIS

3.1. Existing System

There are number of existing memory matching games like memozor and matchmemory. These games typically include an interactive UI, and many more features.

- device, mobile device, etc. The two FA is used two factors to confirm an identity.

3.2. Disadvantages

- Not beginner friendly
- Had a lot of features which will be hard for the user.
- Many memory games were designed to improve their memory skills which may lead a beginner hard to use.

3.3. Proposed System

- The system is a simple web application.
- This enables user to choose the grid size.
- This also enables user to know about time taken and how many steps he used to complete the task.

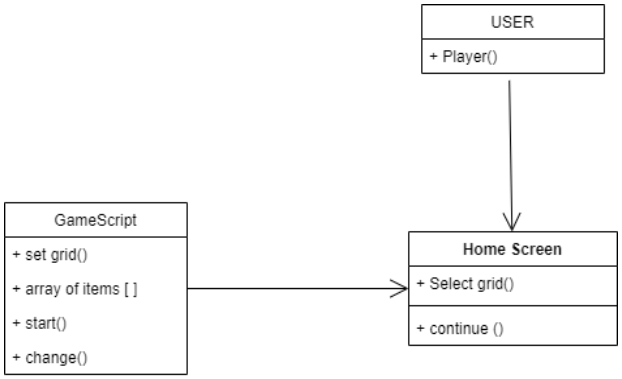
3.4. Advantages

- This helps to improve our memory skills.
- Entertainment
- Sharpens our mind

4. System Design

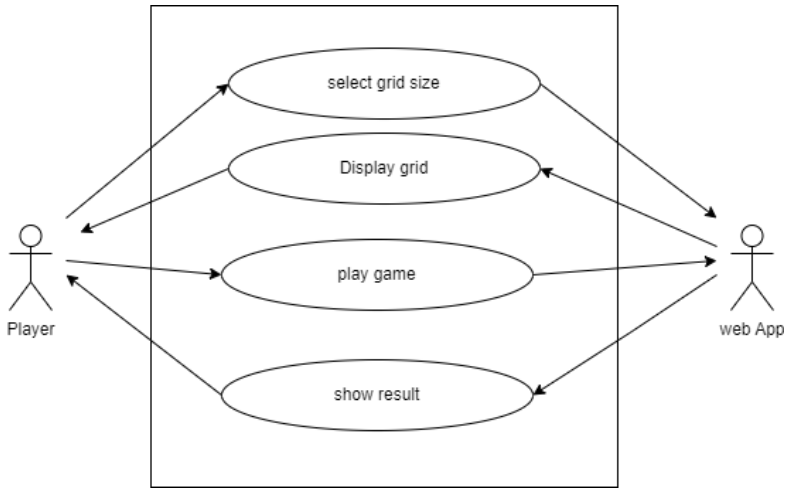
4.1 Class Diagram

A class diagram in the Unified Modelling Language (UML), is a kind of static structure diagram that describes the constitution of a process by showing the system's classes, their attributes, and the relationships between the class. The motive of a class diagram is to depict the classes within a model. In object-oriented software, classes have attributes (member variables), operations (member capabilities), and relations.



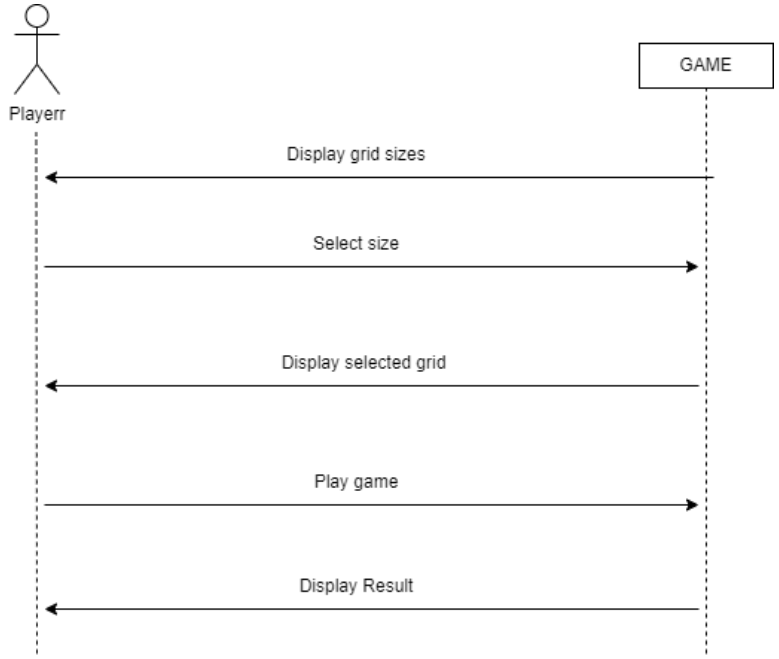
4.2 Use Case Diagram

It visually represents what happens when an actor interacts with the system. A use case diagram captures the functional aspects of a system. The system is shown as a rectangle with the name of the system inside, the actor is shown as stick figures, the use case is shown as solid bordered ovals labeled with the name of the use case and relationships are lines or arrows between the actor and use cases.



4.3 Sequence Diagram

A sequence diagram in Unified Modelling Language (UML) is one variety of interaction diagrams that suggests how methods operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are quite often referred to as event-hint diagrams, event situations, and timing diagrams. A sequence diagram suggests, as parallel vertical traces (lifelines), special systems or objects that are residing at the same time, and, as horizontal arrows, the messages exchanged between them, within the order of the place they occur.



5. SYSTEM IMPLEMENTATION

5.1. Memory Matching Game Implementation:

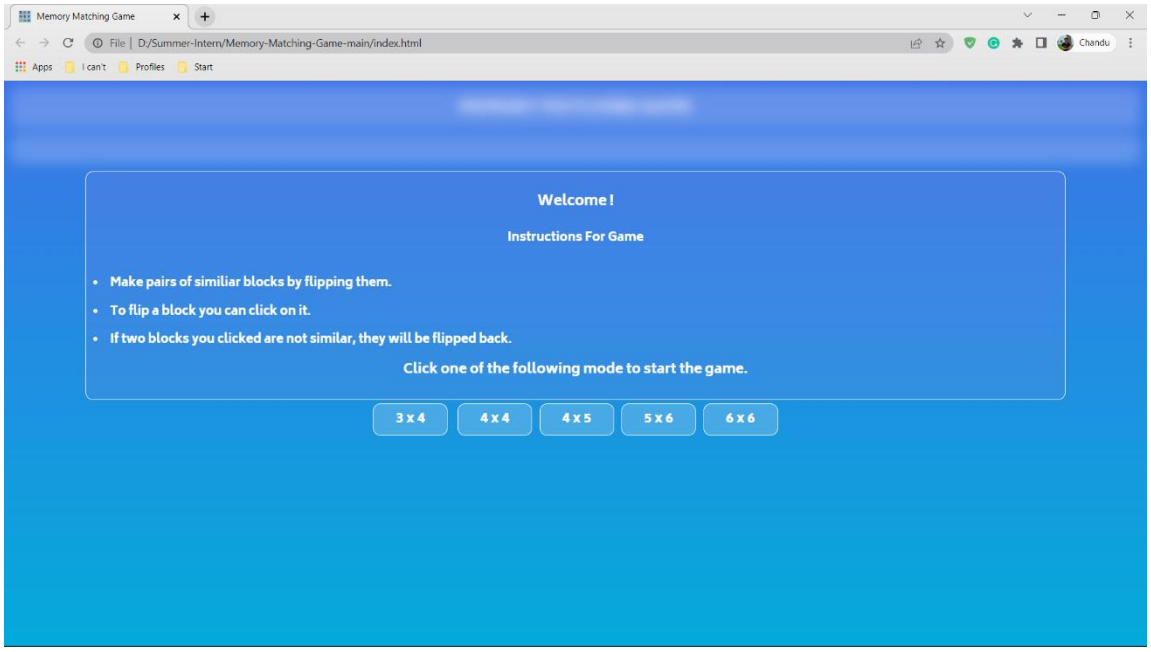
A memory matching game was implemented using HTML, CSS, and JS. jQuery is a small and feature-rich JavaScript library which is used in this application.

The basic steps for building the game include creating a game board with a grid of cards, displaying the cards with images or symbols on them, allowing the player to click on two cards to reveal their images or symbols, checking if the two cards match, and keeping track of the player's score. You can also include additional features like difficulty levels, time limits, and a leaderboard.

5.2. The Web Application:

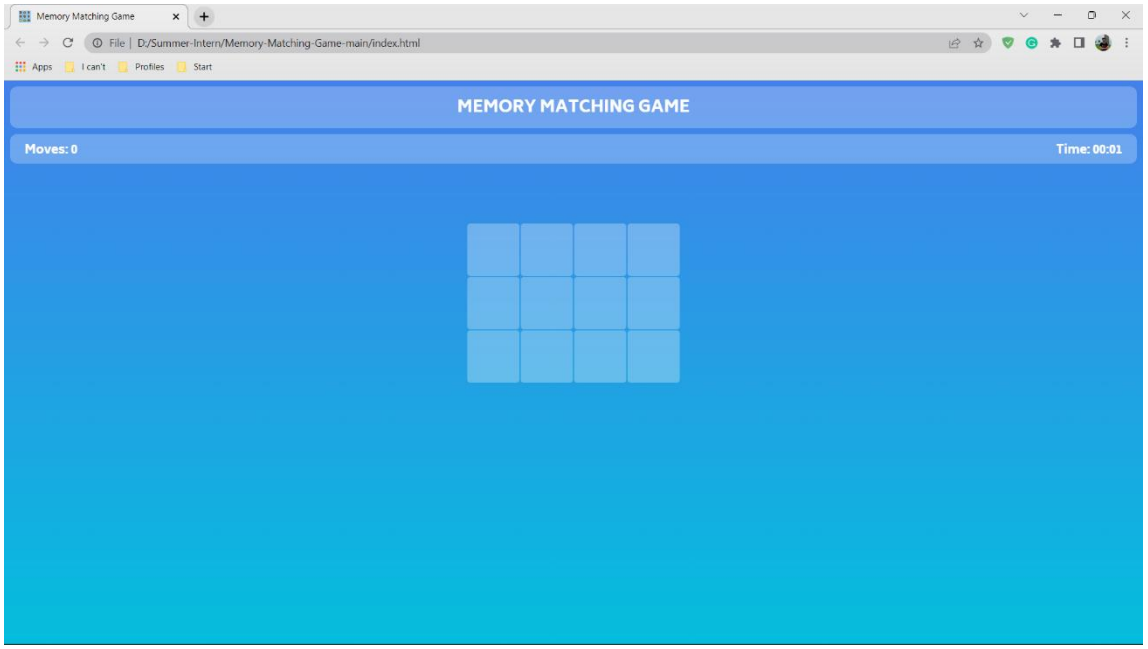
5.2.1. Home Screen

This is the home page where user can select the size of the GRID.

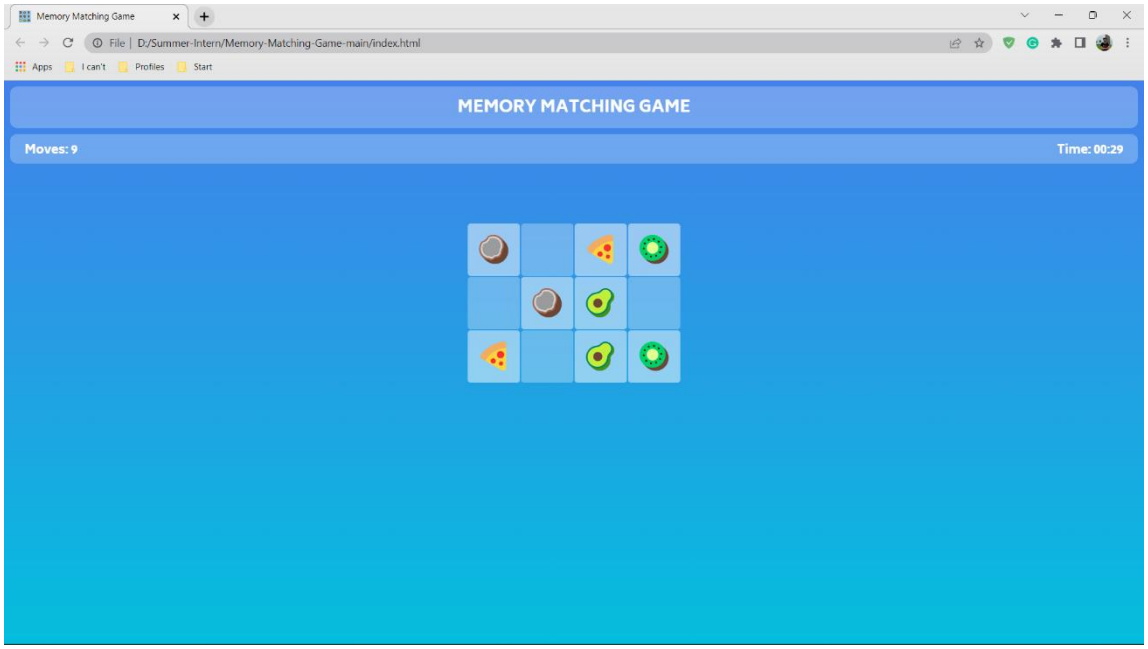


5.2.2. Game Screen

The user is given a grid consisting of images.

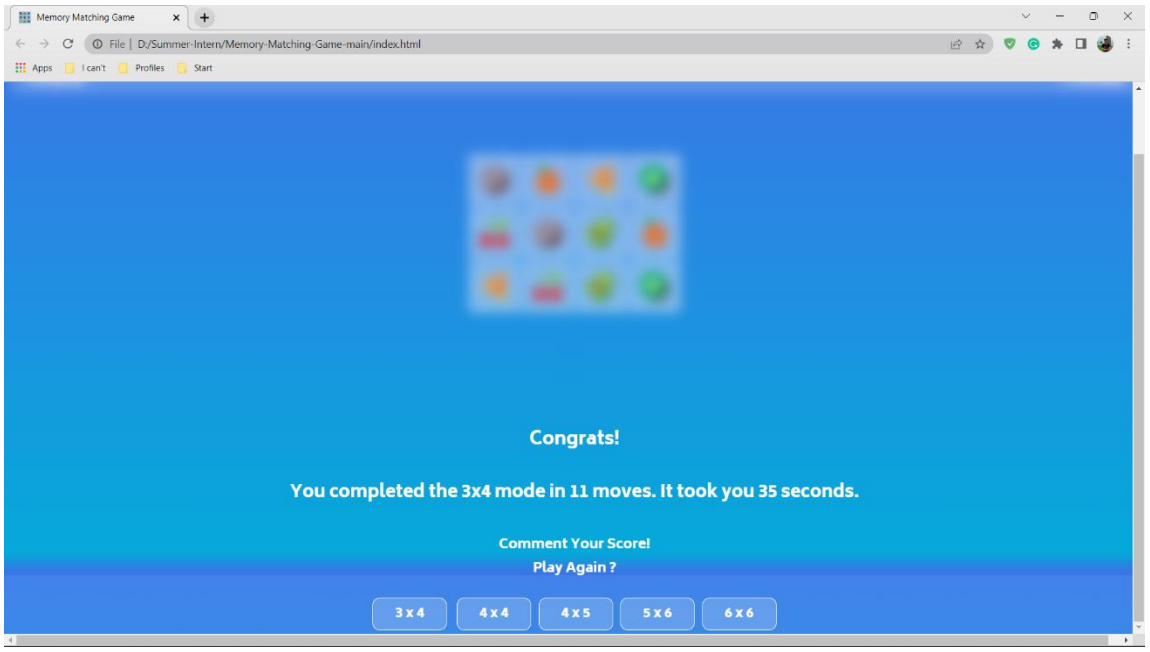


The user plays the game. That is he select the pairs of images and memorize them.



5.2.3. Result

After successfully all the pairs the output page will look like follows. This shows the time taken and number of steps used by the player.



6. SOURCE CODE

6.1.HTML

index.html

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Memory Matching Game</title>
        <link rel="shortcut icon" type="image/jpg" href="Images/fevicon.PNG"/>
        <link rel="stylesheet" href="gameStyle.css">
        <script src="https://code.jquery.com/jquery-3.2.1.min.js"> </script>
        <script src="gameScript.js"> </script>
    </head>
    <body>
        <div id="ol"> </div>
        <div style="height: 8px;"></div>
        <div id="title">
        <span id="logo">MEMORY MATCHING GAME</span>
        </div>

        <div id="title" style="height: 40px;">
            <span id="moves"></span>
            <span id="time"></span>
        </div>

        <center>
            <table cellspacing="0"></table>
        </center>
    </body>
</html>
```

6.2.CSS

index.html ● # gameStyle.css ×

gameStyle.css > .back

```css
@import url('https://fonts.googleapis.com/css2?family=Biryani:wght@800&display=swap');
* {
    font-family: 'Biryani', sans-serif;
}
html {
    width:100vw;
    height:100%;
}
body {
    margin:0px;
    background-image: -webkit-gradient(linear, left top, left bottom, from(#4481eb), to(#04bedd));
    background-image: -o-linear-gradient(top, #4481eb 0%, #04bedd 100%);
    background-image: linear-gradient(to bottom, #4481eb 0%, #04bedd 100%);
}

p {
    font-size: 40px;
    margin-top:5px;
}
td {
    background-color: transparent;
    height:70px;
    width:70px;
}
td, .inner, .front, .back {
    border-radius: 4px;
}
table {
    margin-top: 80px;
}

```

index.html ● # gameStyle.css ×

gameStyle.css > .back

```css

.inner {
    position: relative;
    width: 100%;
    height: 100%;
    text-align: center;
    -webkit-transition: -webkit-transform 0.8s;
            transition: -webkit-transform 0.8s;
         -o-transition: transform 0.8s;
            transition: transform 0.8s;
            transition: transform 0.8s, -webkit-transform 0.8s;
    -webkit-transform-style: preserve-3d;
            transform-style: preserve-3d;
    -webkit-transform: rotateY(0deg);
            transform: rotateY(0deg);
}

.front {
    background-color: rgba(255,255,255,0.3);
}

.back {
    background-color: rgba(255,255,255,0.5);
    -webkit-transform: rotateY(180deg);
            transform: rotateY(180deg);
}

.front, .back {
    position: absolute;
    width: 100%;
    height: 100%;
    -webkit-backface-visibility: hidden;
            backface-visibility: hidden;
}
button:hover, button:active {
    outline:0;
}
```

6.3. JavaScript

index.html ● gameScript.js ×

gameScript.js > ...

```
var em = ["🦂","🌷","🌻","🌼","🌺","🌵","🌈","🌹","🍒","🍎","🍉","🍊","🍋","🍐","🍏","🍈","🍓","🥥","🍅","🌶","🍄","🥦","🥑","🍔","🍕","🍰","🎂","🍖","🍫","🍪"];
//Shuffling above array
var tmp, c, p = em.length;
if(p) while(--p) {
   c = Math.floor(Math.random() * (p + 1));
   tmp = em[c];
   em[c] = em[p];
   em[p] = tmp;
}

//Variables
var pre="", pID, ppID=0, turn=0, t="transform", flip="rotateY(180deg)", flipBack="rotateY(0deg)", time, mode;

//Resizing Screen
window.onresize = init;
function init() {
   W = innerWidth;
   H = innerHeight;
   $('body').height(H+"px");
   $('#ol').height(H+"px");
}

//Showing instructions
window.onload = function() {
    $("#ol").html(`<center><div id="inst"><h3>Welcome !</h3>Instructions For Game<br/><br/><li>Make pairs of similiar blocks by flipping them.</li><li>To flip a block you can click on it.</li><li>If two blocks you clicked are not similar, they will be flipped back.</li><p style="font-size:18px;">Click one of the following mode to start the game.</p></div><button onclick="start(3, 4)">3 x 4</button> <button onclick="start(4, 4)" style="W">4 x 4</button><button onclick="start(4, 5)">4 x 5</button><button onclick="start(5, 6)">5 x 6</button><button onclick="start(6, 6)">6 x 6</button></center>`);
}
```

```
    //If all pairs are matched
    if (rem==0) {
          clearInterval(time);
          if (min==0) {
             time = `${sec} seconds`;
          }
          else {
             time = `${min} minute(s) and ${sec} second(s)`;
          }
          setTimeout(function() {
             $("#ol").html(`<center><div id="iol"><h2>Congrats!</h2><p style="font-size:23px;padding:10px;">You completed the ${mode} mode in ${moves} moves. It took you ${time}.</p><p style="font-size:18px">Comment Your Score!<br/>Play Again ?</p><button onclick="start(3, 4)">3 x 4</button> <button onclick="start(4, 4)" style="W">4 x 4</button><button onclick="start(4, 5)">4 x 5</button><button onclick="start(5, 6)">5 x 6</button><button onclick="start(6, 6)">6 x 6</button></div></center>`);
             $("#ol").fadeIn(750);
          }, 1500);
    }
  }
}
```

7. TESTING

7.1. Introduction:

The cause of testing is to detect mistakes. Making an attempt out is the technique of looking for to realize each viable fault or weakness in a piece product. It is the method of excising program with the intent of constructing certain that the application procedure meets its necessities and client expectations and does no longer fail in an unacceptable process. There are rather plenty of forms of scan. Each experiment sort addresses a special trying out requirement.

7.2. TYPES OF TESTS:

7.2.1. Unit testing:

Unit checking out involves the design of scan circumstances that validate that the Internal application good judgment is functioning safely, and that program inputs produce legitimate outputs. All decision branches and interior code float must be validated. It's the checking out of character application items of the application . It is achieved after the completion of an person unit earlier than integration. It is a structural checking out, that relies on competencies of its construction and is invasive. Unit exams participate in common exams at component level and scan a distinct business approach, utility, and/or process configuration. Unit assessments be certain that every specified course of a industry method performs appropriately to the documented requisites and involves clearly outlined inputs and anticipated results.

7.2.2. Integration testing:

Integration Testing are designed to scan built-in program accessories to determine within the occasion that they evidently run as one software. Trying out is occasion driven and is more concerned with the fundamental final result of screens or fields. Integration assessments reveal that despite the fact that the accessories had been for my part pleasure, as proven through effectively unit checking out, the combo of accessories is correct and regular. Integration checking out is chiefly aimed at exposing the issues that come up from the performance of different components.

7.2.3. Functional testing:

Functional Testing checks provide systematic demonstrations that

capabilities established are to be had as particular by means of the business and technical specifications, method documentation, and consumer manuals. Functional testing is working on below mentioned data:

Legitimate input : identified lessons of legitimate input ought to be accredited.

Invalid enter : recognized lessons of unacceptable effort must be rejected.

Capabilities : recognized features ought to be exercised.

Output : recognized courses of software outputs have got to be exercised.

Systems/Procedures : performance of the system here was invoked Individual and team work of useful checks is fascinated by specifications, key capabilities, or special scan instances. Moreover, systematic insurance plan concerning establish business method flows; data fields, predefined processes, and successive strategies have to be regarded for trying out. Before useful trying out is whole, extra checks are recognized and the strong price of present checks be strong minded.

7.2.4. System testing:

scheme difficult ensure so as to the whole included agenda process meets principles. It exams a pattern to make sure identified and predictable outcome. An illustration of procedure testing is the configuration oriented approach integration scan. System testing is based on approach descriptions and flows, emphasizing pre-driven system links and integration aspects.

7.2.5. White Box Testing:

This testing is a trying out wherein where the application tester has competencies of the interior workings, constitution and software language, or at least its cause. It's rationale. It's used to test areas that can't be reached from a black box stage.

7.2.6. Black Box Testing:

This is testing the software with none advantage of the inside workings, establishment or words of the unit life form veteran. Black field checks, as most other sorts.

7.3. LEVELS OF TESTING

7.3.1. Unit testing strategy:

Unit checking out is most commonly performed as a part of a mixed code and unit experiment part of the software lifecycle, though it be not exceptional for coding and unit checking out to be performed as two targeted phases.

Test strategy and approach:

Field testing out can be carried out manually and sensible assessments shall be written in element.

7.3.2. Integration testing strategy:

Software integration testing is the incremental integration checking out of two otherwise further included software gears on top of a solo stage to fabricate failure induced with the aid of interface defects. The project of the mixing scan is to check that components or program applications, e.g. Components in a program approach or œ one step up œ software purposes at the company degree œ interact without error.

Test Results:

All of the scan circumstances recounted above passed efficiently. No defects encountered.

7.3.3. Acceptance Testing:

User Acceptance testing trying out is a crucial section of any mission and requires enormous participation by the tip user. It additionally ensures that the procedure meets the functional specifications.

Test Results:

The entire test cases recounted above passed effectually. No defects Encountered.

8. CONCLUSION

This project “Memory Matching Game” was successfully developed using HTML, CSS and JavaScript. Compared to existing games, this can be used by both beginners and professionals. Not only a kind of refreshment from a stressful day but this Memory Matching game even helps us to improve our memory power and sharpens our brain.