FULL STACK PROJECT REPORT

**On**

**“Gaming Zone”**

**Submitted by**

**Sakshi Goyal**

**(181500609)**

**Ritika Agrawal**

**(181500581)**

**Priya Rajput**

**(181500507)**

Department of Computer Engineering & Applications

**Institute of Engineering & Technology**



**GLA University**

**Mathura- 281406, INDIA**

**2020**

**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**



**Declaration**

We hereby declare that the work which is being presented in the Full Stack Project “**GAMING ZONE”,** in partial fulfillment of the requirements for Full Stack Project viva voce, is an authentic record of our own work carried by the team members under the supervision of our mentor Mr. Pankaj Kapoor.

Group Members: Sakshi Goyal (181500609)

Priya Rajput (181500507)

Ritika Agrawal (181500581)

Course: B.Tech (Computer Science and Engineering)

Year: 3rd

Semester: 5th

## Supervised By:

Mr. Pankaj Kapoor, Assistant Professor,

GLA University, Department of Computer Engineering & Application

**Department of computer Engineering and Applications**

**GLA University, Mathura**

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,**

**Mathura – 281406**



**Certificate**

This is to certify that the above statements made by the candidates are correct to the best of my/our knowledge and belief.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Supervisor

Mr. Pankaj Kapoor

Technical Trainer

Dept of CEA, GLA University

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Project Coordinator Program Coordinator

(Mr. Mayank Srivastava) (Mr. Shashi Shekar)

**About the Project**

Our full stack project “Gaming Zone” is an online website which consists of different interesting games built with the help of technologies like HTML, CSS and JavaScript. In today’s world, where there is so much of stress, it become really important for us to maintain our mental composure and we really need something through which we can de-stress ourselves and what can be a better option than playing games. Our websites provide simple to play game which relax you and you regain your energy for doing other important daily tasks. Our website is a static website. It doesn’t include any database or any back-end technologies. So. This website has been made with the help of front-end web development. Currently, it includes four games whose names are Guess the color game, Rock Paper and Scissor, WordBeater and Mix or Match. Firstly we get some information about the game, then we can read the instructions i.e., how to play the game, and then we can play the game. Since where’s game, there’s competition, so the opponent in all the games will either be you or the computer system.

**Motivation**

Online gaming is becoming prominent all over the world. This can earn the players a lot of money as long as one has the skills and talent for online gaming. Online gaming is advantageous as it can occupy the mind and avoid people especially youths from getting involved in the life-threatening behavior such as drug taking and joining youth gangs. At least online gaming is more constructive. Online gaming can also help expand and improve gaming knowledge and also experience. Online gaming also encourages competition. One player competes against others playing the same game so as to win more money. Another advantage of online gaming is that it makes a player mentally active and sharper too.

**Future Prospects**

Many more games can be added to the website with the time. We can also give such option where one player can have opponent other than the computer system. So all in all, it is a useful website which can relax you and make your mind sharper.

**Requirements**

**a). Software Requirements:**

* Technology Implemented: Front End Web Development
* Languages Used: HTML, CSS and JavaScript
* IDE Used: Visual Studio Code
* Web Browser: Google Chrome
* GitHub: GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub Repository: A GitHub repository can be used to store a development project. It can contain folders and any type of files (HTML, CSS, JavaScript, Documents, Data, Images). A GitHub repository should also include a license file and a README file about the project. A GitHub repository can also be used to store ideas, or any resources that you want to share.
* Visual Studio Code: Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. [7] 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. Microsoft has released Visual Studio Code's source code on the VS Code repository of GitHub.com, under the permissive MIT License, while the compiled binaries are freeware.

**b). Hardware Requirements:**

* Processor Required: Intel i5
* Operating System: Windows 10
* RAM: 8GB
* Hardware Devices: Computer System
* Hard Disk: 256GB

**Acknowledgement**

We thank the almighty for giving us the courage and perseverance in completing the project. This project itself is an acknowledgement for all those people who have given us their heartfelt co-operation in making this project a grand success. We extend our sincere thanks to Mr. Pankaj Kapoor, Assistant Professor at “GLA University, Mathura” for providing his valuable guidance at every stage of this project work. We are profoundly grateful towards the unmatched services rendered by him. And last but not least, we would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation in doing the main project.

**GAMING ZONE**

**Abstract**

As the name suggests, our project is all about a website consisting number of easy and simple to play games. We have developed this website with the help of languages like HTML, CSS and JavaScript. The visible part which is visible to users is made with the help of HTML and CSS and the functionality to the games is provided by JavaScript. It consists of four games. The first one is game which is very popular i.e., “Rock Paper and Scissor Game”. In this game, you will play against the computer system and whoever wins, his/her score increases by 1. Second game is “guess the color” game in which you will be provided with the color codes and you have to predict which color is that. Third game is WordBeater in which you have to type the word shown on the screen in 5 seconds, if you get successful it will continue, otherwise you lose the game. And the last game is “mix or match” which ask you to choose a theme and then you have to mix and match the cards to win.

We have used an online platform named GitHub to deploy our website.

**Contents**

**Acknowledgment…………………………………….........**08

**Abstract…………………………………………………**09

1. **Introduction:**

Introduction to Web development…......12

Pre-requisites…………………………..13

1. **Technologies Used:**

HTML…………………………………14

CSS…………………….......................15

JavaScript……………………………….......16

Bootstrap........................................................17

**3. List of Figures...……………………………………**19-27

**4. Software Testing..............................................................**28-31

**5.Conclusion……………………………………………**32

**6.Bibliography………………………………………….**33

**Chapter 1**

**Introduction**

**Web Development**

Web development is the work involved in developing a Web site for the Internet (World Wide Web) or an intranet (a private network). Web development can range from developing a simple single static page of plain text to complex Web-based Internet applications (Web apps), electronic businesses, and social network services. A more comprehensive list of tasks to which Web development commonly refers, may include Web engineering, Web design, Web content development, client liaison, client-side/server-side scripting, Web server and network security configuration, and e-commerce development.

Among Web professionals, "Web development" usually refers to the main non-design aspects of building Web sites: writing markup and coding. Web development may use content management systems (CMS) to make content changes easier and available with basic technical skills.

For larger organizations and businesses, Web development teams can consist of hundreds of people (Web developers) and follow standard methods like Agile methodologies while developing Web sites. Smaller organizations may only require a single permanent or contracting developer, or secondary assignment to related job positions such as a graphic designer or information systems technician. Web development may be a collaborative effort between departments rather than the domain of a designated department. There are three kinds of Web developer specialization: front-end developer, back-end developer, and full-stack developer. Front-end developers are responsible for behavior and visuals that run in the user browser, while back-end developers deal with the servers.

**Pre-requisite**

Hands-on knowledge of JavaScript, HTML and CSS is essential before working on the concepts for making of webpages. Make sure that you have the browser or chrome installed and running before opening website.

**Chapter 2**

**Technologies Used**

**HTML**

**Hypertext Markup Language** (**HTML**) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

**CSS**

**Cascading Style Sheets** (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

The name *cascading* comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

CSS saves time − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

• Pages load faster − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

• Easy maintenance − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

• Superior styles to HTML − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

• Multiple Device Compatibility − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing

**JAVASCRIPT**

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web-applications. The vast majority of websites use it for client-side page behavior, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs.

JavaScript engines were originally used only in web browsers, but they are now embedded in some servers, usually via Node.js. They are also embedded in a variety of applications created with frameworks such as Electron and Cordova.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

The use of JavaScript has expanded beyond its web browser roots. JavaScript engines are now embedded in a variety of other software systems, both for server-side website deployments and non-browser applications.

Initial attempts at promoting server-side JavaScript usage were Netscape Enterprise Server and Microsoft's Internet Information Services, but they were small niches. Server-side usage eventually started to grow in the late-2000s, with the creation of Node.js and other approaches.

Electron, Cordova, and other software frameworks have been used to create many applications with behavior implemented in JavaScript. Other non-browser applications include Adobe Acrobat support for scripting PDF documents and GNOME Shell extensions written in JavaScript.

JavaScript has recently begun to appear in some embedded systems, usually by leveraging Node.js.

**BOOTSTRAP**

Bootstrap is a HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

Bootstrap also comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. Each Bootstrap component consists of an HTML structure, CSS declarations, and in some cases accompanying JavaScript code. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields.

The most prominent components of Bootstrap are its layout components, as they affect an entire web page. The basic layout component is called "Container", as every other element in the page is placed in it. Developers can choose between a fixed-width container and a fluid-width container. While the latter always fills the width of the web page, the former uses one of the four predefined fixed widths, depending on the size of the screen showing the page:

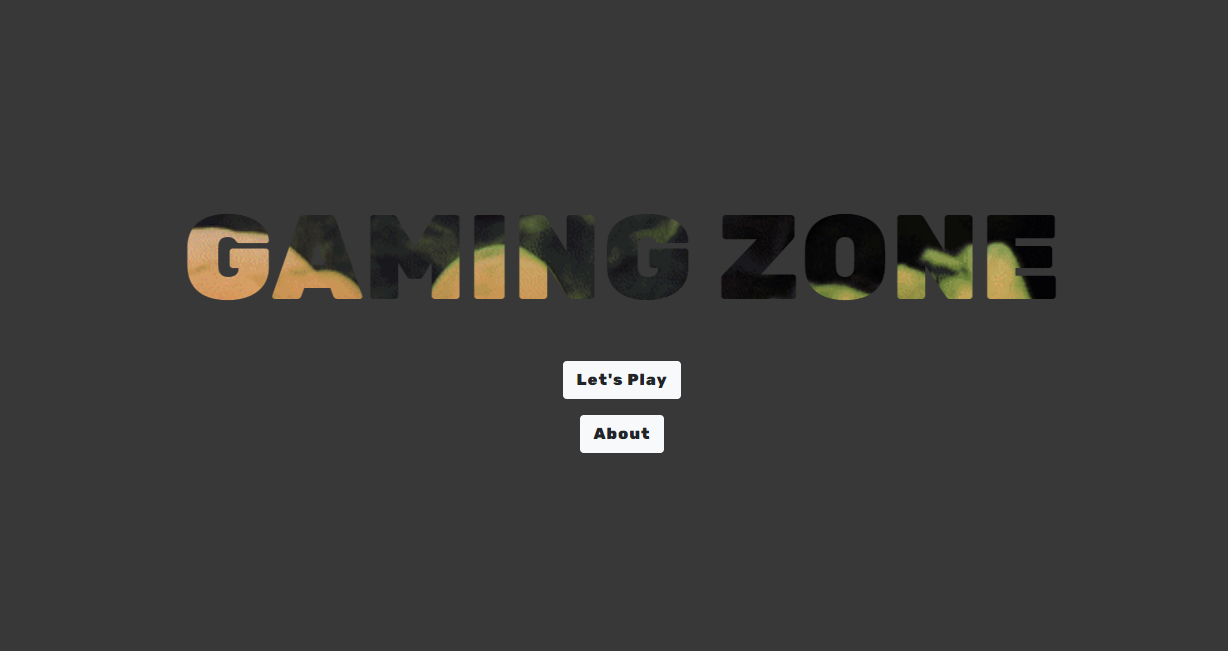
* Smaller than 576 pixels
* 576–768 pixels
* 768–992 pixels
* 992–1200 pixels
* Larger than 1200 pixels

Once a container is in place, other Bootstrap layout components implement a CSS Flexbox layout through defining rows and columns.

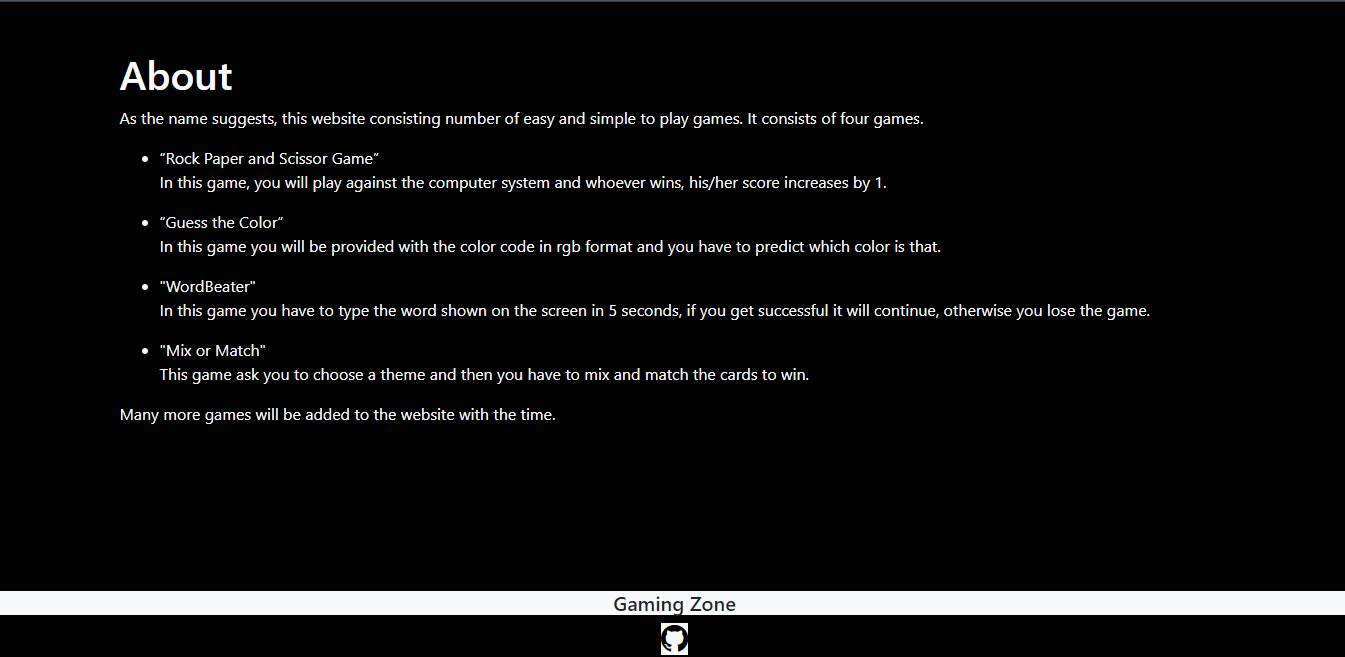
**Chapter 3**

**List of Figures**

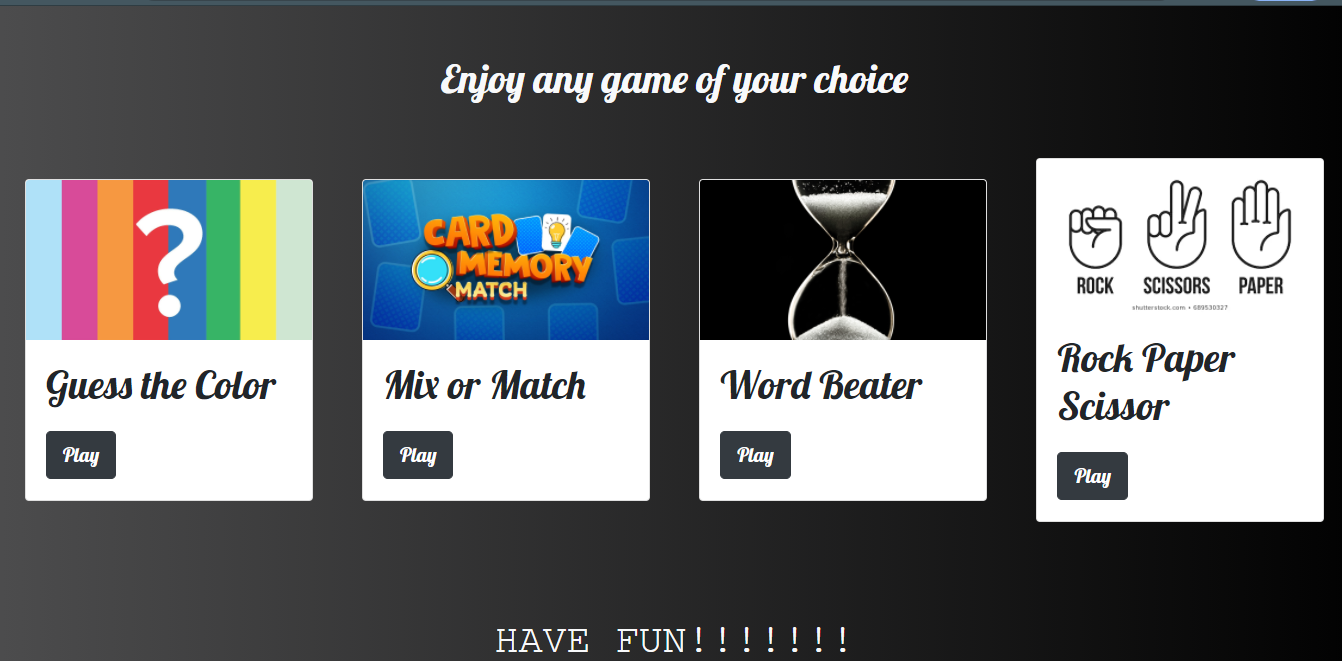
**1. Home Page**

****

**2. About Page**

****

**3. When we click on Let’s Play button, a page containing all the games will come**

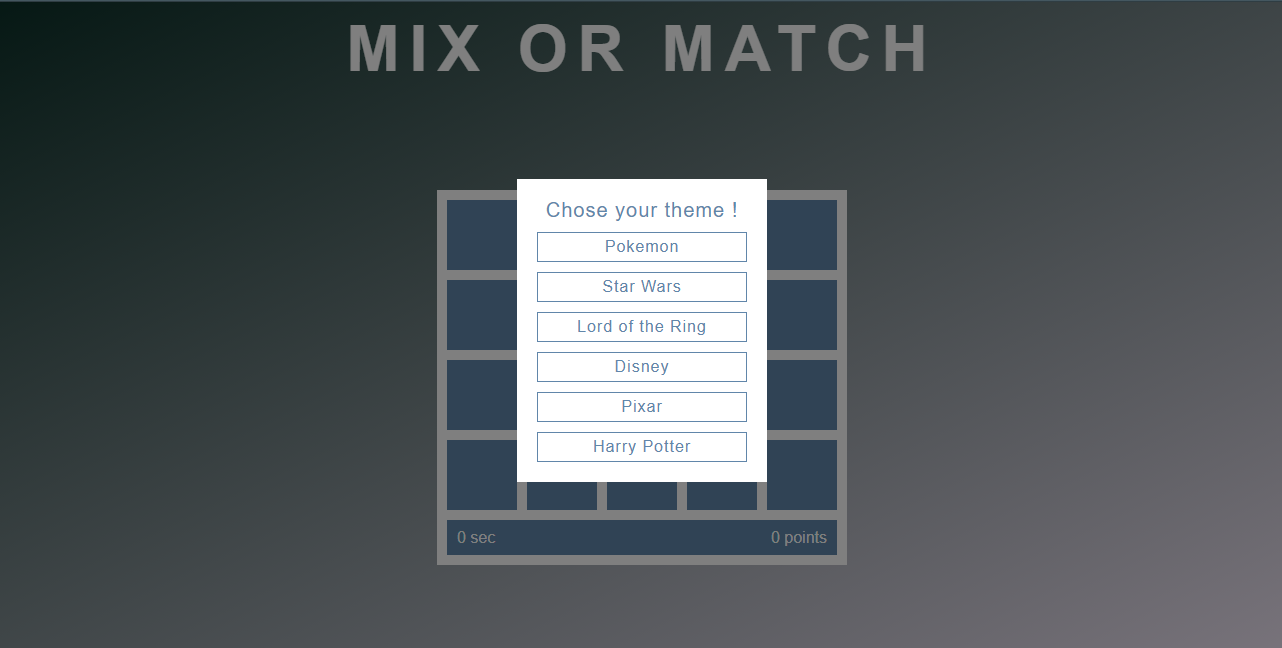
****

**4. Game 1: Guess the color**

****

**5. Game 2: Mix Or Match**

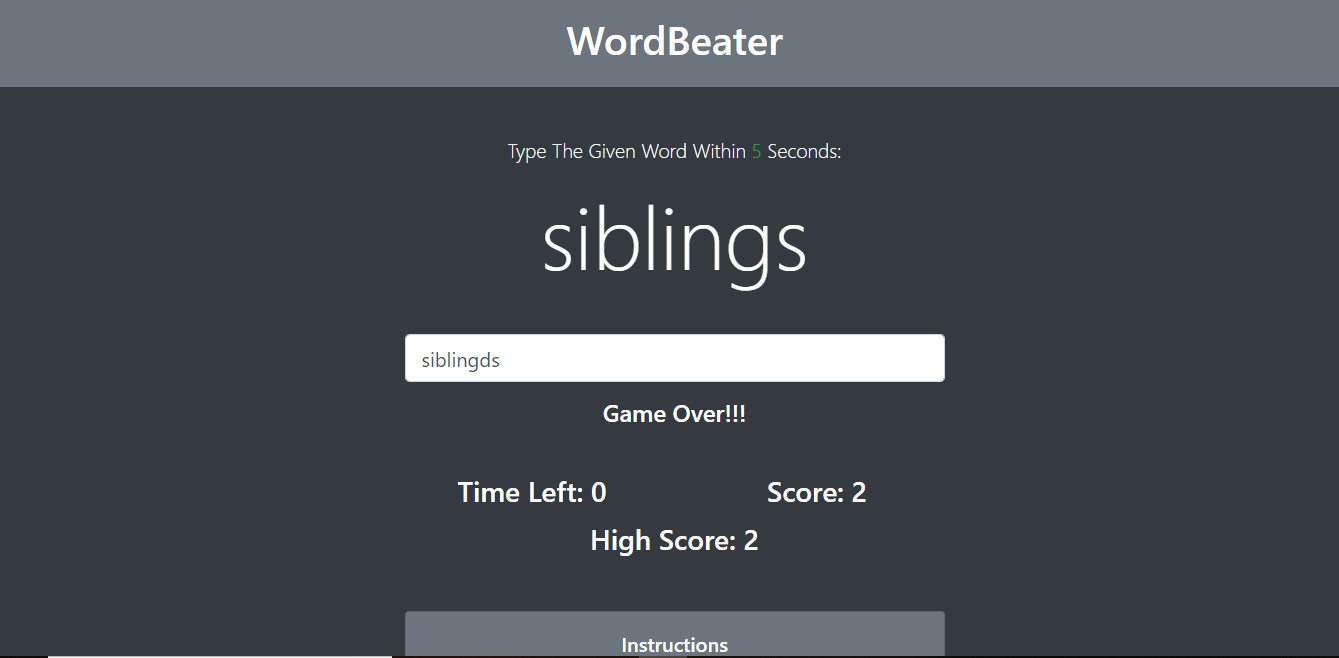
**With entering into this game, it will ask you to choose a theme from given number of themes**



**After choosing any of the theme given, you will be taken to the page and can play the game.**

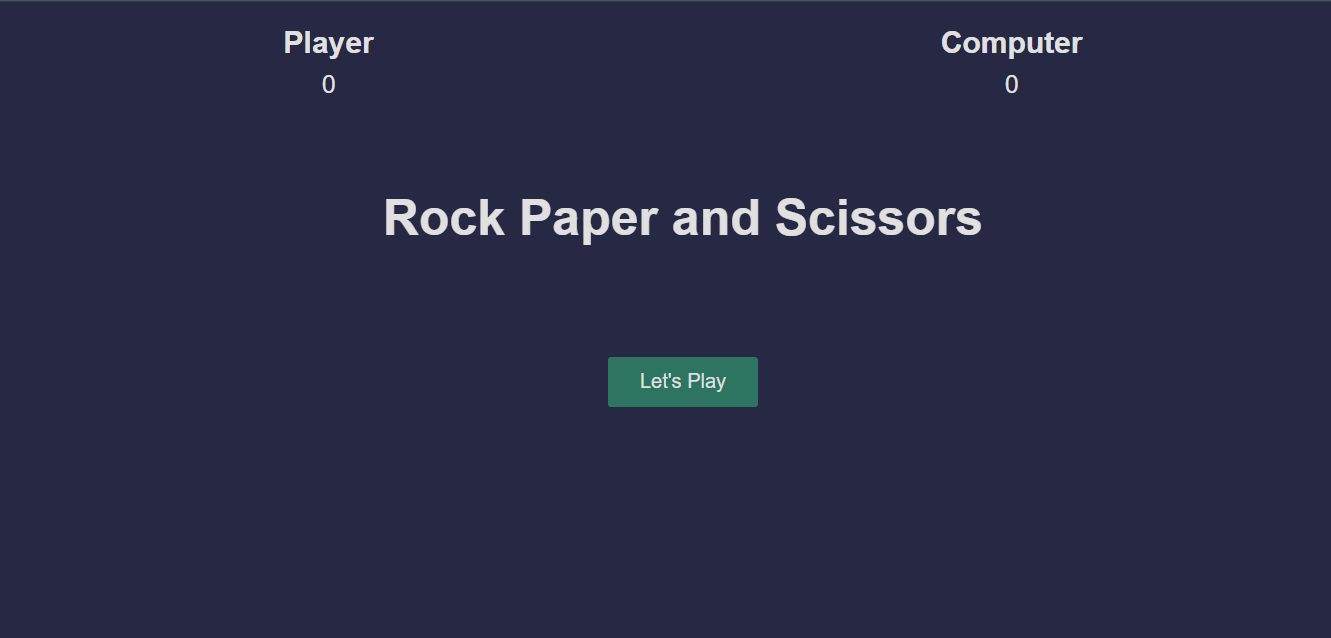
****

**6. Game 3: WordBeater**

****

**6. Game 4: Rock Paper and Scissor**

**Intro Page**

****

**Main Page for the Rock, Paper and Scissor Game**

****

**Chapter 4**

**Software Testing**

Once source code has been generated, software must be tested to uncover as many errors as possible before delivery. It is very important to work the system successfully and achieve high quality of software. Testing include designing a series of test cases that have a high likelihood of finding errors by applying software-testing techniques. System testing makes logical assumptions that if all the parts of the system are correct, the goal will be successfully achieved. The system should be checked logically. Validations and cross checks should be there. Avoid duplications of record that cause redundancy of data. In other Words, Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. It is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

The preliminary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified, and so that debugging, testing and modifications are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmark of good programs, obscurity, cleverness, and complexity are indications of inadequate design and misdirected thinking. Source code clarity is enhanced by structured coding techniques, by good coding style, by, appropriate supporting documents, by good internal comments, and by feature provided in modern programming languages. The implementation team should be provided with a well-defined set of software requirement, an architectural design specification, and a detailed design description. Each team member must understand the objectives of implementation.

4.1 TERMINOLOGY

Error The term error is used in two ways. It refers to the difference between the actual output of software and the correct output, in this interpretation, error is essential a measure of the difference between actual and ideal. Error is also to used to refer to human action that result in software containing a defect or fault.

Fault is a condition that causes to fail in performing its required function. A fault is a basic reason for software malfunction and is synonymous with the commonly used term Bug.

Failure is the inability of a system or component to perform a required function according to its specifications. A software failure occurs if the behavior of the software is the different from the specified behavior. Failure may be caused due to functional or performance reasons.

4.2 TYPES OF TESTING

**a. Unit Testing** The term unit testing comprises the sets of tests performed by an individual programmer prior to integration of the unit into a larger system. A program unit is usually small enough that the programmer who developed it can test it in great detail, and certainly in greater detail than will be possible when the unit is integrated into an evolving software product. In the unit testing the programs are tested separately, independent of each other. Since the check is done at the program level, it is also called program teasing.

**b. Module Testing** A module and encapsulates related component. So can be tested without other system module.

**c. Subsystem Testing** Subsystem testing may be independently design and implemented common problems are sub-system interface mistake in this checking we concenton it. There are four categories of tests that a programmer will typically perform on a program unit.

i Functional test

ii Performance test

iii Stress test

iv Structure test

**Functional Test** Functional test cases involve exercising the code with Nominal input values for which expected results are known; as well as boundary values (minimum values, maximum values and values on and just outside the functional boundaries) and special values.

**Performance Test** Performance testing determines the amount of execution time spent in various parts of the unit, program throughput, response time, and device utilization by the program unit. A certain amount of avoid expending too much effort on fine-tuning of a program unit that contributes little to the overall performance of the entire system. Performance testing is most productive at the subsystem and system levels.

**Stress Test** Stress test are those designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of a program by examining the manner in which a program unit breaks.

**Structure Test** Structure tests are concerned with exercising the internal logic of a program and traversing particular execution paths. Some authors refer collectively to functional performance and stress testing as “black box” testing. While structure testing is referred to as “white box” or “glass box” testing. The major activities in structural testing are deciding which path to exercise, deriving test date to exercise those paths, determining the test coverage criterion to be used, executing the test, and measuring the test coverage achieved when the test cases are exercised.

**Chapter 5**

**Conclusion**

We have completed our project within time limit with the coordination of our team members under the supervision of our mentor Mr. Pankaj Kapoor.

Our website is available at

<https://ritikaagrawal123.github.io/FullStackProject_GamingZone/>

**Chapter 6**

**Bibliography**

[**www.google.com**](http://www.google.com)

**www.geeksforgeeks.org**

[**www.youtube.com**](http://www.youtube.com)

**www.w3schools.com**

**www.beta-labs.in**