

**TRAINING REPORT**

**ON**

**SIMON GAME**

Submitted In Partial Fulfillment of the Requirements

for the Degree of

**Bachelor of Technology**

**In**

**Computer Science & Engineering**

**Submitted By:**

Name: Divyanshu

Roll No.: 8521112

Batch: 2021-2025

**Submitted To:**

**Department Of COMPUTER SCIENCE & ENGINEERING**

Jai Parkash Mukand Lal Innovative Engineering and Technology Institute,

Radaur – 135133 (Yamuna Nagar)

Affiliated to Kurukshetra University, Kurukshetra

**Session:2023-2024**

**CERTIFICATE**



**Acknowledgement**

Apart from the efforts of us, the success of our project depends on the

encouragement and guidelines of many others. We take this opportunity to express

our gratitude to the people who have been instrumental in the completion of the

project .We would like to show our greatest appreciation to our project in-change

Mr. Sriyank Siddharta . We cannot say thank you enough for their tremendous

support and help. We feel motivated and encouraged every time we attended her

classes. Without her encouragement and guidelines this project work would not

have been materialized . We’re highly grateful to Mr. Sriyank Siddharta ,

teacher at Mind luster, for her through guidance one day till end of the

training. he actually laid the ground for conceptual understanding of technologies

and project.

**Divyanshu**

**8521112**

**Declaration**

I, Divyanshu hereby declare that the training report entitled (“FRONT-END DEVELPOMENT”) has not presented as a part of any other academic work to get my degree or certificate except JMIETI Radaur for the fulfilment of the requirement for the degree of Bachelor of COMPUTER SCIENCE & TECHNOLOGY.

**PREFACE**

This basis for this research originally stemmed from my passion for developing my coding skills. As the world moves further into digital age, generating vast amount of data and born digital content, there will be a greater need to access legacy material created with outdated technology. In truth, I could not have achieved my current level of success without a strong support group. First of all, my parents who supported me with love and understanding. And secondly, my committee members, each of whom has provided patient advice and guidance throughout the research process. Thank you all for your unwavering support.

**CONTENT**

1. INTRODUCTION
2. HISTORY
3. TYPES OF WEB DEVELOPMENT
4. FRONT-END WEB DEVELOPMENT
5. FEATURES OF FRONT-END WEB DEVELOPMENT
6. HTML
7. CSS
8. JAVASCRIPT
9. PROJECT SIMON GAME
10. CONCLUSION
11. FUTURE SCOPE
12. REFERENCES

**Introduction of Web development**

**Web development** is the work involved in developing a [website](https://en.wikipedia.org/wiki/Web_site) for the [Internet](https://en.wikipedia.org/wiki/Internet) ([World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web)) or an [intranet](https://en.wikipedia.org/wiki/Intranet) (a private network).[[1]](https://en.wikipedia.org/wiki/Web_development#cite_note-1) Web development can range from developing a simple single [static page](https://en.wikipedia.org/wiki/Static_Web_page) of [plain text](https://en.wikipedia.org/wiki/Plain_text) to complex [web applications](https://en.wikipedia.org/wiki/Web_application), [electronic businesses](https://en.wikipedia.org/wiki/Electronic_business), and [social network services](https://en.wikipedia.org/wiki/Social_network_service). A more comprehensive list of tasks to which Web development commonly refers, may include [Web engineering](https://en.wikipedia.org/wiki/Web_engineering), [Web design](https://en.wikipedia.org/wiki/Web_design), [Web content development](https://en.wikipedia.org/wiki/Web_content_development), client liaison, [client-side](https://en.wikipedia.org/wiki/Client-side_scripting)/[server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting), [Web server](https://en.wikipedia.org/wiki/Web_server) and [network security](https://en.wikipedia.org/wiki/Network_security) configuration, and [e-commerce](https://en.wikipedia.org/wiki/E-commerce) development.

Among Web professionals, "Web development" usually refers to the main non-design aspects of building Web sites: writing [markup](https://en.wikipedia.org/wiki/Markup_language) and [coding](https://en.wikipedia.org/wiki/Computer_programming).[[2]](https://en.wikipedia.org/wiki/Web_development#cite_note-2) Web development may use [content management systems](https://en.wikipedia.org/wiki/Content_management_system) (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](https://en.wikipedia.org/wiki/Web_developer)) and follow standard methods like [Agile methodologies](https://en.wikipedia.org/wiki/Agile_software_development) 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](https://en.wikipedia.org/wiki/Graphic_designer) or [information systems](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Front-end_Web_development), back-end developer, and full-stack developer.[[3]](https://en.wikipedia.org/wiki/Web_development#cite_note-3) Front-end developers are responsible for behavior and visuals that run in the user browser, while back-end developers deal with the servers. Since the commercialization of the Web with [Tim Berners-Lee](https://en.wikipedia.org/wiki/Tim_Berners-Lee)[[4]](https://en.wikipedia.org/wiki/Web_development#cite_note-4) developing the World Wide Web at CERN, the industry has boomed and has become one of the most used technologies ever.

**History of Web development**

The idea of the internet had existed in some form for at least a half a century before it finally became a common household utility in the 1990s. Conceived in the 1980s, the World Wide Web gained significant traction with the introduction of the [Mosaic](https://www.techopedia.com/definition/280/mosaic) browser in 1993. Shortly thereafter, businesses began recognizing the web’s commercial potential, as [network infrastructure](https://www.techopedia.com/definition/16955/network-infrastructure) grew to accommodate what would prove to be a massive influx of online activity. Then the tech bubble grew and burst, the survivors of which (Google, Amazon and the like) went from being key tech influencers to veritable corporate giants within about a decade.

In 1989, [Tim Berners-Lee](https://www.techopedia.com/definition/17012/tim-berners-lee) outlined his concept of a computer platform that could facilitate collaboration among researchers who are based in different parts of the world. This led to the invention of the [Hypertext Markup Language](https://www.techopedia.com/definition/1892/hypertext-markup-language-html) (HTML) in 1990. [Strongly based on the Standard Generalized Markup Language](https://www.w3.org/People/Raggett/book4/ch02.html) (SGML), HTML became the fundamental building block of the World Wide Web, and remains at the core of its coding and infrastructure. The standard enabled coders with the ability to organize web page layouts that could be understood and interacted with over interconnected networks.

The internet had long been around in some form by that point, with the first [wide area network](https://www.techopedia.com/definition/5409/wide-area-network-wan) (WAN) having been established in 1965 and the first local [area network](https://www.techopedia.com/definition/5526/local-area-network-lan) (LAN) around 1983. [Twisted-pair](https://www.techopedia.com/definition/13433/twisted-pair-cable), [coaxial](https://www.techopedia.com/definition/15981/coaxial-cable) and [fiber optic](https://www.techopedia.com/definition/14931/fiber-optic) cable had been in development for many decades, and had already been broadly applied in the field of [telecommunications](https://www.techopedia.com/definition/5570/telecommunications). But when all of these technologies converged with Berners-Lee’s model, the modern web was born. Interest in the technology quickly ramped up, and the first commercial web pages were up and running by the mid-1990s.

Prior to the Mosaic browser, much of the web’s presentation consisted of text and tables. Although not the first browser, Mosaic innovated the format by displaying text together with images in a way that defined a path for the future of web design, and served as the World Wide Web’s [killer app](https://www.techopedia.com/definition/7953/killer-application-killer-app). Experiments ensued in the form and function of web page elements. From menu hierarchies to fonts to color schemes, web design grew into an art form that blended tech savvy with aesthetic sensitivity.

## **The Turn of the Century**

The web advanced a great deal in the years following the tech crash of 2000–2001. During this time, government started to play an increasingly influential role in the web, while concurrently, strong tech companies emerged from the ashes of the big collapse to set the new course for [digital commerce](https://www.techopedia.com/definition/23336/digital-commerce-d-commerce) and culture. And as this newer and more solid foundation was laid, the internet increasingly became the main channel for telecommunications in the modern age.

As hardware improvements cultivated broader networks and greater bandwidth, [web development](https://www.techopedia.com/definition/23889/web-development) responded by enabling designers with an array of [multimedia](https://www.techopedia.com/definition/3118/multimedia) to incorporate into the growing and diversifying art of web presentation. [Cascading Style Sheets](https://www.techopedia.com/definition/26268/cascading-style-sheet-css) afforded web design with new ways to organize and display content. [Flash](https://www.techopedia.com/definition/1991/adobe-flash) video forged a new and entirely unique style of web art and animation, and [video streaming](https://www.techopedia.com/definition/9927/video-streaming) changed the way that people consume motion picture for good. Yet still, with all of these revolutions and progressions in web development – the basic interface and structure of the web page has maintained its integrity and balance of form and function.

**Types of Web development**

## Front-end web development

1. Back-end web development
2. Full Stack web development

## 

**FRONT-END WEB DEVELOPMENT**

Front-end web development, also known as client-side development is the practice of producing HTML, CSS and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.

The objective of designing a site is to ensure that when the users open up the site, they see the information in a format that is easy to read and relevant. This is further complicated by the fact that users now use a large variety of devices with varying screen sizes and resolutions thus forcing the designer to take into consideration these aspects when designing the site. They need to ensure that their site comes up correctly in different browsers (cross-browser), different operating systems (cross-platform) and different devices (cross-device), which requires careful planning on the side of the developer.

**FEATURES OF FRONT-END WEB DEVELOPMENT**

* Optimizing the user experience.
* Using HTML, JavaScript and CSS to bring concepts to life.
* Developing and maintaining the user interface.
* Implementing design on mobile websites.
* Creating tools that improve site interaction regardless of the browser.
* Managing software workflow.
* Following SEO best practices.
* Fixing bugs and testing for usability.

**HTML**

**Introduction:**

HTML (Hyper Text Mark-Up Language) is what is known as a “Mark-Up Language” whose role is to prepare written documents using formatting tags. The tags indicate how the document is presented and how it links to other documents.

The **World Wide Web** (WWW for short), or simply the Web, is the worldwide network formed by all the documents (called “**web pages”**) which are connected to one another by hyperlinks.

Web pages are usually **organized** around a main page, which acts as a hub for browsing other pages with hyperlinks. This group of web pages joined by hyperlinks and centered around a main pages is called a **website.**

The Web is a vast living archive composed of a myriad of web sites, giving people access to web pages that may contain formatted text, images, sounds, video, etc.

**What is the Web?**

The Web is composed of web pages stored on the web servers, which are machines that are constantly connected to the internet and which provide the pages that users request. Every web page, and more generally any online resources, such as images, video, music and animation, is associated with a unique address called a URL. The key element for viewing web pages is the **browser,** a software program which sends requests to web servers, then processes the resulting data and displays the information as intended, based on instructions in the HTML page.

The most commonly used browsers on the Internet include:

* Mozilla Firefox,
* Microsoft Internet Explorer,
* Netscape Navigator,
* Safari,
* Opera

**Versions of HTML**

HTML was designed by Tim Berners-Lee, at the time a researcher at CERN (Chinese Ecosystem Research Network), beginning in 1989. He officially announced the creation of the web on Usenet in August 1991. However, it wasn’t until 1993 that HTML was considered advanced enough to call it a language (HTML was then symbolically christened HTM 1.0)

RFC 1866, dated November 1995, represented the first official version of HTML, called HTML 2.0. After the brief appearance of HTML 3.0, which was never officially released, HTML 3.2 became the official standard on January 14, 1997. The most significant changes to HTML 3.2 were the standardization of tables, as well as many features relating to the presentation of web pages.

On December 18, 1997, HTML 4.0 was released. Version 4.0 of HTML was notable for standardizing style sheets and frames. HTML version 4.01, which came out on December 24, 1999, made several minor modifications to HTML 4.0.

**Example-**

<HTML>

<HEAD>

</HEAD>

<BODY>

<H5>THIS IS AN EXAPLE</H5>

</BODY>

</HTML>

**Basic HTML structure**

HTML is composed of elements. These elements structure the webpage and define its content.

A tag and the content between it is called an HTML element.

 Let’s quickly review each part of the element :

* HTML element (or simply, element) — a unit of content in an HTML document formed by HTML tags and the text or media it contains.
* HTML Tag — the element name, surrounded by an opening (<) and closing (>) angle bracket.
* Opening Tag — the first HTML tag used to start an HTML element. The tag type is surrounded by opening and closing angle brackets.
* Content — The information (text or other elements) contained between the opening and closing tags of an HTML element.
* Closing tag — the second HTML tag used to end an HTML element. Closing tags have a forward slash (/) inside of them, directly after the left angle bracket.

Our first html code :-

**<html>**

**<head>**

**<title>**

**Document**

**</title>**

**</head>**

**<body>**

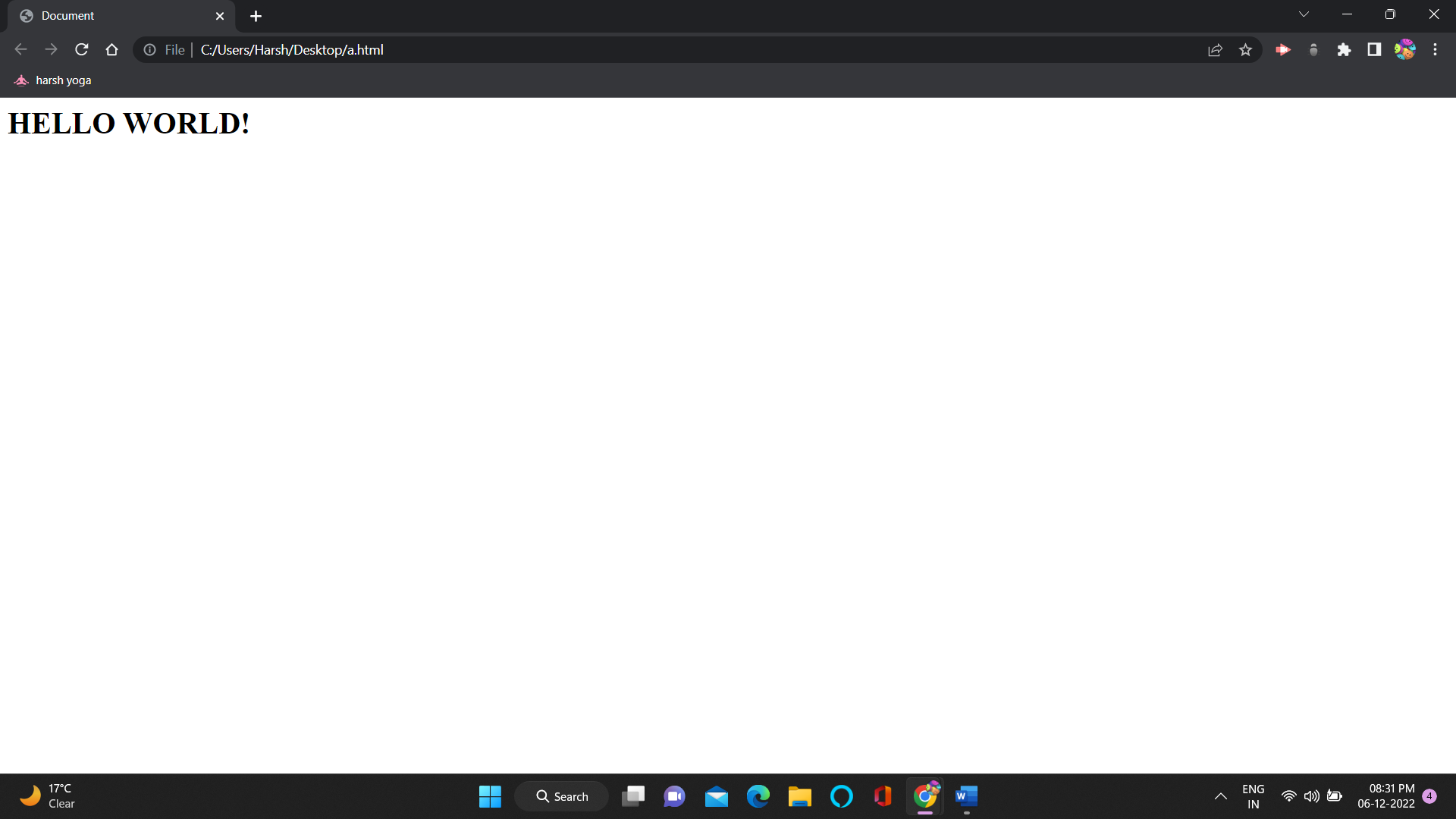
**<h1>**

**HELLO WORLD!**

**</h1>**

**</body>**

**</html>**



**Explanation:**

**HTML Tag**

Anything between <html> and </html> will be considered HTML code. Without these tags, it's possible that browsers could incorrectly interpret your HTML code and present HTML content in unexpected ways.

**Head & Title Tag**

Let's also give the browser some information about the page. We can do this by adding a <head> element.

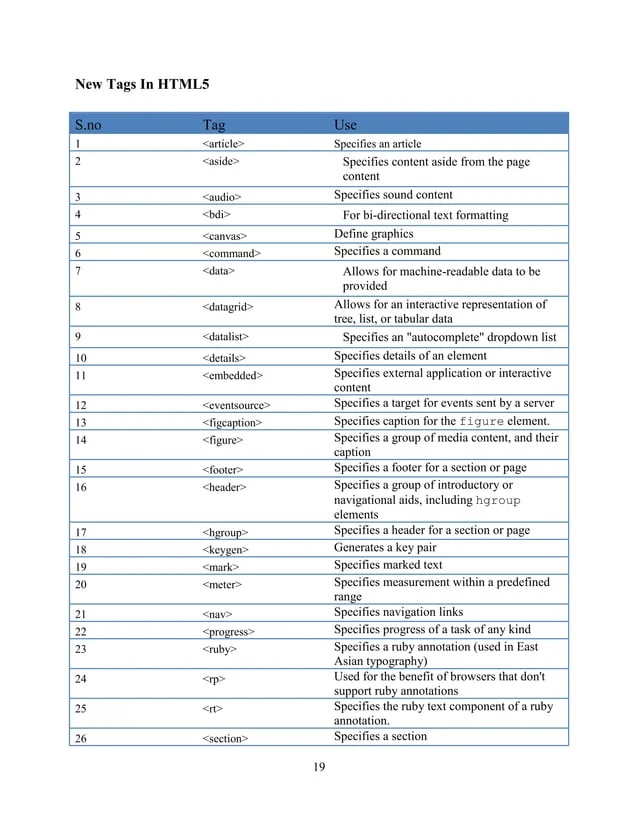
The <head> element will contain information about the page that isn't displayed directly on the actual web page i,e title of the page

The browser displays the title of the page because the title can be specified directly inside of the <head> element, by using a <title> element.

**Body Tag**

Before we can add content, we have to add a body to the HTML file. Once the file has a body, many different types of content can be added within the body, like text, images, buttons, and much more.

<body> </body>

****

**CSS**

**What is CSS?**

* CSS stands for Cascading Style Sheets.
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files.
* CSS describes how HTML elements should be displayed.
* CSS Saves a Lot of Work! The style definitions are normally saved in external .css files.
* With an external stylesheet file, we can change the look of an entire website by changing just one file!
* CSS can be either external or internal.

**CSS Syntax:**

A CSS rule-set consists of a selector and a declaration block:

CSS selector:

The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

Each declaration includes a CSS property name and a value, separated by a colon.

A CSS declaration always ends with a semicolon, and declaration bolocks are surrounded by curly braces.

The **External CSS** can be declared in the required HTML page as:

<link rel = ” stylesheet ” href = ” CSS\_file\_name.css ”>

The external CSS file is saved by using the .css extension, whereas the internal CSS is saved in corresponding HTML file using the **<style>** tag. Using External CSS is much better than using Internal. **Here are a few reasons this is better.**

* Easier Maintenance
* Reduced File Size
* Reduced Bandwidth
* Improved Flexibility

The selectors that can be used to select the HTML part are-

* Id selector
* Class selector

**Id Selector:**

The id selector uses the id attribute of an HTML element to select a specific element. The id of an element should be unique within a page, so the id selector is used to select one unique element! To select an element with a specific id, write a hash (#) character, followed by id of the element. The style rule below will be applied to the HTML element with id=” para1”:

**Example-**

Suppose the HTML content is as follow,

<h1 id=”para1”>content </h1>

Then Id will be declared as

# para1{

text - align: center;

color : blue;

font - family: jokerman;

}

**The class Selector:**

The class selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the

Class.

**Example-**

. para1

{

text - align: center;

color : blue;

font-family : jokerman;

}

**CSS Comments:**

Comments are used to explain the code, and may help when you edit the source code at a later date. Comments are ignored by browsers. A CSS comment starts with /\* and ends with /\*. Comments can also span multiple lines.

**Example-**

.para1{

Text-align: center;

Color; blue;

Font-family: jokerman; /\*this is the single line comment \*/

}

In the example above, all HTML elements with class=para1 will be blue and center – aligned.

**CSS Style:**

* Background properties
* Border properties
* Padding
* Margin
* Color
* Font properties
* Text properties, link properties/ navigation bar properties

1. CSS3 Borders

CSS3 contain following border properties

* Border-radius
* Box-shadow
* Border-image

With CSS3, you can create rounded borders, add shadow to boxes, and use an image as a border without using a design program, like Photoshop. The CSS3 border-radius property allows web developers to easily utilise rounder corners in their design elements. without the need for comer images or the use of multiple div tags, and is perhaps one of the most talked about aspects of CSS3.

Since first being announced in 2005 the border-radius property has come to enjoy widespread browser support (although with some discrepancies) and, with relative ease of use, web developers have been quick to make the most Of this emergtng technology.

1. CSS3 Background-effects

CSS3 contain following border properties

* Border-radius
* Box-shadow
* Border-image

CSS3 contains several new background properties, which allow greater control of the background element. CSS3 allows web designers to specify multiple background images for box element, using nothing more than a simple comma-separated list. The property adds new functionality to CSS allowing designers to specify the size of background images using either lengths, percentages, or by using one of two keywords; contain or cover.

1. CSS3 Text Effects

CSS3 contain following Text Effect properties:

* text-shadow
* word-wrap

The new CSS3 properties give developers a wonderful chance to enhance their designs in a way that’s quick and easy, yet visually impressive. To give a website a visually impressive look, designers put emphasis on neat and stylish typography. For years designers have depended on Photoshop, but CSS3 is a revolution with easy-to-create text effects. Almost all of the major browsers now support most of the CSS3 features so that’s another reason for mastering the new techniques.

1. CSS3 Fonts

Before CSS3, web designers had to use fonts that were already installed on the user's computer. With CSS3, web designers can use whatever font he/she likes. When you have found/bought the font you wish to use, include the font file on your web server, and it will be automatically downloaded to the user when needed,

1. CSS3 2D Transforms

A transform is an effect that lets an clement change shape, size and position,

CSS3 contain following 2D Transforms properties:

* translate()
* rotate()

1. CSS3 Animations

With CSS3, we can create animations, which can replace animated images, Flash animations, and Java Scripts in many web pages

**JAVASCRIPT**

**What is JavaScript?**

JavaScript is an object-based scripting language that is lightweight and cross-platform. JavaScript is not compiled by translated. The JavaScript translator (embedded in browser) is responsible to translate the JavaScript code.

It is mainly used for

* Client- side validation
* Dynamic drop- down menus.
* Displaying data and time.
* Displaying popup windows and dialog boxes(like alert dialog box, confirm dialog box and prompt dialog box).
* Displaying clocks etc.

**Example of JavaScript-**

<h2>Welcome to JavaScript document</h2>

<script>

document.write(“ hello JavaScript by JavaScript”);

</script>

Here, <script> tag is used to initialize the script and document.write() is a function used to write. Like CSS, JavaScript also can be placed in:

* Between the body tag of html
* In .js file (external javascript)
* Between the head tag of html

**JavaScript Example: code between the body tag-**

In the given example, we have displayed the dynamic content using JavaScript. Let’s see the simple example of JavaScript that displays alert dialog box.

<script type=”text/javascript”>

alert(“hello JavaScript”);

</script>

**JavaScript Example: code in .JS file-**

* **Message.js file**

Function msg()

{

alert(“Hello JavaScript”);

}

* **Index.html**

<head>

<script type=”text/javascript” src=”message.js”></script>

</head>

<body>

<p> Welcome to JavaScript</p>

<form>

<input type=”button” value=”click” onclick=”msg()”/>

</form>

</body>

We can create external JavaScript file and embed it in many html page.

It provides **code re usability** because single JavaScript file can be used in several html pages. An external JavaScript file must be saved by .js extension. It is reconnected to embed all JavaScript files into a single file. It increases the speed of the webpage.

**Between the head tag of html**

In the example given below, we are having a function msg() which is called. To create a function, we use function name with keyword **function.** For function call, we need to have an event.

**Example-**

<head>

<script type=”text/javascript”>

function msg()

{

alert(“Hello JavaScript”);

}

</script>

</head>

<body>

<p> Welcome to JavaScript</p>

<form>

<input type=”button” value=”click” onclick=”msg()”/>

</form>

</body>

**How to Change Content of HTML using a JavaScript?**

One of many JavaScript HTML methods is **getElementById()**

This example uses the method to “find” an HTML element (with id=”demo”) and change the element content (**innerHTML**) to “Hello JavaScript”.

**Example-**

document.getElementById(“demo”).innerHTML = “Hello JavaScript”;

document.getElementById(“demo”).style.fontSize=”25px”;

<html>

<head>

<script>

Function myFunction()

{

Document.getElementById(“demo”).innerHTML = “Paragraph changed.”;

}

</script>

</head>

<body>

<h1> My web Page</h1>

<p id=”demo”> A Paragraph</p>

<button type=”button” onclick=”myFunction()”>Try it </button>

</body>

</html>

**Comments in JavaScript:**

The **JavaScript comments** are meaningful way to deliver message. It is used to add information about the code, warning or suggestions so that end user can easily interpret the code. The JavaScript comment is ignored by the JavaScript engine i.e. embedded in in the browser.

**Advantages of JavaScript comments:**

There are mainly two advantages of JavaScript comments

* **To make code easy to understand:** It can be used to elaborate the code so that end user can easily understand the code.
* **To make code easy to understand:** It can be used to elaborate the code so that end user can easily understand the code.
* **To avoid the unnecessary code:** It can be used to avoid the code being executed. Sometimes, we add the code to perform some action. But after sometimes, there may be need to disable the code. In such case, it is better to use comments.

**Example-**

<script type=”text/javascript”>

Function msg()

{

Alert(“Hello Javatpoint”); /\***this is a comment\*/**

**}**

**<**script>

**JavaScript Variable:**

A **JavaScript variable** is simply a name of storage location. There are two types of variable in JavaScript: local variable and global variable. There are some rules while declaring a JavaScript variable (also known as identifiers).

* Name must start with a letter (a to z or A to Z), underscore ( \_ ), or dollar( $) sign.
* After first letter we can digits ( 0 to 9), for example value 1.

JavaScript variables are case sensitive, for example x and X are different variables.

**JavaScript Form Validation:**

It is important to validate the form submitted by the user because it can have inappropriate values. So validation is must.

The JavaScript provides you the facility the validate the form on the client side so processing will be fast than server- side validation. So, most of the web developers prefer JavaScript form validation.

Through JavaScript, we can validate name, password, email, date, mobile number etc. fields .

**Example-**

<script>

function validateform()

{

var name = document.myform.name.value;

var password = document.myform.password.value;

if ( name== null || name== “ ”)

{

alert(“ Name can’t be blank”);

return false;

}

else if ( password.length<6)

{

alert(“ Password must be at least 6 characters long.”);

return false;

}

}

</script>

<body>

<form name= “myform” method= “post” action= “abc.jsp” onsubmit= “return validateform()” >

Name: <input type= “text” name= “name”><br/>

Password: <input type= “password” name= “password”><br/>

<input type= “submit” value= “register”>

</form>

</body>

In this example, we are going to validate the name and password. The name can’t be empty and password can’t be less than 6 characters long. Here, we are validating the form on form submit. The user will not be forwarded to the next page until given values are correct.

**JavaScript Functions:**

**JavaScript functions** are used to perform operations. We can call JavaScript function many times to reuse the code.

**Advantages of JavaScript Function:**

There are mainly two advantages of JavaScript functions.

* Code reusability
* Less coding

**JavaScript Function Syntax**

The syntax of declaring function is given below.

function functionName( [arg1, arg2,….argN])

{

//code to be executed

}

JavaScript Functions can have 0 or more arguments.

**Example-**

<script>

function msg()

**{**

alert( “hello! This is message”);

}

</script>

<input type= “button” onclick = “msg()” value= “call function”/>

**Output of the above example:**

Hello! This is message

**JavaScript Control Statements:**

**If – else:**

It evaluates the content whether condition is true or false. The syntax of JavaScript if – else statement is given below

If(expression)

{

//content to be evaluated if condition is true

}

**e**lse

{

//content to be evaluated if condition is false

}

**Example-**

**<**script>

var a=20;

If (a%2==0)

{

document.write( “a is even number”);

}

else

{

document.write(“ a is odd number”);

}

</script>

**JavaScript Switch:**

The **JavaScript switch statement** is used to execute one code from multiple expressions. It is just like else if statement that we have learned in previous page. But it is convenient than if..else..if because it can be used with numbers, characters etc. The signature of JavaScript switch statement is given below.

switch (expression){

case value 1:

code to be executed;

break;

case value 2:

code to be executed;

break;

default: code to be executed if above values are not matched;

}

**PROJECT OF FRONT-END WEB DEVELOPMENT**

**SIMON GAME**

This is a simple Simon game using HTML, CSS, JS, JQuery, Bootstrap. Beginners can use this as a small project to boost their web developing skills and understanding logic.

1. The [Chrome](https://docs.python.org/3/library/turtle.html#module-turtle) is a extended browser which supports the web developing.
2. **The Game:** Here, a random number is generated and one of the coloured boxes is automatically pressed on the screen for which we have to press later on, as it is. The level starts from zero.
3. We have to remember the sequence of the pressed boxes and in the same order we also have to press the boxes. Simultaneously, each time we hit the correct sequence, the level increases by 1.
4. If we break the sequence and press the other one then the game gets over with an alert sound.

**SOURCE CODE OF PROJECT**

**(Index.html file)**

<!DOCTYPE html>

<html lang="en" dir="ltr">

<head>

<meta charset="utf-8">

<title>Simon</title>

<link rel="stylesheet" href="styles.css">

<link href="https://fonts.googleapis.com/css?family=Press+Start+2P" rel="stylesheet">

</head>

<body class = "mainBody">

<h1 id="level-title">Press A Key to Start</h1>

<div class="container">

<div lass="row">

<div type="button" id="green" class="btn green">

</div>

<div type="button" id="red" class="btn red">

</div>

</div>

<div class="row">

<div type="button" id="yellow" class="btn yellow">

</div>

<div type="button" id="blue" class="btn blue">

</div>

</div>

</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="index.js" charset="utf-8"></script>

</body>

</html

**(Styles.css file)**

body {

text-align: center;

background-color: #011F3F;

}

#level-title {

font-family: 'Press Start 2P', cursive;

font-size: 3rem;

margin: 5%;

color: #FEF2BF;

}

.container {

display: block;

width: 50%;

margin: auto;

}

.btn {

margin: 25px;

display: inline-block;

height: 200px;

width: 200px;

border: 10px solid black;

border-radius: 20%;

}

.game-over {

background-color: red;

opacity: 0.8;

}

.red {

background-color: red;

}

.green {

background-color: green;

}

.blue {

background-color: blue;

}

.yellow {

background-color: yellow;

}

.pressed {

box-shadow: 0 0 20px white;

background-color: grey;

}

**(Index.js file)**

var buttonColors = ["red", "blue", "green", "yellow"];

var gamePattern = [];

var userClickedPattern = [];

var level = 0;

var started = 0;

$(document).keypress(function(){

if(started === 0){

nextSequence();

started = 1;

console.log("in keypress");}

});

$(".btn").on("click", function(){

var userChosenColour = this.id;

userClickedPattern.push(userChosenColour);

animatePress(userChosenColour);

playSound(userChosenColour);

checkAnswer(userClickedPattern.length - 1);

console.log(userClickedPattern);

});

function checkAnswer(currentLevel){

if(userClickedPattern[currentLevel] === gamePattern[currentLevel]){

console.log("success");

if(userClickedPattern.length == gamePattern.length){

setTimeout(function(){

nextSequence();

}, 1500);

}}

else{

playSound("wrong");

setTimeout(function(){

$("body").toggleClass("game-over");

}, 200);

$("h1").text("Game Over, Press Any Key to Restart");

$("body").toggleClass("game-over");

startOver();

}

}

function nextSequence(){

userClickedPattern = [];

level++;

$("#level-title").text("Level " + level);

var randomNumber = Math.floor((Math.random())\*4);

var randomChosenColour = buttonColors[randomNumber];

gamePattern.push(randomChosenColour);

console.log("in nextSequence");

playSound(randomChosenColour);

$("#"+randomChosenColour).fadeIn(100).fadeOut(100).fadeIn(100);

}

function playSound(name){

var makeSound = new Audio("sounds/" + name + ".mp3");

makeSound.play();

}

function animatePress(currentColor){

setTimeout(function(){

$("." + currentColor).toggleClass("pressed")

}, 100);

$("." + currentColor).toggleClass("pressed");

}

function startOver(){

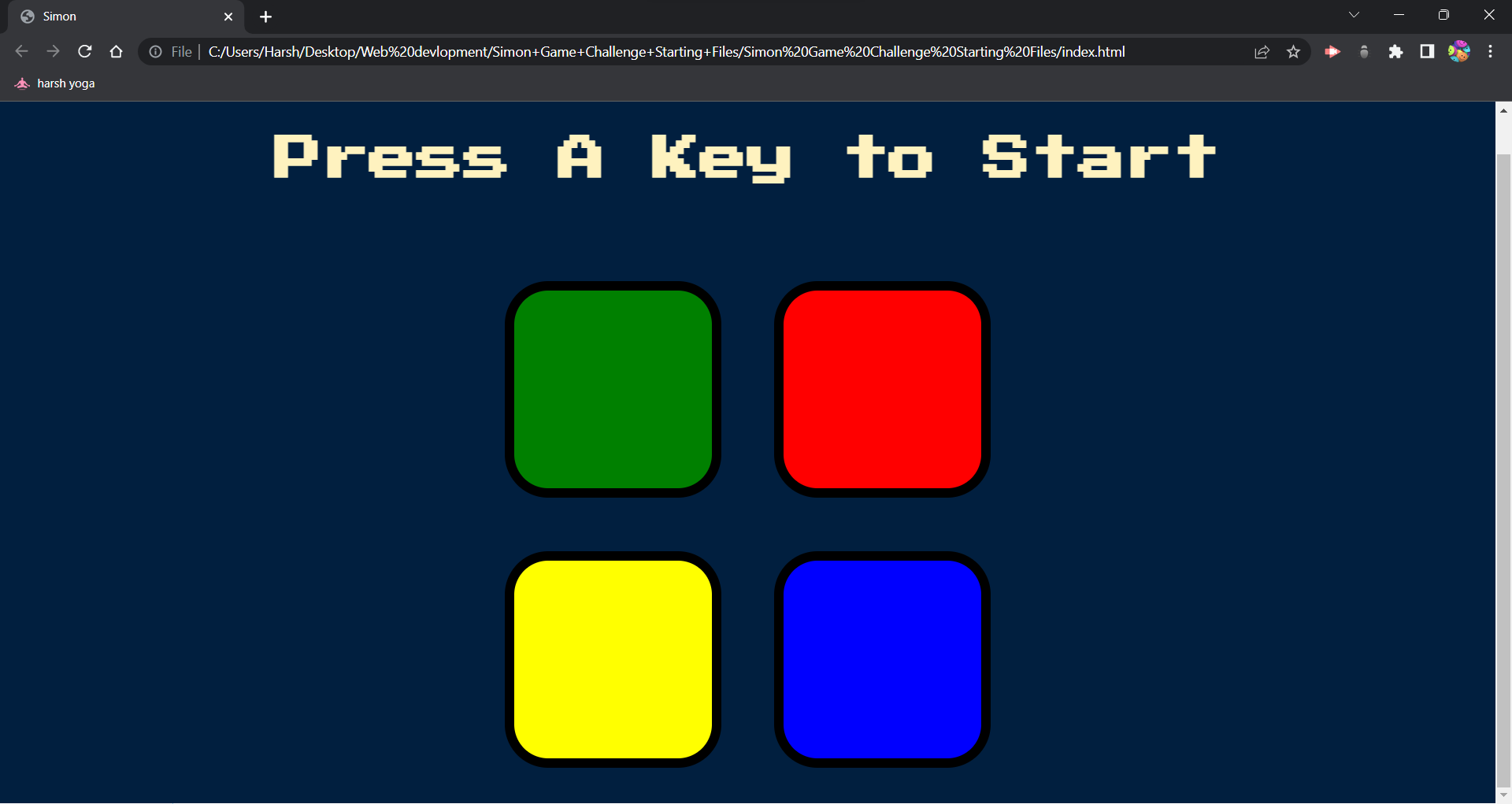
started = 0;

level = 0 ;

gamePattern = [];

}

**OUTPUT OF PROJECT**



**The Game:** Here, a random number is generated and one of the coloured boxes is automatically pressed on the screen for which we have to press later on, as it is. The level starts from zero.



We have to remember the sequence of the pressed boxes and in the same order we also have to press the boxes. Simultaneously, each time we hit the correct sequence, the level increases by 1.



If we break the sequence and press the other one then the game gets over with an alert sound.

**CONCLUSION**

In this course I have experienced a joy of coding, developing and explore tremendous world of software engineering and also it has made my logical skills to reach new heights.

Front-end web development is just basic to start your way onto the web development, several games and web applications can be made from Front-end web development like I have made a game named Simon Game which is very interesting and joy to play I will try to make few more interesting games with the help of web development and explore more about full stack development.

**FUTURE SCOPE**

A Front-End Developer is responsible for developing new user-facing features, determining the structure and design of web pages, building reusable codes, optimizing page loading times, and using a variety of markup languages to create the web pages.

**REFERENCES**

1. Teacher Mr. Sriyank Siddharta
2. Wikipedia
3. W3schools.com
4. developer.mozilla.org
5. devdocs.io