

## MINI PROJECT

(2019-2020)

# GAMING CONTROL USING GESTURES

INSTITUTE OF ENGINEERING AND  
APPLICATIONS



### Submitted To:

Mr.Sharad Gupta

(Assistant Professor)

### Submitted By:

Nitya Nagpal(171500210)

Kirti Verma(171500162)

## **ACKNOWLEDGEMENT**

We would like to express my special thanks of gratitude to my supervisors Mr. Sharad Gupta for providing their invaluable guidance, comments and suggestions throughout the course of the project. I would specially thank him for constantly motivating me to work harder.

NITYA NAGPAL(171500210)

KIRTI VERMA(171500162)

## ABSTRACT

*Hand gesture is an innovative technique of interaction between human and computer. As compared to existing techniques, hand gesture technique has the advantage of being easy to use. By using this technique the traditional way of using mouse and keyboard will be change because one can then interact with the computer with hand gestures. In this technique, ultrasonic sensor is used to classify the hand movement in real-time. Ultrasonic sensor measures the distance of hand by using sound waves. The main idea of our approach is to speed up the interaction with computer, using general purpose hardware like personal computer, Node MCU and low cost sensor like ultrasonic sensor. In this way any user can easily interact with the computer using hand gestures.*

**KEYWORDS** — Hand Gesture, ultrasonic sensor, Node MCU, Python



**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 Mini Project “Steganography”, in complete fulfilment of the requirements for Mini Project viva voce, is an authentic record of our own work carried under the supervision of Mr. **Pankaj Sharma, Assistant Professor, GLA University, Mathura.**

Nitya Nagpal (171500210)

Sign: \_\_\_\_\_

Kirti Verma (171500169)

Sign: \_\_\_\_\_

Course: B. Tech (CSE)

Year: 3rd

Semester: V



**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 project entitled “Steganography” carried out in mini project Lab is bonafide work done by Nitya Nagpal, Kirti Verma and is submitted in complete fulfilment of the requirements for the award of degree Bachelor of Technology(Computer Science and Engineering).

Signature of Supervisor:

Name of Supervisor: Sharad Gupta

Date:

## CONTENTS

- INTRODUCTION
- PROPOSED WORK
- INTRODUCTION TO ULTRASONIC SENSOR AND APPLICATIONS
- IOT IMPLEMENTATION
- COMPONENTS REQUIRED
- CIRCUIT DIAGRAM
- ARDUINO CODE
- INTRODUCTION TO PYAUTOGUI MODULE
- PYTHON CODE
- INTRODUCTION TO JAVASCRIPT
- INTRODUCTION TO HTML
- ABOUT THE GAME
- DISCRIPTION AND CODE
- REFERENCES

## **INTRODUCTION**

Today's world is a high technological world; imagining life without computer is impossible. Now computer is an essential part of human's day-to-day life. Vision and gesture are the important approach for communication between human beings just in the same way as keyboard and mouse play a role for interacting with computer. Numbers of effective techniques are available for interaction with computer and one of them is hand gesture technique.

In that technique hand gesture is used as an input which replaces functionality of mouse and keyboard shortcut keys. Hand gesture is an attractive and faster technique. For determining user hand distance ultrasonic sensors are used. By using hand gesture user can communicate with computer easily and there is no need of any physical connection between user and system.

## **PROPOSED WORK**

This paper introduces a technique based on determining distance by the sensor and accordingly a particular function is performed. Some recognition method of the gestures are proposed and then actions are recognized using sensor. We set up few mainstream methods based on the action recognition by the sensors. The sensor device is attached on computer at head of the screen, for quick operation. In this field much research work has been done but that work is related to hand recognition, real time finger recognition and recognition of alphabet characters. Real time human computer interaction using hand gesture, are also used for many functionalities such as video control, music player, gaming, controlling the functions of PDF reader etc. All these interactions have real time gesture recognition techniques. A gesture controller resolution always requires a physical device which follows and recognizes the body language or movements, so that the computer can clarify them. By using ultrasonic sensor, the distance of hand can be found which acts as an input. According to the distance of hand, particular function is performed.

Our project is divided in three parts:

1. Website Implementation
2. Game implementation and code
3. IOT Implementation

# **IOT IMPLEMENTATION**

## **Introduction to ULTRASONIC SENSOR:**

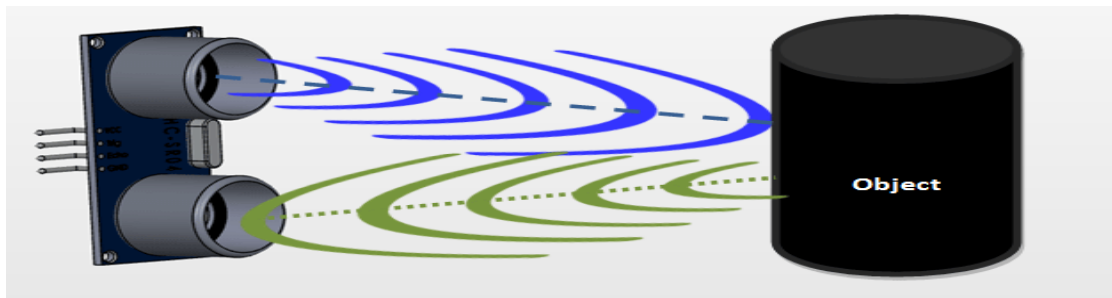
### **HC-SR04 Sensor Features**

- Operating voltage: +5V
- Theoretical Measuring Distance: 2cm to 450cm
- Practical Measuring Distance: 2cm to 80cm
- Accuracy: 3mm
- Measuring angle covered: <15°
- Operating Current: <15mA
- Operating Frequency: 40Hz

The **HC-SR04 Ultrasonic (US) sensor** is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required. The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the simple high school formula that

$$\text{Distance} = \text{Speed} \times \text{Time}$$

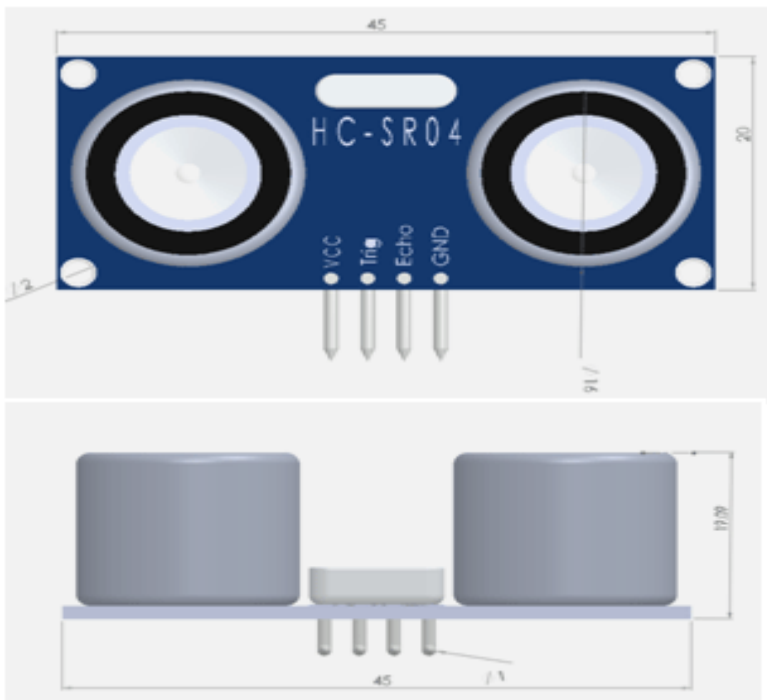
The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module.





## Applications

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

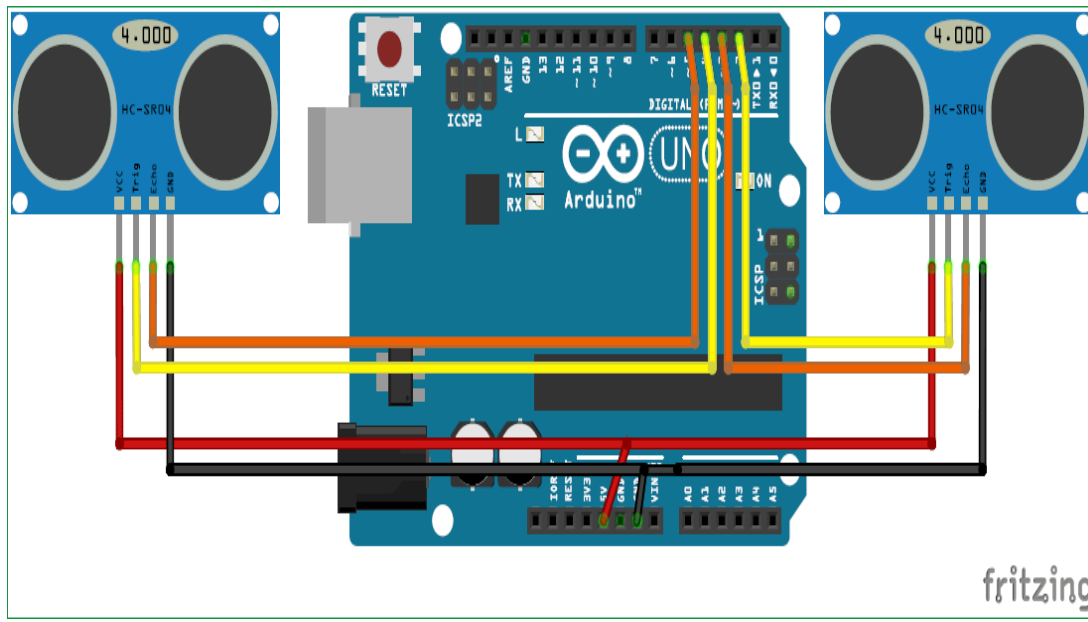


## COMPONENTS REQUIRED .

- 2 Ultrasonic Sensors
- Arduino
- Breadboard
- Jumper wires

We will place two Ultrasonic (US) sensors on top of our monitor and will read the distance between the monitor and our hand using Arduino, based on this value of distance we will perform certain actions. To perform actions on our computer we use Python pyautogui library. The commands from Arduino are sent to the computer through serial port (USB). This data will be then read by python which is running on the computer and based on the read data an action will be performed.

## **CIRCUIT DIAGRAM:**



## **ARDUINO PROGRAMMING:..**

The Arduino should be programmed to read the distance of hand from the US sensor.

Reading the value of distance we can arrive at certain actions to be controlled with gestures, for example in this program I have programmed **5 actions** as a demo.

**Action 1:** When both the hands are placed up before the sensor at a particular far distance then the video in VLC player should Play/Pause.

**Action 2:** When right hand is placed up before the sensor at a particular far distance then the video should Fast Forward one step.

**Action 3:** When left hand is placed up before the sensor at a particular far distance then the video should Rewind one step.

**Action 4:** When right hand is placed up before the sensor at a particular near distance and then if moved towards the sensor the video should fast forward and if moved away the video should Rewind.

**Action 5:** When left hand is placed up before the sensor at a particular near distance and then if moved towards the sensor the volume of video should increase and if moved away the volume should Decrease.

### **CODE:**

```
const int trigger1 = 2;
```

```
const int echo1 = 3;
```

```
const int trigger2 = 4;
```

```
const int echo2 = 5;
```

```
long time_taken;
```

```
int dist,distL,distR;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  pinMode(trigger1, OUTPUT);
```

```
  pinMode(echo1, INPUT);
```

```
  pinMode(trigger2, OUTPUT);
```

```
pinMode(echo2, INPUT);  
  
}
```

```
void calculate_distance(int trigger, int echo)  
{  
    digitalWrite(trigger, LOW);  
    delayMicroseconds(2);  
    digitalWrite(trigger, HIGH);  
    delayMicroseconds(10);  
    digitalWrite(trigger, LOW);
```

```
    time_taken = pulseIn(echo, HIGH);  
    dist= time_taken*0.034/2;  
    if (dist>50)  
        dist = 50;  
}
```

```
void loop() {  
    calculate_distance(trigger1,echo1);  
    distL =dist;  
  
    calculate_distance(trigger2,echo2);  
    distR =dist; //get distance of right sensor
```

```
//Uncomment for debudding
```

```
Serial.print("R=");
```

```
Serial.println(distR);
```

```
//Pause Modes -Hold
```

```
if ((distL >40 && distR>40) && (distL <50 && distR<50)) //Detect both hands
```

```
{ Serial.println("Play/Pause"); delay (500);}
```

```
calculate_distance(trigger1,echo1);
```

```
distL =dist;
```

```
calculate_distance(trigger2,echo2);
```

```
distR =dist;
```

```
if (distL>=13 && distL<=17)
```

```
{
```

```
    delay(100);
```

```
calculate_distance(trigger1,echo1);

distL =dist;

if (distL>=13 && distL<=17)

{

    Serial.println("Left Locked");

    while(distL<=40)

    {

        calculate_distance(trigger1,echo1);

        distL =dist;

        if (distL<10)

            { Serial.println ("Vup"); delay (300);}

        if (distL>20)

            { Serial.println ("Vdown"); delay (300);}

    }

}

}
```

```
if (distR>=13 && distR<=17)

{

    delay(100);

    calculate_distance(trigger2,echo2);

    distR =dist;
```

```

if (distR>=13 && distR<=17)
{
  Serial.println("Right Locked");
  while(distR<=40)
  {
    calculate_distance(trigger2,echo2);

    distR =dist;

    if (distR<10)

    {Serial.println ("Rewind"); delay (300);}

    if (distR>20)

    {Serial.println ("Forward"); delay (300);}

  }
}

}

}

delay(200);

}

```

## **PYTHON PROGRAMMING:**

We just have to establish a serial communication with Arduino through the correct baud rate and then perform some basic keyboard actions. The first step with python would be to install the **pyautogui** module.

## **PYAUTOGUI module:**

PyAutoGUI is a cross-platform GUI automation Python module for human beings. Used to programmatically control the mouse & keyboard.

```
import serial
```

```
import time
```

```
import pyautogui
```

```
ArduinoSerial = serial.Serial('com7',9600)
```

```
time.sleep(2)
```

```
while 1:
```

```
    incoming = str (ArduinoSerial.readline())
```

```
    if 'Play/Pause' in incoming:
```

```
        pyautogui.typewrite(['space'], 0.2)
```

```
    if 'Rewind' in incoming:
```

```
        pyautogui.hotkey('ctrl', 'left')
```

```
    if 'Forward' in incoming:
```

```
        pyautogui.hotkey('ctrl', 'right')
```

```
    if 'Vup' in incoming:
```

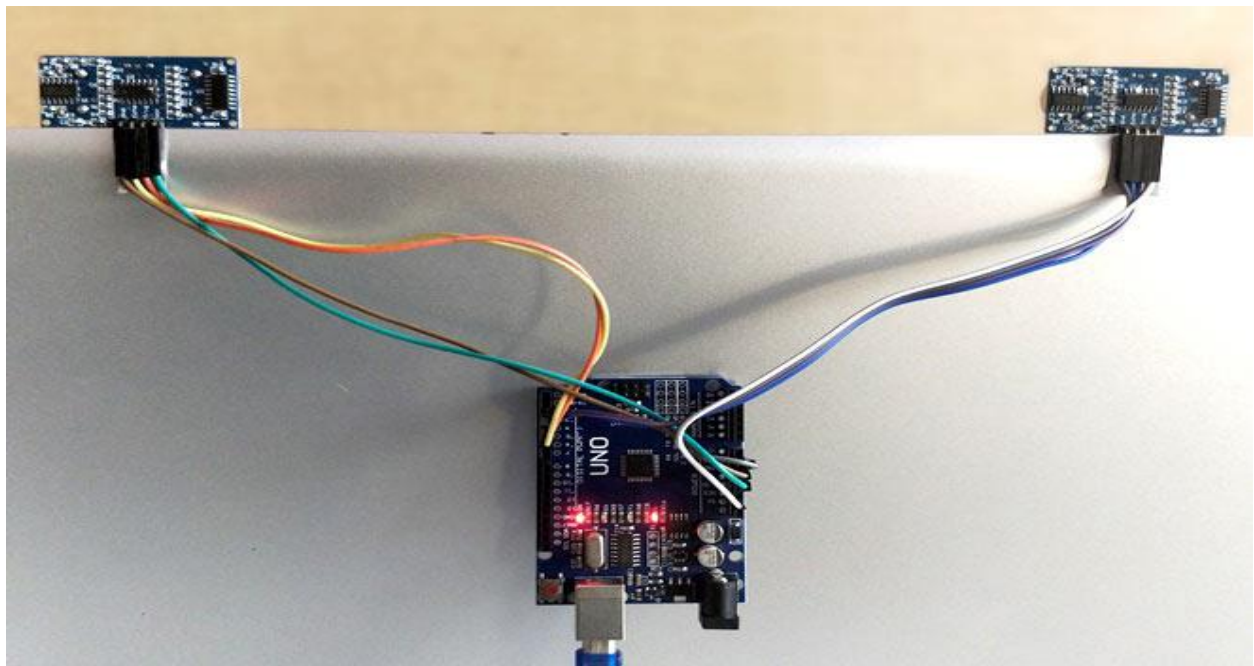
```
        pyautogui.hotkey('ctrl', 'down')
```



```
if 'Vdown' in incoming:
```

```
    pyautogui.hotkey('ctrl', 'up')
```

```
incoming = "";
```



## **About the game....**

We create a game called "2D Breakout" using JavaScript and HTML 5. In the 2D Breakout game, you need to break all the bricks using a bouncing ball, a ball that you need to prevent from leaving the game area using the paddle.

In the first part, we will be able, to draw the paddle and the ball, we will be able to control the paddle using the left and the right arrows on the keyboard, we will make the ball moves, and we will also implement the collision detection logic, so when the ball hits a wall it must change the direction.

When the ball hit the paddle, the ball should go in a definite direction based on where the ball hit the paddle so the paddle won't act like a wall. which means you can determine where the ball goes when you hit it with the paddle.

In the second part, we will implement the bricks in the game, add a collision detection function, when the ball hit a brick, the brick must disappear and then increment the player's score.

The player has 3 lives, when he loses a life, we reset the ball position and give him a chance to continue playing, when he loses all the 3 lives, it's a game over, we show him a game over message, and a "play again" button to play over.

To win the game, the player has to break all the bricks on each level, and he has to complete 7 levels, when he passes from a level to another, the number of bricks increases and also the speed of the ball.

We will also implement some sounds in the game, and we will create a button for when you want to turn ON/OFF the sounds.

## **Used Technologies in Project**

- HTML
- CSS
- Javascript
- Bootstrap(responsive web design framework)

## **What is HTML?**

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

**Hyper Text:** HyperText simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. HyperText is a way to link two or more web pages (HTML documents) with each other.

**Markup language:** A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

**Web Page:** A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages.**

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

### **Example of HTML.**

```
<!DOCTYPE>  
<html>  
<head>  
<title>Web page title</title>  
</head>  
<body>  
<h1>Write Your First Heading</h1>  
<p>Write Your First Paragraph.</p>  
</body>  
</html>
```

## **HTML Tags**

HTML tags are like keywords which defines that how web browser will format and display the content. With the help of tags, a web browser can distinguish between an HTML content and a simple content. HTML tags contain three main parts: opening tag, content and closing tag. But some HTML tags are unclosed tags.

- All HTML tags must enclosed within < > these brackets.
- Every tag in HTML perform different tasks.
- If you have used an open tag <tag>, then you must use a close tag </tag> (except some tags)

### **Some html tags**

<html>- It represents root of an HTML document.

<body>- It is used to define the body section of an HTML document.

**<head>** - It defines the head section of an HTML document.

**<title>** - It defines the title or name of an HTML document.

**<a>** - It is termed as anchor tag and it creates a hyperlink or link

**<marquee>** -It is used to insert the scrolling text or an image either horizontally or vertically

**<form>**- It is used to define an HTML form.

**<p>**- It represents a paragraph in an HTML document.

**<tr>** -It defines the row cells in an HTML table

**<th>**-It defines the row cells in an HTML table

**<u>**- It is used to render enclosed text with an underline.

## What is CSS ?

CSS provides basic and advanced concepts of CSS technology. Our CSS tutorial is developed for beginners and professionals. The major points of CSS are given below:

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.

- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.

## CSS Example

```
<!DOCTYPE>
<style>
h1{
color:white;
background-color:red;
padding:5px;
}
p{
color:blue;
}
</style>
</head>
<body>
<h1>Write Your First CSS Example</h1>
<p>This is Paragraph.</p>
</body>
</html>
```

## **What does CSS do?**

- You can add new looks to your old HTML documents.
- You can completely change the look of your website with only a few changes in CSS code.

## **Why use CSS**

These are the three major benefits of CSS:

### 1) Solves a big problem

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

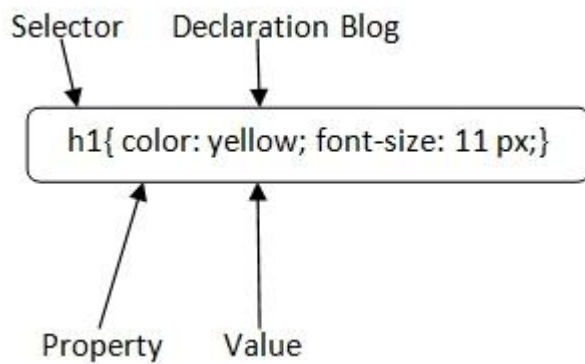
### 2) Saves a lot of time

CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

### 3) Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of the website

## **CSS Syntax**



**Selector:** Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.

**Declaration Block:** The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1. `color: yellow;`
2. `font-size: 11 px;`

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

**Property:** A Property is a type of attribute of HTML element. It could be color, border etc.

**Value:** Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

Selector{Property1: value1; Property2: value2; .....;}



# **Bootstrap**



Bootstrap is the popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.

Our Bootstrap tutorial includes all topics of Bootstrap such as jumbotron, table, button, grid, form, image, alert, wells, container, carousel, panels, glyphicon, badges, labels, progress bar, pagination, pager, list group, dropdown, collapse, tabs, pills, navbar, inputs, modals, tooltip, popover and scrollspy.

## **Why use Bootstrap**

Following are the main advantage of Bootstrap:

- It is very easy to use. Anybody having basic knowledge of HTML and CSS can use Bootstrap.
- It facilitates users to develop a responsive website.
- It is compatible on most of browsers like Chrome, Firefox, Internet Explorer, Safari and Opera etc.

## **What is a responsive website?**

A website is called responsive website which can automatically adjust itself to look good on all devices, from smart phones to desktops etc.

### **Bootstrap Example**

```
<div class="jumbotron text-center">  
  <h1>My First Bootstrap Page</h1>  
  <p>Resize this responsive page to see the effect!</p>  
</div>
```

```
<div class="container">  
  <div class="row">  
    <div class="col-sm-4">  
      <h3>Column 1</h3>  
      <p>Lorem ipsum dolor..</p>  
    </div>  
    <div class="col-sm-4">  
      <h3>Column 2</h3>  
      <p>Lorem ipsum dolor..</p>  
    </div>
```

```
<div class="col-sm-4">
  <h3>Column 3</h3>
  <p>Lorem ipsum dolor..</p>
</div>
</div>
</div>
```

## JavaScript

**JavaScript** is designed for beginners and professionals both. JavaScript is used to create client-side dynamic pages.

JavaScript is *an object-based scripting language* which is lightweight and cross-platform.

JavaScript is not a compiled language, but it is a translated language. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser.

### Application of JavaScript

JavaScript is used to create interactive websites. It is mainly used for:

Client-side validation,

Dynamic drop-down menus,

Displaying date and time,

Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box) Displaying clocks etc.

Front page code-

```
<!DOCTYPE html>
<html>
```

```
<head>

<title>PINGPONG</title>

<link rel="stylesheet" type="text/css" href="css/bootstrap.css">

<link rel="stylesheet" type="text/css" href="css/bootstrap-theme.css">

<link rel="stylesheet" type="text/css" href="css/font-awesome.css">

<link rel="stylesheet" type="text/css" href="css/style.css">

<script src="js/jquery-3.1.1.js"></script>

<script src="js/bootstrap.js"></script>

<style>

body {

background-image: url("background_img.jpg");


}

.button {

display: inline-block;

border-radius: 4px;

background-color: #f4511e;

border: none;

color: #FFFFFF;

text-align: center;

font-size: 28px;

padding: 20px;

width: 200px;

transition: all 0.5s;

cursor: pointer;
```

```
margin: 5px;  
}
```

```
.button span {  
  cursor: pointer;  
  display: inline-block;  
  position: relative;  
  transition: 0.5s;  
}
```

```
.button span:after {  
  content: '\00bb';  
  position: absolute;  
  opacity: 0;  
  top: 0;  
  right: -20px;  
  transition: 0.5s;  
}
```

```
.button:hover span {  
  padding-right: 25px;  
}
```

```
.button:hover span:after {  
  opacity: 1;
```

```
right: 0;
```

```
}
```

```
</head>
```

```
</style>
```

```
<body>
```

```
<center><h1 style="font-size:700%; color:yellow; margin-top:
200px;">PINGPONG</h1></center>
```

```
<center><button class="button" style="vertical-align:middle"><span><a href="loading.php"
Style="text-decoration: none; color:white;">Let play!<a></span></button></center>
```

```
<audio controls autoplay>
```

```
<source src="mario_theme_song.mp3" type="audio/mpeg">
```

```
</audio>
```

```
</div>
```

```
</body>
```

```
</html>
```

Fig-



Second page-

```
<html>
```

```
<head>
```

```
<title>PINGPONG</title>
```

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

```
<meta http-equiv="Refresh" content="10;url=start.php">
```

```
<link rel="stylesheet"
```

```
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
<script
```

```
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
```

```
<style>
```

```
body {
```

```
background-image: url("background_img.jpg");
```

```
}  
  
.button {  
  
display: inline-block;  
  
border-radius: 4px;  
  
background-color: #f4511e;  
  
border: none;  
  
color: #FFFFFF;  
  
text-align: center;  
  
font-size: 28px;  
  
padding: 20px;  
  
width: 200px;
```

```
margin: 5px;
```

```
}
```

```
</head>
```

```
</style>
```

```
<body>
```

```
<center><h1 style="font-size:700%; color:yellow; margin-top:  
200px;">PINGPONG</h1></center>
```

```
<center><button class="button" ><span class="spinner-border spinner-border-s" style="margin-  
top:5px;"></span><a href="Start.php" Style="text-decoration: none;  
color:white;">loading</a></button></center>
```



```
</div>
</body>
</html>
```



```
<html>
<head>
<title>PINGPONG</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Refresh" content="10;url=start.php">
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
<style>
body {
background-image: url("img6.png");
```

```
}  
  
.button {  
  display: inline-block;  
  border-radius: 4px;  
  background-color: #f4511e;  
  border: none;  
  color: #FFFFFF;  
  text-align: center;  
  font-size: 28px;  
  padding: 20px;  
  width: 200px;  
  
  margin: 5px;  
}
```

```
</head>
```

```
</style>
```

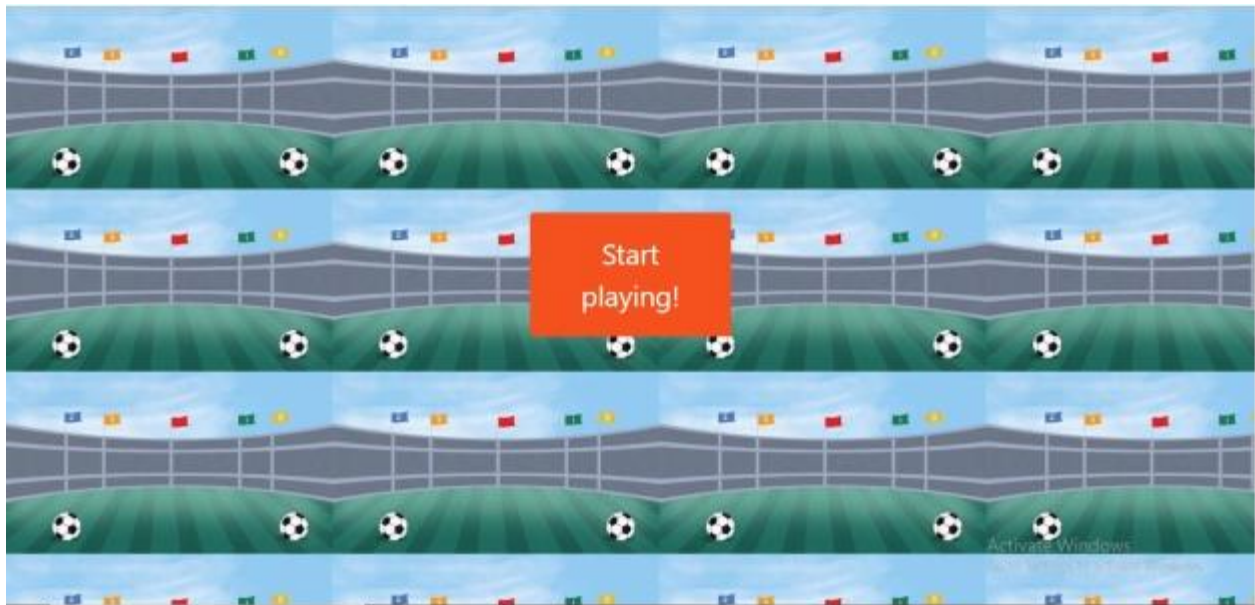
```
<body>
```

```
<center style="margin:200px;"><button class="button" style="vertical-align:middle"><span><a  
href="index.html" Style="text-decoration: none; color:white;">Start  
playing!<a></span></button></center>
```

```
</div>
```

```
</body>
```

```
</html>
```



```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>ping_pong</title>
```

```
<link rel="stylesheet" href="http://fonts.googleapis.com/css?family=Germania+One">
```

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
```

```
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
```

```
<style>
```

```
*{
```

```
font-family: "Germania One";
```

```
}
```

```
#sound{
```

```
width: 30px;
```

```
height : 30px;
```

```
cursor: pointer;
```

```
position: absolute;
```

```
top : 465px;
```

```
left : 15px;
```

```
}
```

```
#gameover{
```

```
position: absolute;
```

```
width : 1236px;
```

```
height: 579px;
```

```
padding-left:250px
```

```
background: rgba(0,0,0,0.5);
```

```
display: none;
```

```
}  
  
#youwon{  
width: 100px;  
position: absolute;  
padding-left:250px;  
top : 200px;  
left : 150px;  
display: none;  
}  
  
#youlose{  
width:800px;  
position: absolute;  
padding-left:360px;  
top: 100px;  
left : 0px;  
display: none;  
}  
  
#restart{  
position: absolute;  
top: 310px;  
left : 155px;  
padding-left:350px;  
cursor: pointer;  
font-size: 1.25em;
```

```

color : #FFF;

}

</style>

</head>

<body>

<div class="sound">



</div>


<div id="gameover">





<div id="restart"><button type="button" class="btn btn-warning">Play Again!</button>
<button type="button" class="btn btn-warning"><a href="front.php" Style="text-decoration:
none; color:white;">Exit Game!</a></button></div>

</div>


<canvas id="breakout" width="1236" height="574"></canvas>


<script src="component.js"></script>

<script src="games.js"></script>

</body>

</html>

// SELECT CANVAS ELEMENT

const cvs = document.getElementById("breakout");

const ctx = cvs.getContext("2d");

```

```
// ADD BORDER TO CANVAS
```

```
cvs.style.border = "3px solid #0ff";
```

```
// MAKE LINE THIK WHEN DRAWING TO CANVAS
```

```
ctx.lineWidth = 3;
```

```
// GAME VARIABLES AND CONSTANTS
```

```
const PADDLE_WIDTH = 200;
```

```
const PADDLE_MARGIN_BOTTOM = 50;
```

```
const PADDLE_HEIGHT = 15;
```

```
const BALL_RADIUS = 10;
```

```
let LIFE = 3; // PLAYER HAS 3 LIVES
```

```
let SCORE = 0;
```

```
const SCORE_UNIT = 10;
```

```
let LEVEL = 1;
```

```
const MAX_LEVEL = 3;
```

```
let GAME_OVER = false;
```

```
let leftArrow = false;
```

```
let rightArrow = false;
```

```
// CREATE THE PADDLE
```

```
const paddle = {
```

```
  x : cvs.width/2 - PADDLE_WIDTH/2,
```

```
  y : cvs.height - PADDLE_MARGIN_BOTTOM - PADDLE_HEIGHT,
```

```
width : PADDLE_WIDTH,  
height : PADDLE_HEIGHT,  
dx :10  
}
```

```
// DRAW PADDLE
```

```
function drawPaddle(){  
  ctx.fillStyle = "#2e3548";  
  ctx.fillRect(paddle.x, paddle.y, paddle.width, paddle.height);  
  
  ctx.strokeStyle = "#ffcd05";  
  ctx.strokeRect(paddle.x, paddle.y, paddle.width, paddle.height);  
}
```

```
// CONTROL THE PADDLE
```

```
document.addEventListener("keydown", function(event){  
  if(event.keyCode == 37){  
    leftArrow = true;  
  }else if(event.keyCode == 39){  
    rightArrow = true;  
  }  
});  
  
document.addEventListener("keyup", function(event){  
  if(event.keyCode == 37){  
    leftArrow = false;
```



```
}else if(event.keyCode == 39){  
  rightArrow = false;  
}  
});
```

```
// MOVE PADDLE
```

```
function movePaddle(){  
  if(rightArrow && paddle.x + paddle.width < cvs.width){  
    paddle.x += paddle.dx;  
  }else if(leftArrow && paddle.x > 0){  
    paddle.x -= paddle.dx;  
  }  
}
```

```
// CREATE THE BALL
```

```
const ball = {  
  x : cvs.width/2,  
  y : paddle.y - BALL_RADIUS,  
  radius : BALL_RADIUS,  
  speed : 4,  
  dx : 3 * (Math.random() * 2 - 1),  
  dy : -3  
}
```

```
// DRAW THE BALL
```

```
function drawBall(){
    ctx.beginPath();

    ctx.arc(ball.x, ball.y, ball.radius, 0, Math.PI*2);
    ctx.fillStyle = "#ff0000";
    ctx.fill();

    ctx.strokeStyle = "#ffff00";
    ctx.stroke();

    ctx.closePath();
}

// MOVE THE BALL
function moveBall(){
    ball.x += ball.dx;
    ball.y += ball.dy;
}

// BALL AND WALL COLLISION DETECTION
function ballWallCollision(){
    if(ball.x + ball.radius > cvs.width || ball.x - ball.radius < 0){
        ball.dx = - ball.dx;
        WALL_HIT.play();
    }
}
```

```
if(ball.y - ball.radius < 0){  
    ball.dy = -ball.dy;  
    WALL_HIT.play();  
}
```

```
if(ball.y + ball.radius > cvs.height){  
    LIFE--; // LOSE LIFE  
    LIFE_LOST.play();  
    resetBall();  
}  
}
```

```
// RESET THE BALL
```

```
function resetBall(){  
    ball.x = cvs.width/2;  
    ball.y = paddle.y - BALL_RADIUS;  
    ball.dx = 3 * (Math.random() * 2 - 1);  
    ball.dy = -3;  
}
```

```
// BALL AND PADDLE COLLISION
```

```
function ballPaddleCollision(){  
    if(ball.x < paddle.x + paddle.width && ball.x > paddle.x && paddle.y < paddle.y +  
        paddle.height && ball.y > paddle.y){
```

```
// PLAY SOUND
```

```
PADDLE_HIT.play();
```

```
// CHECK WHERE THE BALL HIT THE PADDLE
```

```
let collidePoint = ball.x - (paddle.x + paddle.width/2);
```

```
// NORMALIZE THE VALUES
```

```
collidePoint = collidePoint / (paddle.width/2);
```

```
// CALCULATE THE ANGLE OF THE BALL
```

```
let angle = collidePoint * Math.PI/3;
```

```
ball.dx = ball.speed * Math.sin(angle);
```

```
ball.dy = - ball.speed * Math.cos(angle);
```

```
}
```

```
}
```

```
// CREATE THE BRICKS
```

```
const brick = {
```

```
  row : 1,
```

```
  column : 9,
```

```
  width : 60,
```

```
  height : 20,
```

```
offSetLeft : 60,  
offSetTop : 20,  
marginTop : 40,  
fillColor : "    #b2d8d8",  
strokeColor : "black"  
}
```

```
let bricks = [];
```

```
function createBricks(){  
  for(let r = 0; r < brick.row; r++){  
    bricks[r] = [];  
    for(let c = 0; c < brick.column; c++){  
      bricks[r][c] = {  
        x : c * ( brick.offSetLeft + brick.width ) + brick.offSetLeft,  
        y : r * ( brick.offSetTop + brick.height ) + brick.offSetTop + brick.marginTop,  
        status : true  
      }  
    }  
  }  
}
```

```
createBricks();
```

```
// draw the bricks
```

```
function drawBricks(){
  for(let r = 0; r < brick.row; r++){
    for(let c = 0; c < brick.column; c++){
      let b = bricks[r][c];

      // if the brick isn't broken
      if(b.status){
        ctx.fillStyle = brick.fillColor;
        ctx.fillRect(b.x, b.y, brick.width, brick.height);

        ctx.strokeStyle = brick.strokeColor;
        ctx.strokeRect(b.x, b.y, brick.width, brick.height);
      }
    }
  }
}

// ball brick collision
function ballBrickCollision(){
  for(let r = 0; r < brick.row; r++){
    for(let c = 0; c < brick.column; c++){
      let b = bricks[r][c];

      // if the brick isn't broken
      if(b.status){
        if(ball.x + ball.radius > b.x && ball.x - ball.radius < b.x + brick.width && ball.y + ball.radius >
        b.y && ball.y - ball.radius < b.y + brick.height){
```

```
BRICK_HIT.play();

ball.dy = - ball.dy;

b.status = false; // the brick is broken

SCORE += SCORE_UNIT;

}

}

}

}

}

// show game stats

function showGameStats(text, textX, textY, img, imgX, imgY){

// draw text

ctx.fillStyle = "#FFF";

ctx.font = "25px Germania One";

ctx.fillText(text, textX, textY);


// draw image

ctx.drawImage(img, imgX, imgY, width = 25, height = 25);

}


// DRAW FUNCTION

function draw(){

drawPaddle();
```

```
drawBall();
```

```
drawBricks();
```

```
// SHOW SCORE
```

```
showGameStats(SCORE, 35, 25, SCORE_IMG, 5, 5);
```

```
// SHOW LIVES
```

```
showGameStats(LIFE, cvs.width - 25, 25, LIFE_IMG, cvs.width-55, 5);
```

```
// SHOW LEVEL
```

```
showGameStats(LEVEL, cvs.width/2, 25, LEVEL_IMG, cvs.width/2 - 30, 5);
```

```
}
```

```
// game over
```

```
function gameOver(){
```

```
if(LIFE <= 0){
```

```
showYouLose();
```

```
GAME_OVER = true;
```

```
}
```

```
}
```

```
// level up
```

```
function levelUp(){
```

```
let isLevelDone = true;
```

```
// check if all the bricks are broken
```



```
for(let r = 0; r < brick.row; r++){  
  for(let c = 0; c < brick.column; c++){  
    isLevelDone = isLevelDone && ! bricks[r][c].status;  
  }  
}
```

```
if(isLevelDone){  
  WIN.play();
```

```
  if(LEVEL >= MAX_LEVEL){  
    showYouWin();  
    GAME_OVER = true;  
    return;  
  }  
  brick.row++;  
  createBricks();  
  ball.speed += 0.5;  
  resetBall();  
  LEVEL++;  
}
```

```
// UPDATE GAME FUNCTION
```

```
function update(){  
  movePaddle();
```

```
moveBall();
```

```
ballWallCollision();
```

```
ballPaddleCollision();
```

```
ballBrickCollision();
```

```
gameOver();
```

```
levelUp();
```

```
}
```

```
// GAME LOOP
```

```
function loop(){
```

```
// CLEAR THE CANVAS
```

```
ctx.drawImage(BG_IMG,0,0,1250,600);
```

```
draw();
```

```
update();
```

```
if(! GAME_OVER){
```

```
requestAnimationFrame(loop);
```

```
}
```

```
}
```

```
loop();
```

```
// SELECT SOUND ELEMENT
```

```
const soundElement = document.getElementById("sound");
```

```
soundElement.addEventListener("click", audioManager);
```

```
function audioManager(){
```

```
// CHANGE IMAGE SOUND_ON/OFF
```

```
let imgSrc = soundElement.getAttribute("src");
```

```
let SOUND_IMG = imgSrc == "img/SOUND_ON.png" ? "img/SOUND_OFF.png" :  
"img/SOUND_ON.png";
```

```
soundElement.setAttribute("src", SOUND_IMG);
```

```
// MUTE AND UNMUTE SOUNDS
```

```
WALL_HIT.muted = WALL_HIT.muted ? false : true;
```

```
PADDLE_HIT.muted = PADDLE_HIT.muted ? false : true;
```

```
BRICK_HIT.muted = BRICK_HIT.muted ? false : true;
```

```
WIN.muted = WIN.muted ? false : true;
```

```
LIFE_LOST.muted = LIFE_LOST.muted ? false : true;
```

```
}
```

```
// SHOW GAME OVER MESSAGE

/* SELECT ELEMENTS */

const gameover = document.getElementById("gameover");
const youwin = document.getElementById("youwin");
const youlose = document.getElementById("youlose");
const restart = document.getElementById("restart");
```

```
// CLICK ON PLAY AGAIN BUTTON

restart.addEventListener("click", function(){
location.reload(); // reload the page
})
```

```
// SHOW YOU WIN

function showYouWin(){
gameover.style.display = "block";
youwon.style.display = "block";
}
```

```
// SHOW YOU LOSE

function showYouLose(){
gameover.style.display = "block";
youlose.style.display = "block";
}
```

Component used in site code-

```
//////// LOAD IMAGES //////////
```

```
// LOAD BG IMAGE
```

```
const BG_IMG = new Image();
```

```
BG_IMG.src = "img/img9.jpg";
```

```
const LEVEL_IMG = new Image();
```

```
LEVEL_IMG.src = "img/level.png";
```

```
const LIFE_IMG = new Image();
```

```
LIFE_IMG.src = "img/life.png";
```

```
const SCORE_IMG = new Image();
```

```
SCORE_IMG.src = "img/score.png";
```

```
//////// END LOAD IMAGES //////////
```

```
// ***** //
```

```
//////// LOAD SOUNDS //////////
```

```
const WALL_HIT = new Audio();
```

```
WALL_HIT.src = "sounds/wall.mp3";
```

```
const LIFE_LOST = new Audio();
```

```
LIFE_LOST.src = "sounds/life_lost.mp3";
```

```
const PADDLE_HIT = new Audio();
```

```
PADDLE_HIT.src = "sounds/paddle_hit.mp3";
```

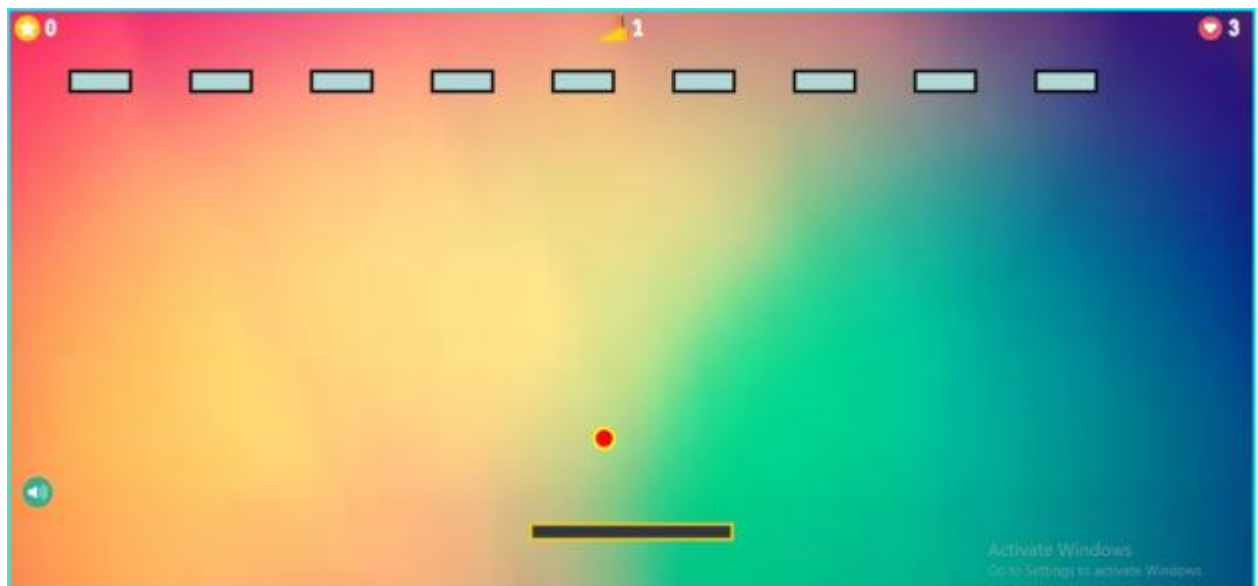
```
const WIN = new Audio();
```

```
WIN.src = "sounds/win.mp3";
```

```
const BRICK_HIT = new Audio();
```

```
BRICK_HIT.src = "sounds/brick_hit.mp3";
```

```
//////// END LOAD SOUNDS //////////
```







## **REFERENCES:**

### **ONLINE:**

- YouTube.COM
- Wikipedia
- www.w3schools.com

### **Book:**

- The complete reference HTML and CSS
- The Internet Of Things by David Etter

### **Faculty:**

- Mr.Ajitesh Sir
- Mr. Amir Khan Sir
- Mrs. Mona Kumari Mam





