

6.1 LOGIN PAGE

```
<!DOCTYPE html>
<html>
<head>
<title>Login/Signup</title>
<!-- custom-theme -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="Simple Login and Signup Form web template Responsive,
Login form web template,Flat Pricing tables,Flat Drop downs  Sign up Web Templates,
Flat Web Templates, Login sign up Responsive web template, SmartPhone Compatible web
template, free webdesigns for Nokia, Samsung, LG, SonyEricsson, Motorola web design" />
<script type="application/x-javascript"> addEventListener("load", function()
{ setTimeout(hideURLbar, 0); }, false);
        function hideURLbar(){ window.scrollTo(0,1); } </script>
<!-- //custom-theme -->
<link href="css/style1.css" rel="stylesheet" type="text/css" media="all" />
<!-- js -->
<script src="js/jquery-1.9.1.min.js"></script>
<!--// js -->
<link rel="stylesheet" type="text/css" href="css/easy-responsive-tabs.css " />
<link href="//fonts.googleapis.com/css?family=Questrial" rel="stylesheet">
</head>
<body class="bg agileinfo">
    <h1 class="agile_head text-center"> Simple Login and Signup Form</h1>
    <div class="w3layouts_main wrap">
        <!--Horizontal Tab-->
        <div id="parentHorizontalTab_agile">
            <ul class="resp-tabs-list hor_1">
                <li>LogIn</li>
                <li>SignUp</li>
            </ul>
            <div class="resp-tabs-container hor_1">
                <div class="w3_agile_login">
                    <form action="index.html" method="post" class="agile_form">
                        <p>Email</p>
                        <input type="email" name="username"
required="required" />
                        <p>Password</p>
                        <input type="password" name="username">
```

```

required="required" class="password" />
        <div class="check">
            <label class="checkbox w3l"><input
type="checkbox" name="checkbox" required="required"><i> </i>I accept the terms and
conditions</label>

            </div>
            <input type="submit" value="LogIn" class="agileinfo"
/>

        </form>
        <div class="login_w3ls">
            <a href="#">Forgot Password</a>
        </div>

    </div>
    <div class="agile_its_registration">
        <form action="index.html" method="get" class="agile_form">
            <p>Username</p>
            <input type="text" name="username"
required="required" />

            <p>Email</p>
            <input type="email" name="email" required="required"
/>

            <p>Password</p>
            <input type="password" name="Password"
id="password1" required="required">

            <p>Confirm Password</p>
            <input type="password" name="Confirm Password"
id="password2" required="required">

            <div class="check w3_agileits">
                <label class="checkbox w3"><input
type="checkbox" name="checkbox" required="required" ><i> </i>I accept the terms and
conditions</label>

                </div>
                <input type="submit" value="Signup"/>
                <!-- <input type="reset" value="Reset" /> -->
            </form>
        </div>

    </div>
</div>
    <!-- //Horizontal Tab -->
</div>
    <div class="agileits_w3layouts_copyright text-center">
        </div>

```

```

<!--tabs-->
<script src="js/easyResponsiveTabs.js"></script>
<script type="text/javascript">
$(document).ready(function() {
    //Horizontal Tab
    $('#parentHorizontalTab_agile').easyResponsiveTabs({
        type: 'default', //Types: default, vertical, accordion
        width: 'auto', //auto or any width like 600px
        fit: true, // 100% fit in a container
        tabidentify: 'hor_1', // The tab groups identifier
        activate: function(event) { // Callback function if tab is switched
            var $tab = $(this);
            var $info = $('#nested-tabInfo');
            var $name = $('span', $info);
            $name.text($tab.text());
            $info.show();
        }
    });
});
</script>
<script type="text/javascript">
    window.onload = function () {
        document.getElementById("password1").onchange = validatePassword;
        document.getElementById("password2").onchange = validatePassword;
    }
    function validatePassword(){
        var pass2=document.getElementById("password2").value;
        var pass1=document.getElementById("password1").value;
        if(pass1!=pass2)

        document.getElementById("password2").setCustomValidity("Passwords Don't Match");
        else
            document.getElementById("password2").setCustomValidity('');
            //empty string means no validation error
        }
    }
</script>
<!--//tabs-->
</body>
</html>

```

6.2 INDEX/HOME PAGE

```
<!DOCTYPE HTML>
<html>
<head>
<title></title>
<link href="css/style.css" rel="stylesheet" >
</head>
<body>
  <div class="header-top">
    <div class="wrap">
      <div class="cssmenu">
        <ul>
          <li class="active"><a href="#">Account</a></li> |
          <li><a href="login.html">Log In</a></li> |
          <!-- <li><a href="#">Sign Up</a></li> -->
        </ul>
      </div>
      <div class="clear"></div>
    </div>
  </div>
  <div class="header-bottom">
    <div class="wrap">
      <div class="header-bottom-left">
        <div class="logo">
          <a href="index.html"></a>
        </div>
      </div>
    </div>
  </div>
  <!--/slider -->
<div class="main">
  <div class="wrap">
    <div class="section group">
      <div class="cont span_2_of_3">
        <h2 class="head">Featured Games</h2>
        <div class="top-box">
          <div class="col_1_of_3 span_1_of_3">
            <div class="inner_content clearfix">
              <div class="product_image">
                

```

```

        </div>
        <a href="movethebox.html">
<div class="price">
    <div class="cart-left">
        <p class="title">Move The Box</p>
    </div>
</div>
</div>
</a>
</div>
<div class="col_1_of_3 span_1_of_3">
    <div class="inner_content clearfix">
        <div class="product_image">
            
        </div>
        <a href="snake.html">
<div class="price">
            <div class="cart-left">
                <p class="title">Snake</p>
            </div>
        </div>
    </div>
</a>
</div>
<div class="col_1_of_3 span_1_of_3">
    <a href="quiz/quiz.html">
        <div class="inner_content clearfix">
            <div class="product_image">
                
            </div>
        </div>
    </a>
<div class="sale-box1"></div>
    <div class="price">
        <div class="cart-left">
            <p class="title">Quiz Time</p>
        </div>
    </div>
</div>
</a>
</div>
<div class="clear"></div>
</div>
<div class="top-box">
    <div class="col_1_of_3 span_1_of_3">

```

```
<a href="psychic.html">
    <div class="inner_content clearfix">
        <div class="product_image">
            
        </div>
        <div class="price">
            <div class="cart-left">
                <p class="title">Psychic Game</p>
            </div>
        </div>
    </div>
</a>
    </div>
    <div class="clear"></div>
</div>
<div class="top-box1">
    <div class="clear"></div>
</div>

</div>
<div class="clear"></div>
</div>
</div>
</div>
<div class="footer">
    <div class="footer-bottom">
        <div class="wrap">
            <div class="copy">
                <p>Developed by Atul and Anwesha</a></p>
            </div>
            <div class="f-list2">
                <ul>
  
  
  
  
  
  
  
  
  
  
  
  
  

                </ul>
            </div>
            <div class="clear"></div>
        </div>
    </div>
</div>
</body>
</html>
```

6.3.MOVE THE BOX GAME

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0"/>
<style>
canvas {
    border: 1px solid black;
    background-color: white;
}
</style>
</head>
<body onload="startGame()">
<script>
function startGame() {
    myGameArea.start();
}
var myGamePiece;
var myObstacles = [];
var myScore;
function startGame() {
    myGameArea.start();
    myGamePiece = new component(30, 30, "red", 2, 2);
    myObstacle = new component(10, 200, "green", 300, 120);
    myScore = new component("30px", "Consolas", "black", 280, 40, "text");
}
var myGameArea = {
    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 750;
        this.canvas.height = 400;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas,
document.body.childNodes[0]);

        this.frameNo = 0;
        this.interval = setInterval(updateGameArea, 20);
        //keyboard control
        window.addEventListener('keydown', function (e) {
            myGameArea.key = e.keyCode;
```

```

    })
    window.addEventListener('keyup', function (e) {
        myGameArea.key = false;
    })
    },
    //for moving purpose
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
    },
    stop : function() {
        clearInterval(this.interval);
    }
}
function everyinterval(n) {
    if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
    return false;
}
function component(width, height, color, x, y,type) {
    this.type = type;
    this.width = width;
    this.height = height;
    this.x = x;
    this.y = y;
    this.speedX = 0;//movement purpose
    this.speedY = 0;

    this.update = function(){//move continuously and update
        ctx = myGameArea.context;

        //introduce score
        if (this.type == "text") {
            ctx.font = this.width + " " + this.height;
            ctx.fillStyle = color;
            ctx.fillText(this.text, this.x, this.y);
        } else {
            ctx.fillStyle = color;
            ctx.fillRect(this.x, this.y, this.width, this.height);
        }

        /* ctx.fillStyle = color;
        ctx.fillRect(this.x, this.y, this.width, this.height);*/
    }
    this.newPos = function() {

```



```

    this.x += this.speedX;
    this.y += this.speedY;
}
this.crashWith = function(otherobj) {
    var myleft = this.x;
    var myright = this.x + (this.width);
    var mytop = this.y;
    var mybottom = this.y + (this.height);
    var otherleft = otherobj.x;
    var otherright = otherobj.x + (otherobj.width);
    var othertop = otherobj.y;
    var otherbottom = otherobj.y + (otherobj.height);
    var crash = true;
    if ((mybottom < othertop) ||
        (mytop > otherbottom) ||
        (myright < otherleft) ||
        (myleft > otherright)) {
        crash = false;
    }
    return crash;
}
}
function updateGameArea() {
    var x, y, height, gap, minHeight, maxHeight, minGap, maxGap;
    myGameArea.clear();
    myGamePiece.speedX = 0;
    myGamePiece.speedY = 0;

    //control box by keyboard
    if (myGameArea.key && myGameArea.key == 37) {myGamePiece.speedX = -1; }
    if (myGameArea.key && myGameArea.key == 39) {myGamePiece.speedX = 1; }
    if (myGameArea.key && myGameArea.key == 38) {myGamePiece.speedY = -1; }
    if (myGameArea.key && myGameArea.key == 40) {myGamePiece.speedY = 1; }

    for (i = 0; i < myObstacles.length; i += 1) {
        if (myGamePiece.crashWith(myObstacles[i])) {
            myGameArea.stop();
            return;
        }
    }
    myGameArea.clear();
    myGameArea.frameNo += 1;

```

```

    if (myGameArea.frameNo == 1 || everyinterval(150)) {
        x = myGameArea.canvas.width;
        minHeight = 20;
        maxHeight = 200;
        height = Math.floor(Math.random()*(maxHeight-minHeight+1)+minHeight);
        minGap = 50;
        maxGap = 200;
        gap = Math.floor(Math.random()*(maxGap-minGap+1)+minGap);
        myObstacles.push(new component(10, height, "green", x, 0));
        myObstacles.push(new component(10, x - height - gap, "green", x, height
+ gap));
    }
    for (i = 0; i < myObstacles.length; i += 1) {
        myObstacles[i].x += -1;
        myObstacles[i].newPos();
        myObstacles[i].update();
    }
    //score
    myScore.text = "SCORE: " + myGameArea.frameNo;
    document.getElementById("scor").innerHTML=myScore.text;
    myScore.update();
    myGamePiece.newPos();
    myGamePiece.update();
}
function moveup() { myGamePiece.speedY -= 1;}
function movedown() { myGamePiece.speedY += 1;}
function moveleft() { myGamePiece.speedX -= 1;}
function moveright() { myGamePiece.speedX += 1;}
function stopMove() {
    myGamePiece.speedX = 0;
    myGamePiece.speedY = 0;
}
</script>
<p>move your arrow keys to move the object box</p>
<h2>Your Score is: </h2> <h1 id="scor"></h1>
</body>
</html>

```

6.4 SNAKE GAME PAGE

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <style>
    html, body {
      height: 100%;
      margin: 0;
    }
    body {
      background: black;
      display: flex;
      align-items: center;
      justify-content: center;
    }
    canvas {
      border: 1px solid white;
    }
  </style>
</head>
<body>
<canvas width="400" height="400" id="game"></canvas>
<h2 style="color:red;">&nbsp;score:</h2><h1 id="scorr" style="color:green;">0</h1>
<script>
var canvas = document.getElementById('game');
var context = canvas.getContext('2d');
var grid = 16;
var count = 0;
var score=0;
var snake = {
  x: 160,
  y: 160,

  // snake velocity. moves one grid length every frame in either the x or y
direction
  dx: grid,
  dy: 0,
```

```

    // keep track of all grids the snake body occupies
    cells: [],

    // length of the snake. grows when eating an apple
    maxCells: 4
};
var apple = {
    x: 320,
    y: 320
};
// get random whole numbers in a specific range
function getRandomInt(min, max) {
    return Math.floor(Math.random() * (max - min)) + min;
}
// game loop
function loop() {
    requestAnimationFrame(loop);
    // slow game loop to 15 fps instead of 60 (60/15 = 4)
    if (++count < 4) {
        return;
    }
    count = 0;
    context.clearRect(0,0,canvas.width,canvas.height);
    // move snake by it's velocity
    snake.x += snake.dx;
    snake.y += snake.dy;
    // wrap snake position horizontally on edge of screen
    if (snake.x < 0) {
        snake.x = canvas.width - grid;
    }
    else if (snake.x >= canvas.width) {
        snake.x = 0;
    }

    // wrap snake position vertically on edge of screen
    if (snake.y < 0) {
        snake.y = canvas.height - grid;
    }
    else if (snake.y >= canvas.height) {
        snake.y = 0;
    }
    // keep track of where snake has been. front of the array is always the head
    snake.cells.unshift({x: snake.x, y: snake.y});

```

```

// remove cells as we move away from them
if (snake.cells.length > snake.maxCells) {
    snake.cells.pop();
}
// draw apple
context.fillStyle = 'red';
context.fillRect(apple.x, apple.y, grid-1, grid-1);
// draw snake one cell at a time
context.fillStyle = 'green';
snake.cells.forEach(function(cell, index) {

    // drawing 1 px smaller than the grid creates a grid effect in the snake body
    so you can see how long it is
    context.fillRect(cell.x, cell.y, grid-1, grid-1);
    // snake ate apple
    if (cell.x === apple.x && cell.y === apple.y) {
        snake.maxCells++;
        score++;
        document.getElementById("score").innerHTML=score;
        console.log(score);
        // canvas is 400x400 which is 25x25 grids
        apple.x = getRandomInt(0, 25) * grid;
        apple.y = getRandomInt(0, 25) * grid;
    }

    // check collision with all cells after this one (modified bubble sort)
    for (var i = index + 1; i < snake.cells.length; i++) {

        // snake occupies same space as a body part. reset game
        if (cell.x === snake.cells[i].x && cell.y === snake.cells[i].y) {
            snake.x = 160;
            snake.y = 160;
            snake.cells = [];
            snake.maxCells = 4;
            snake.dx = grid;
            snake.dy = 0;
            apple.x = getRandomInt(0, 25) * grid;
            apple.y = getRandomInt(0, 25) * grid;
        }
    }
});
}

```

```

// listen to keyboard events to move the snake
document.addEventListener('keydown', function(e) {
    // prevent snake from backtracking on itself by checking that it's
    // not already moving on the same axis (pressing left while moving
    // left won't do anything, and pressing right while moving left
    // shouldn't let you collide with your own body)

    // left arrow key
    if (e.which === 37 && snake.dx === 0) {
        snake.dx = -grid;
        snake.dy = 0;
    }
    // up arrow key
    else if (e.which === 38 && snake.dy === 0) {
        snake.dy = -grid;
        snake.dx = 0;
    }
    // right arrow key
    else if (e.which === 39 && snake.dx === 0) {
        snake.dx = grid;
        snake.dy = 0;
    }
    // down arrow key
    else if (e.which === 40 && snake.dy === 0) {
        snake.dy = grid;
        snake.dx = 0;
    }
});
// start the game
requestAnimationFrame(loop);
</script>
</body>
</html>

```

6.5 QUIZ GAME PAGE

6.5.1 quiz.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Quiz</title>
    <link rel="stylesheet" href="style.css">
    <script src="javascr.js"></script>
  </head>
  <body>
    <h1>"The best HTML Quiz"</h1>
    <form id="quiz" name="quiz">
      <p>HTML stands for Hypertext _____ Language.</p>
      <input id="textbox" type="text" name="question1">
      <p>Html element is the root element of an HTML _____ document.</p>
      <input type="radio" id="mc" name="question2" value="true">true<br>
      <input type="radio" id="mc" name="question2" value="false">false<br>
      <p>Which tag is used to define the html images?</p>
      <input type="radio" id="mc" name="question3" value="img">img<br>
      <input type="radio" id="mc" name="question3" value="src">src<br>
      <input type="radio" id="mc" name="question3" value="href">href<br>
      <input type="radio" id="mc" name="question3" value="alt">alt<br>

      <p> Which of the following is an advantage of putting presentation information
in a separate CSS file rather than in HTML itself?</p>
      <input type="radio" id="mc" name="question4" value="i"> The content becomes
easy to manage<br>
      <input type="radio" id="mc" name="question4" value="m">Becomes easy to make
site for different devices like mobile by making separate CSS files<br>
      <input type="radio" id="mc" name="question4" value="g">CSS Files are generally
cached and therefore decrease server load and network traffic.<br>
      <input type="radio" id="mc" name="question4" value="im">All of the above<br>

      <p>Where in HTML document is the conventionally correct place to put reference
to an external style sheet?</p>
      <input type="radio" id="mc" name="question5" value="a">In the HEAD section<br>
      <input type="radio" id="mc" name="question5" value="b">In the BODY section<br>
      <input type="radio" id="mc" name="question5" value="c">Both of the Above<br>
```

```

<input type="radio" id="mc" name="question5" value="d">None of the Above<br>
<input id="button" type="button" value="I'm finished!" onclick="check();">
</form>
<div id="after_submit"></div>
<p id="messages"></p>
<p id="number_correct"></p>
<img id="pictures">
</body>
</html>

```

6.5.2 style.css

```

body {
    font-family: 'Lato', sans-serif;
}
#quiz {
    margin-left: 0px;
    background: #d2def2;
    padding: 10px 20px 10px 20px;
    width: device-width;
    border-radius: 20px;
    float: left;
}
h1 {
    text-align: center;
}
#input {
    margin-bottom: 20px;
    display: block;
}
#textbox {
    height: 22px;
    width: 120px;
    font-size: 16px;
    border-radius: 5px;
    border: none;
    padding-left: 5px;
}
#button {
    background: green;
    border: none;
    border-radius: 5px;
    padding: 10px;
}

```



```

        color:white;
        font-size:16px;
        transition-duration:.5s;
        margin-top:15px;
    }
    #button.hover {
        background:white;
        border:1px solid green;
        color:black;
        cursor:pointer;
    }
    #after_submit {
        visibility:hidden;
        background:#0000ff;
        padding:10px 20px 10px 20px;
        width:device-width;
        border-radius:20px;
        float:left;
        margin-left:20px;
        font-size:30px;
    }
    #pictures {
        width:20%;
        height:40%;
    }

```

6.5.3 javascr.js

```

function check(){
var question1=document.quiz.question1.value;
var question2=document.quiz.question2.value;
var question3=document.quiz.question3.value;
var question4=document.quiz.question4.value;
var question5=document.quiz.question5.value;
var correct=0;
    if (question1=="Markup"){
        correct++;
    }
    if (question2=="true"){
        correct++;
    }

```

```

    }
    if (question3=="img"){
        correct++;
    }
    if (question4=="im"){
        correct++;
    }
    if (question5=="a"){
        correct++;
    }
    var messages=["Great job!","That's just okay!","You really need to do better"];
    var pictures=["https://media1.giphy.com/media/1lsBLVxNs7v6WA/giphy.gif","https://
    media1.giphy.com/media/Nm9hS20D4swV0/200w.gif","https://media.giphy.com/media/
    xUA7aRaGvA53VSlxUk/giphy.gif"];
    var range;
    if(correct<1){
        range=2;
    }
    if(correct>0&&correct<6){
        range=1;
    }
    if(correct>6){
        range=0;
    }
    document.getElementById("after_submit").style.visibility="visible";
    document.getElementById("messages").innerHTML=messages[range];
    document.getElementById("number_correct").innerHTML="&rarr;you got "+correct+"
    correct.";
    document.getElementById("pictures").src=pictures[range];
}

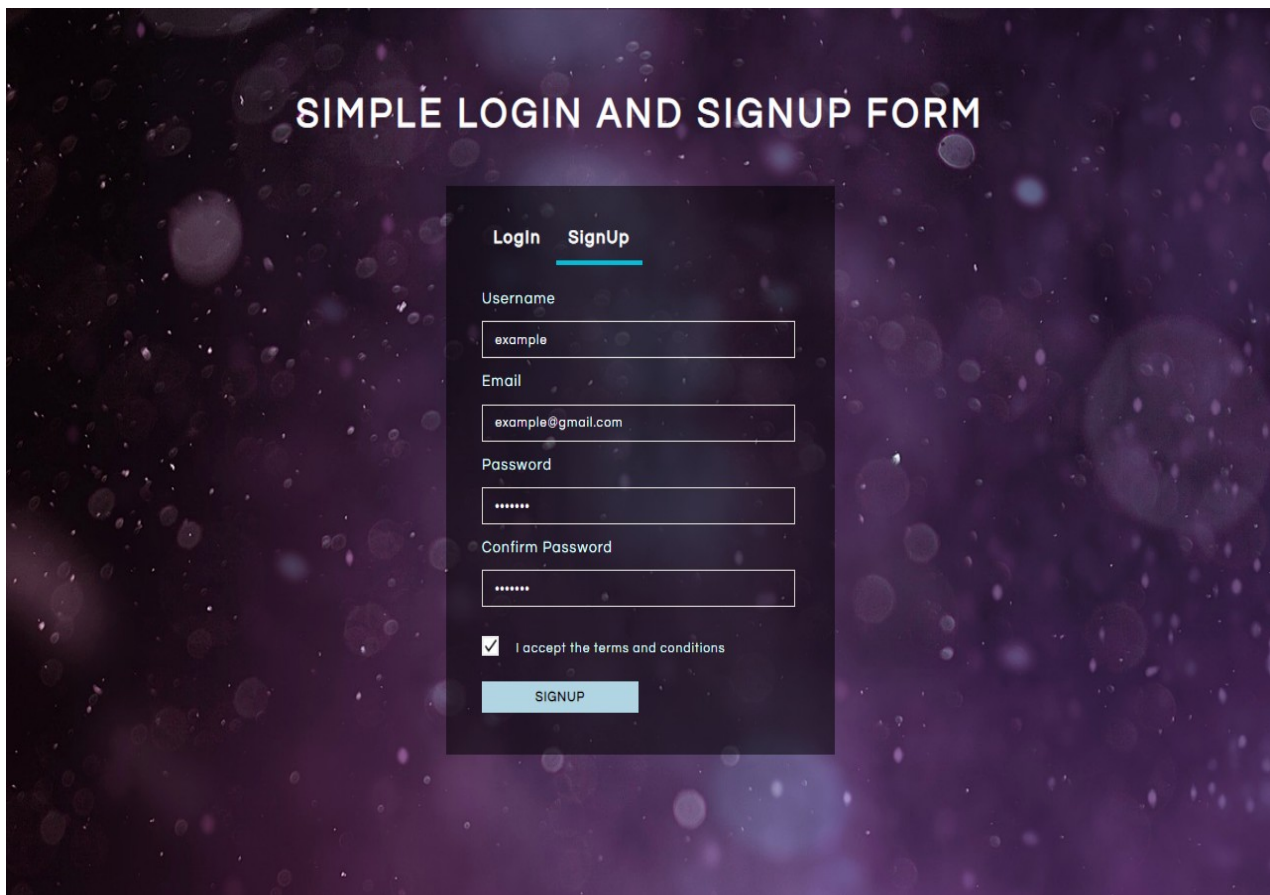
```

6.6 PSYCHIC GAME PAGE

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>The Guessing Game</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
    <script type="text/javascript" src="assets\javascript\game.js"></script>
    <link rel="stylesheet" type="text/css" href="assets\css\reset.css">
    <link rel="stylesheet" type="text/css" href="assets\css\style.css">
  </head>
  <body>
    <nav>
    </nav>
    <nav class="navbar navbar-inverse">
    </nav>
    <div class="container-fluid">
    </div>
    <div class="navbar-header">
      <div>
        <div id="psychicGame">
          <h1>The Psychic Game</h1>
          <p>Guess what letter I'm thinking of:</p>
          <h2 style="display: none;" id="para">correct letter was =
</h2><h2 id="res"></h2>
        </div>
      </div>
    </div>
  </body>
</html>
```

SCREENSHOTS

1. USER REGISTRATION PAGE



The screenshot displays a user registration interface on a dark purple background with a bokeh effect. The title "SIMPLE LOGIN AND SIGNUP FORM" is centered at the top in white. Below the title is a dark gray form container. Inside the container, there are two tabs: "Login" and "SignUp", with "SignUp" being the active tab, indicated by a blue underline. The form includes four input fields: "Username" (containing "example"), "Email" (containing "example@gmail.com"), "Password" (containing seven dots), and "Confirm Password" (containing seven dots). Below these fields is a checkbox labeled "I accept the terms and conditions", which is checked. At the bottom of the form is a blue button labeled "SIGNUP".

Simple Login and Signup Form

Login SignUp

Username
example

Email
example@gmail.com

Password

Confirm Password

☒ I accept the terms and conditions

SIGNUP

2. USER LOGIN PAGE

SIMPLE LOGIN AND SIGNUP FORM

Login **SignUp**

Email

Password


☐ I accept the terms and conditions

LOGIN

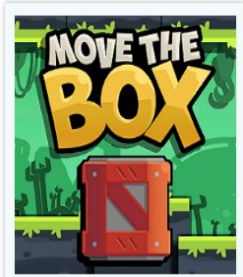
[Forgot Password](#)

3. HOME PAGE

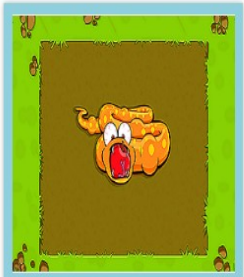
Account | Log In |




FEATURED GAMES




Move The Box



Snake



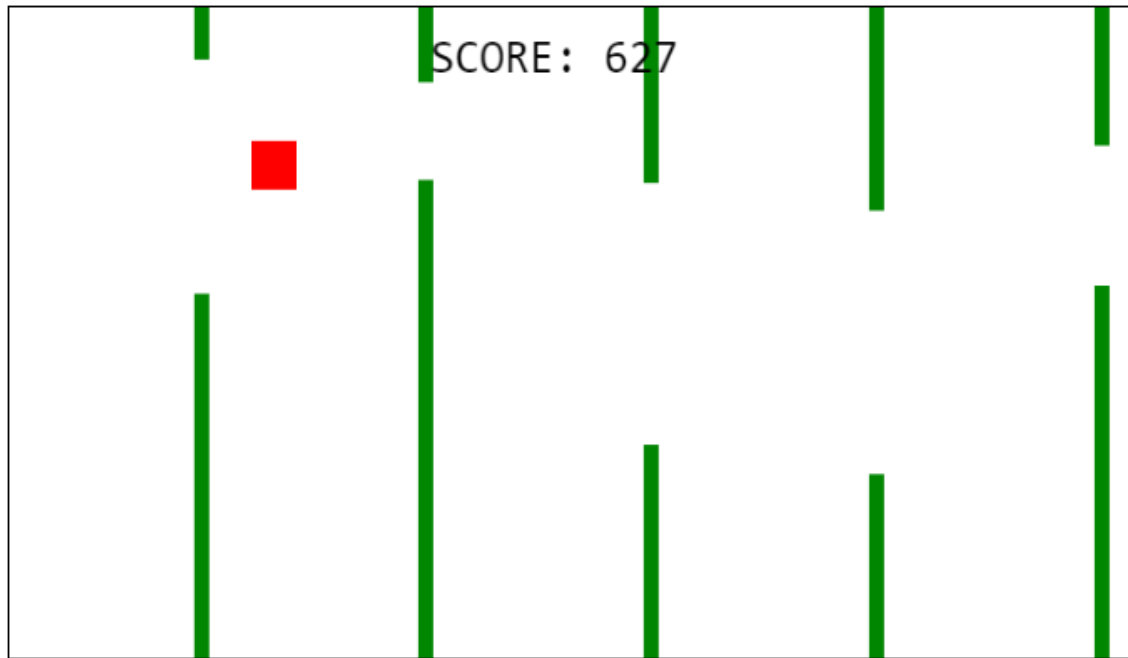
Quiz Time



Psychic Game

Developed by Atul and Anwesha

4. MOVE THE BOX GAME

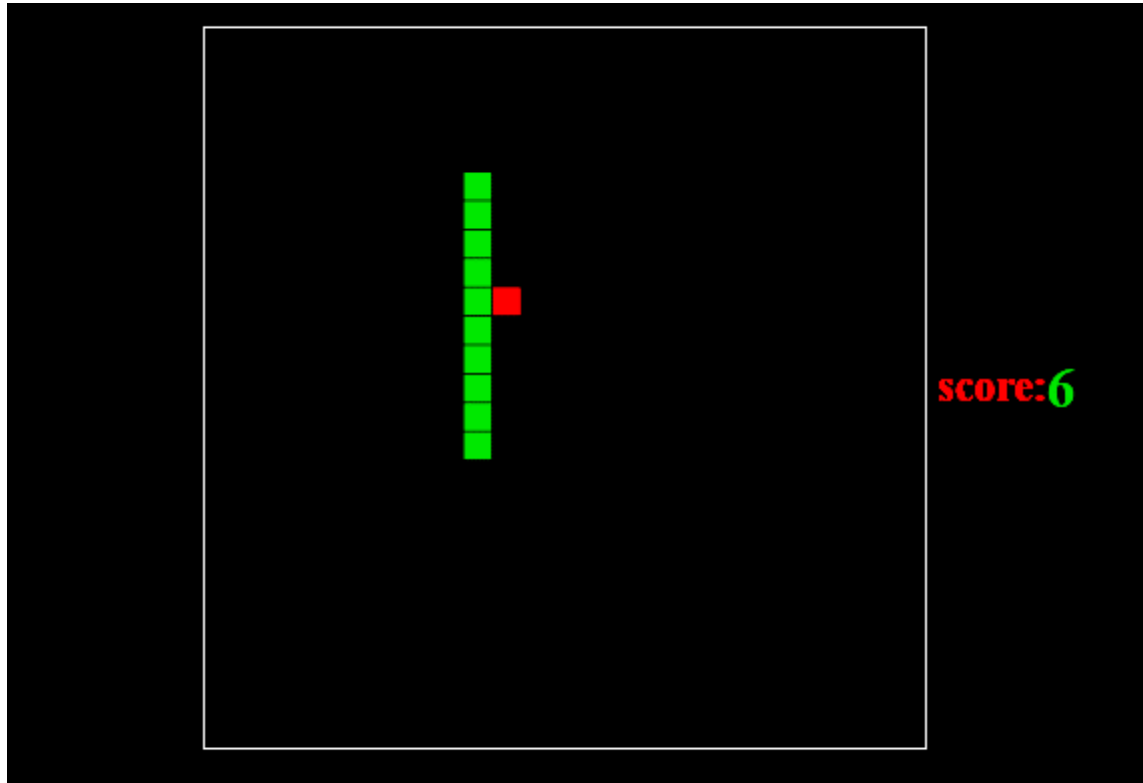


move your arrow keys to move the object box

Your Score is:

SCORE: 627

5. SNAKE XENZIA



6. QUIZ GAME

"The best HTML Quiz"

HTML stands for Hypertext _____ Language.

Html element is the root element of an HTML document.

☒ true
☐ false

Which tag is used to define the html images?

☒ img
☐ src
☐ href
☐ alt

Which of the following is an advantage of putting presentation information in a separate CSS file rather than in HTML itself?

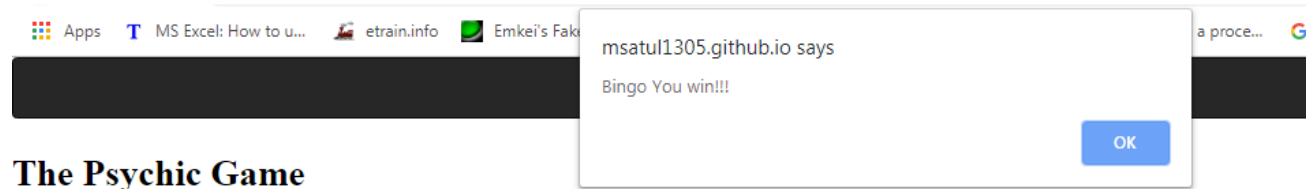
☒ The content becomes easy to manage
☐ Becomes easy to make site for different devices like mobile by making separate CSS files
☐ CSS Files are generally cached and therefore decrease server load and network traffic.
☐ All of the above

Where in HTML document is the conventionally correct place to put reference to an external style sheet?

☒ In the HEAD section
☐ In the BODY section
☐ Both of the Above
☐ None of the Above

☒ That's just okay!
→you got 4 correct.

7. PSYCHIC GAME



The Psychic Game

Guess what letter I'm thinking of:

Wins: 0

Losses: 0

Guesses Left: 8

Your Guesses So Far: a