

ASSIGNMENT 3: JavaScript Event Handling Mechanisms, DOM

LEARNING OBJECTIVES:

- To create a hospital registration form
- To implement a memory matching game using Event Handling mechanisms and DOM

PROGRAM FILES:

Index.html

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
  <title>Patient Reg. Form</title>
  <link href="css/styles.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <h3 id="currentTime" class="titleText">00:00:00</h3>
  <hr>
  <h1 class="titleText">Patient Registration Form</h1>
  <br><br>

  <div class="reg-page">
    <form name="regform" id="regform" method="POST">
      <label for="fname">First Name</label><br>
      <input type="text" id="fname" placeholder="John" size="30"
required onfocus="nameFocus()"
      onblur="nameBlur()" oninvalid="nameInvalid()" pattern="^[a-zA-
Z]+$"
      title="Enter English alphabets only">
      <br><br>

      <label for="lname">Last Name</label><br>
      <input type="text" id="lname" placeholder="Doe" size="30" required
pattern="^[a-zA-Z]+$"
      title="Enter English alphabets only">
      <br><br>

      <label for="phone">Phone Number</label><br>
```

```

        <input type="text" id="phone" placeholder="( +91) 123-456-7890"
size="30" required
        oninvalid="phoneInvalid()">
        <br><br>

        <label for="email">E-Mail</label><br>
        <input type="text" id="email" placeholder="john.doe@email.com"
size="30" required>
        <br><br>

        <label for="dob">Date of Birth</label><br>
        <input type="date" id="dob" required><br><br>

        <label for="age">Age</label><br>
        <input type="number" id="age" min="1" max="120" step="1"
placeholder="21" required
        onkeypress="ageKeyPress()">
        <br><br>

        <label>Gender</label>
        <br>
        <input type="radio" id="male" name="gender" value="Male" required>
        <label for="male">Male</label>
        <input type="radio" id="female" name="gender" value="Female"
required>
        <label for="female">Female</label>
        <input type="radio" id="other" name="gender" value="Other"
required>
        <label for="other">Other</label>

        <br><br>

        <label for="marital">Marital Status</label><br>
        <select id="marital" required onchange="maritalChange()">
            <option>Single</option>
            <option>In A Relationship</option>
            <option>Married</option>
            <option>Separated</option>
            <option>Divorced</option>
            <option>Widowed</option>
        </select>
        <br><br>

        <label for="addr">Address</label><br>
        <textarea cols="30" rows="3" id="addr" required
onselect="addrSelect()"></textarea>
        <br><br>

```

```

<label for="addict">Addictions</label><br>
<textarea cols="30" rows="3" id="addict" required></textarea>
<ul id="addict-list">
    <li draggable="true">Drugs </li>
    <li draggable="true">Alcohol </li>
    <li draggable="true">Behavioral </li>
</ul>
<br><br>

    <input type="submit" class="btn" name="Submit" id="Submit"
value="Register"></input>
    <input type="reset" class="btn" name="Reset" id="Reset"
value="Clear All"></input>
    <input type="button" class="btn" name="Game" id="Game" value="Play
A Game!" onclick="playGame()"></input>
    <input type="button" class="btn" name="Quit" id="Quit"
value="Quit" onclick="exitPage()"></input>

</form>
</div>

```

```

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

</form>
</body>

</html>

```

Game.html

```

<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Memory Game</title>
    <link rel="stylesheet" href="css/gamestyles.css">
  </head>

  <body>
    <aside>
      <h1>&#128377; Memory Game &#128377;</h1>
      <br><hr>
      <br><br>
      <div>&#127919; Level: <span id="level">1</span></div><br>
      <div>&#127993; Score: <span id="score">0</span></div><br>
      <div>&#128737; Time Left: <span id="seconds">0</span></div><br>
    </aside>
  </body>
</html>

```

```

</aside>

<section class="memory-game">
  <div class="memory-card" data-framework="react">
    
    
  </div>
  <div class="memory-card" data-framework="react">
    
    
  </div>
  <div class="memory-card" data-framework="angular">
    
    
  </div>
  <div class="memory-card" data-framework="angular">
    
    
  </div>
  <div class="memory-card" data-framework="apache-hive">
    
    
  </div>
  <div class="memory-card" data-framework="apache-hive">
    
    
  </div>
  <div class="memory-card" data-framework="tableau">
    
    
  </div>
  <div class="memory-card" data-framework="tableau">
    
    

```

```

    </div>
    <div class="memory-card" data-framework="bitcoin">
        

        
    </div>
    <div class="memory-card" data-framework="bitcoin">
        

        
    </div>
    <div class="memory-card" data-framework="gnu">
        
        
    </div>
    <div class="memory-card" data-framework="gnu">
        
        
    </div>
</section>

<script src="js/gamescript.js"></script>
</body>
</html>

```

Table.html

```

var today = new Date();
var time = today.toLocaleTimeString();
document.getElementById("currentTime").innerHTML = time;

const urlParams = new URLSearchParams(window.location.search);

document.getElementById("fnamecell").innerHTML = urlParams.get("fname");
document.getElementById("lnamecell").innerHTML = urlParams.get("lname");
document.getElementById("dobcell").innerHTML = urlParams.get("dob");
document.getElementById("agecell").innerHTML = urlParams.get("age");
document.getElementById("gendercell").innerHTML = urlParams.get("gdr");
document.getElementById("maritalcell").innerHTML = urlParams.get("mstat");
document.getElementById("addictscell").innerHTML = urlParams.get("addn");
document.getElementById("addrcell").innerHTML = urlParams.get("addr");
document.getElementById("phonecell").innerHTML = urlParams.get("phn");
document.getElementById("emailcell").innerHTML = urlParams.get("mail");

```

Game_Script.js

```
//To select all memory card class objects
const cards = document.querySelectorAll(".memory-card");

//To check if user has clicked the first card or the second
let hasFlippedCard = false;
let firstCard, secondCard;

//To lock the board once a flip has been done
let lockBoard = false;

//To maintain game state
let isGameFinished = false;
let score = 0;
let countdowns = [90, 60, 30];
let level = 1;
let count = countdowns[level - 1];

function flipCard(){
    //Flips a chosen card

    if(lockBoard){
        //If the board is locked, do not flip any more cards
        return;
    }

    if(this === firstCard){
        //If the same card is clicked again, don't disable its eventListener
        return;
    }

    this.classList.add("flip");

    if(!hasFlippedCard){
        //First Card clicked
        hasFlippedCard = true;
        firstCard = this;

        return;
    }

    //Second Card clicked
    secondCard = this;

    //console.log(firstCard, secondCard);

    //Check if card matches using data-framework attribute
```

```
//console.log(firstCard.dataset.framework);
//console.log(secondCard.dataset.framework);
checkForMatch();
}

function checkForMatch(){
    //Checks if the two flipped cards match

    if(firstCard.dataset.framework === secondCard.dataset.framework){
        lockBoard = true; //Do not allow to flip other cards before existing
ones are removed
        setTimeout(disableCards, 450);
        updateScore();
    }
    else{
        unflipCards();
    }
}

function disableCards(){
    //Disables the chosen cards

    firstCard.removeEventListener("click", flipCard);
    secondCard.removeEventListener("click", flipCard);

    firstCard.style.setProperty("visibility", "hidden");
    secondCard.style.setProperty("visibility", "hidden");

    resetBoard();
}

function unflipCards(){
    //Unflips the chosen cards

    lockBoard = true; //Disable flipping during this operation

    //Flip within 1.5 seconds
    setTimeout(() => {
        firstCard.classList.remove("flip");
        secondCard.classList.remove("flip");

        resetBoard();
    }, 1200);
}

function resetBoard(){
    //Reset the current board status variables
```

```
[hasFlippedCard, lockBoard] = [false, false];
[firstCard, secondCard] = [null, null];
}

function shuffleCards(){
  //Ordering by CSS order property

  cards.forEach(card => {
    //Random number between 0 - 11
    let randomPos = Math.floor(Math.random() * 12);
    card.style.order = randomPos;
  });
}

function updateScore(){
  //Updates the score and takes care of level progress

  score++;
  document.getElementById("score").innerHTML = score;

  if(score == 6){
    isGameFinished = true;

    setTimeout(() => { //Proceed to next level after 2 seconds
      proceedToNextLevel();
      shuffleCards();
    }, 2000);
  }
}

function proceedToNextLevel(){
  //Proceeds the user to the next level, if he passes the current one

  window.alert("You passed Level " + level + "!");
  level++;
  score = 0;

  //Update the parameters in the webpage
  document.getElementById("level").innerHTML = level;
  document.getElementById("score").innerHTML = score;

  count = countdowns[level - 1];

  //Enable the cards back and remove flip class, and make all cards visible
  cards.forEach(card => card.addEventListener("click", flipCard));
  cards.forEach(card => card.classList.remove("flip"));
  cards.forEach(card => card.style.setProperty("visibility", "visible"));
```



```
    if(level == 4){
        //No more levels, so reload the page
        window.alert("You passed the challenge, congrats!");
        location.reload();
    }
}

//Countdown
var counter = setInterval(runTimer, 1000);

function runTimer(){
    //Runs the timer based on current counter variable value
    count -= 1;

    if(count <= 0){
        clearInterval(counter);    //Stop the counter

        if(score < 6){
            //Reload the page after intimating the user
            window.alert("Sorry, you lost!");
            location.reload();
        }
    }

    //Display the time remaining
    document.getElementById("seconds").innerHTML = count;
}

//Add the eventListener for clicking in the beginning
cards.forEach(card => card.addEventListener("click", flipCard));

//Shuffle the cards for the initial level
shuffleCards();
```

index.js

```
function nameFocus() {
    //On focus, change the text color
    document.getElementById("fname").style.color = "green";
}

function nameBlur() {
    //On blur, change the background color of the field
    document.getElementById("fname").style.backgroundColor = "azure";
}

function nameInvalid() {
    //On invalid, inform the user with alert
```

```
var nameF = document.regform.fname.value;
var nameL = document.regform.lname.value;
var namePattern = new RegExp("[a-zA-Z]+$");

if (nameF == null || nameF == "") {
    window.alert("First Name cannot be left blank!");
}

else if (nameL == null || nameL == "") {
    window.alert("Last Name cannot be left blank!");
}
}

function addrSelect() {
    //If text is selected, change its color
    document.getElementById("addr").style.color = "green";
}

function ageKeyPress() {
    //If key is pressed, change the text's font color & weight
    document.getElementById("age").style.color = "red";
    document.getElementById("age").style.fontWeight = "bold";
}

function maritalChange() {
    //If value is changed, alert the user
    var status = document.getElementById("marital").value;
    window.alert("Your marital status: " + status);
}

function phoneInvalid() {
    //If phone number is invalid, alert the user
    var num = document.getElementById("phone").value;

    if (isNaN(num)) {
        window.alert("Phone number can only consist of numbers!");
    }
}

//Drag and Drop
document.addEventListener("dragstart", function (dragEvent) {
    dragEvent.dataTransfer.setData('Text', dragEvent.target.innerHTML);
});

//Submission event
document.getElementById("regform").addEventListener("submit", function
(submitEvent) {
    submitEvent.preventDefault();    //prevent default action
```

```

//Get the values from the form
var firstName = document.getElementById("fname").value;
var lastName = document.getElementById("lname").value;
var dob = document.getElementById("dob").value;
var age = document.getElementById("age").value;

var genderList = document.getElementsByName("gender");

for (i = 0; i < genderList.length; i++) {
    if (genderList[i].checked == true)
        var gender = genderList[i].value;
}

var maritalStatus = document.getElementById("marital").value;
var additions = document.getElementById("addict").value;
var address = document.getElementById("addr").value;
var phone = document.getElementById("phone").value;
var email = document.getElementById("email").value;

window.open(`submission.html?fname=${firstName}&lname=${lastName}&dob=${dob}&age=${age}&gdr=${gender}&mstat=${maritalStatus}&addn=${additions}&addr=${address}&phn=${phone}&mail=${email}`);

document.getElementById("regform").submit();
});

function playGame() {
    window.open("game.html");
}

function exitPage() {
    window.close();
}

var today = new Date();
var time = today.toLocaleTimeString();
document.getElementById("currentTime").innerHTML = time;

```

Table_script.js

```

//Display time
var today = new Date();
var time = today.toLocaleTimeString();
document.getElementById("currentTime").innerHTML = time;

```

```
//Display the appropriate patient details from the localStorage

//Get the URL from the window
const urlParams = new URLSearchParams(window.location.search);

//Filter the parameters and display it in the table
document.getElementById("fnamecell").innerHTML = urlParams.get("fname");
document.getElementById("lnamecell").innerHTML = urlParams.get("lname");
document.getElementById("dobcell").innerHTML = urlParams.get("dob");
document.getElementById("agecell").innerHTML = urlParams.get("age");
document.getElementById("gendercell").innerHTML = urlParams.get("gdr");
document.getElementById("maritalcell").innerHTML = urlParams.get("mstat");
document.getElementById("addictscell").innerHTML = urlParams.get("addn");
document.getElementById("addrcell").innerHTML = urlParams.get("addr");
document.getElementById("phonecell").innerHTML = urlParams.get("phn");
document.getElementById("emailcell").innerHTML = urlParams.get("mail");
```

Styles.css

```
body {
    background: #76b852;
    background: linear-gradient(to left, #76b852, #8dc26f);
}

.titleText {
    text-align: center;
}

.reg-page {
    width: 500px;
    position: relative;
    z-index: 1;
    background: #ffffff;
    max-width: 500px;
    margin: 0 auto 100px;
    padding: 45px;
    text-align: left;
    box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}

.reg-page input,
.reg-page textarea,
.reg-page select {
    outline: 0;
}
```

```
background: #f2f2f2;
width: 100;
border: 0;
margin: 15px 0px 15px 40px;
padding: 15px;
box-sizing: border-box;
font-size: 16px;
font-family: "Barlow Semi Condensed", sans-serif;
}

.reg-page label {
  font-weight: 650;
}

.reg-page textarea {
  resize: none;
}

#Submit,
#Reset,
#Game,
#Quit {
  font-family: "Barlow Semi Condensed", sans-serif;
  font-size: 16px;
  font-weight: 300;
  text-transform: uppercase;
  outline: 0;
  background: #4caf50;
  width: 200px;
  border: 0;
  padding: 15px;
  color: #ffffff;
  transition: all 0.3 ease;
  cursor: pointer;
  user-select: none;
}

#Submit:hover,
#Reset:hover,
#Game:hover,
#Quit:hover,
#Submit:active,
#Reset:active,
#Game:active,
#Quit:active,
#Submit:focus,
#Reset:focus,
#Game:focus,
```

```
#Quit:focus {  
    font-weight: bold;  
    background: #43a047;  
}  
  
#addict-list li {  
    display: inline-block;  
    appearance: button;  
    padding: 2px 5px;  
    background-color: #76b852;  
    border-radius: 25px;  
}  
  
table,  
td,  
th {  
    font-size: 25px;  
    border: 2px solid black;  
    border-collapse: collapse;  
    text-align: center;  
}  
  
table {  
    width: 75%;  
}  
  
tr:hover {  
    background-color: azure;  
}
```

SCREENSHOTS:

First Name
John

Last Name
Doe

Phone Number
(+91) 123-456-7890

E-Mail
john.doe@email.com

Date of Birth
dd/mm/yyyy

Age
21

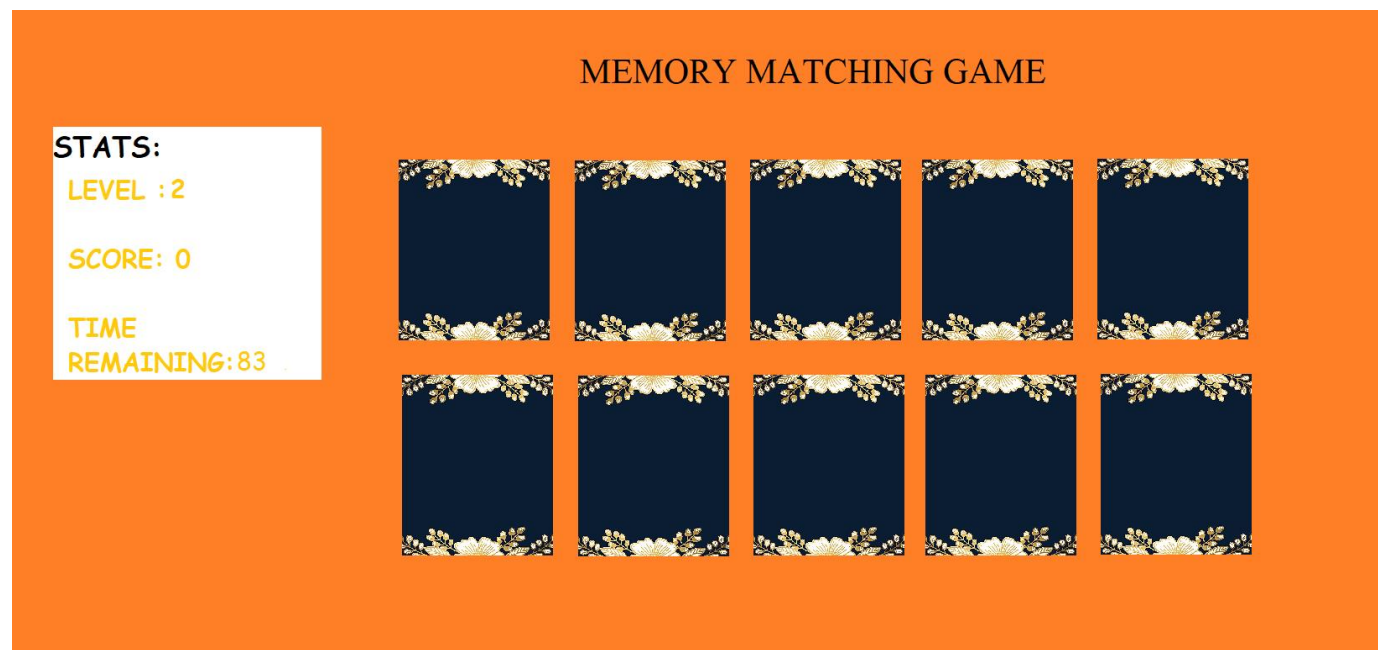
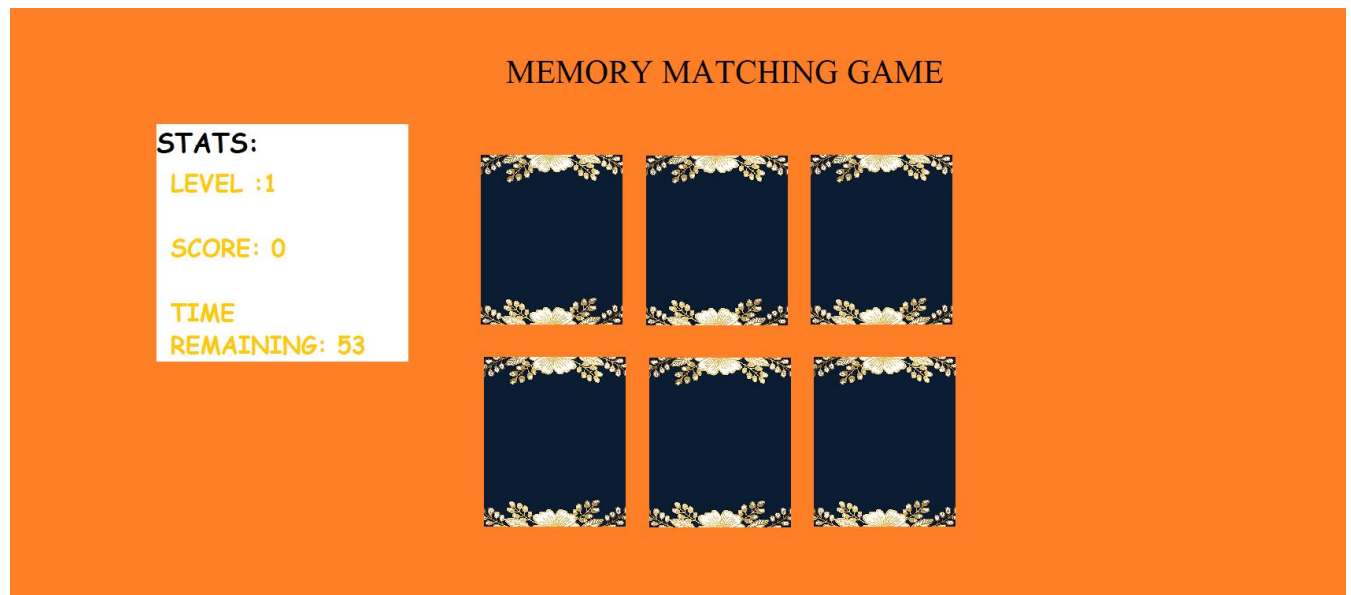
Gender
☐ Male ☐ Female ☐ Other

Marital Status
Single

Address

Addictions

Drugs Alcohol Behavioral Gaming NA

**LEARNING OUTCOMES:**

- Learnt to build a form from scratch using HTML5.
- Learnt to implement drag and drop element in JavaScript

- Learnt to implement user defined functions in JavaScript
- Learnt to implement timers in JavaScript