Ex. No. 5	JavaScript Styles and Animation
Date of Exercise	12/09/2025

Aim

To develop web applications using JavaScript for style modification and animation, demonstrating how JavaScript manipulates CSS styles, user interaction, and object movement.

Description

This exercise illustrates the power of JavaScript in manipulating CSS styles and implementing basic animations.

- In the first experiment, we dynamically change the properties of a `DIV` element such as background color, text color, font size, box width, height, opacity, and border radius using form inputs and JavaScript events.

- In the second experiment, we use JavaScript timers (`setInterval()` and `clearInterval()`) to animate a car image moving across the screen, detect collision with a pillar, and show a warning message.

These two tasks demonstrate DOM manipulation, event handling, and animation control in JavaScript.

Event	Description
onload	Triggers when the document loads
onclick	Triggers on a mouse click
onchange	Triggers when an element changes
onkeyup	Triggers when a key is released
onkeydown	Triggers when a key is pressed
onmouseover	Triggers when the mouse pointer moves over an element
onmousedown	Triggers when a mouse button is pressed
onsubmit	Triggers when a form is submitted

Experiment 1: Style Modification

Code Files

1. A.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <title>Style Modification</title>

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

    <script src="a.js" defer></script>

  </head>

  <body>

    <div id = "box">Welcome Text</div><br>



    background Color : <input type = "color" id = "bgcolor"><br>

    Text Color : <input type = "color"  id = "txtcolor"><br>

    Text Size : <input type = "text" id = "txtsize"><br>

    Box Width : <input type = "text" id = "width"><br>

    Box Height: <input type = "text" id = "height"><br>

    Opacity: <input type="range" min="0" max="1" step="0.1" id="opacity"><br>

    Border Radius : <input type = "range"  min = '0' max = '100' id = 'radius'><br>

  </body>

</html>
```

2. Stylemod.css

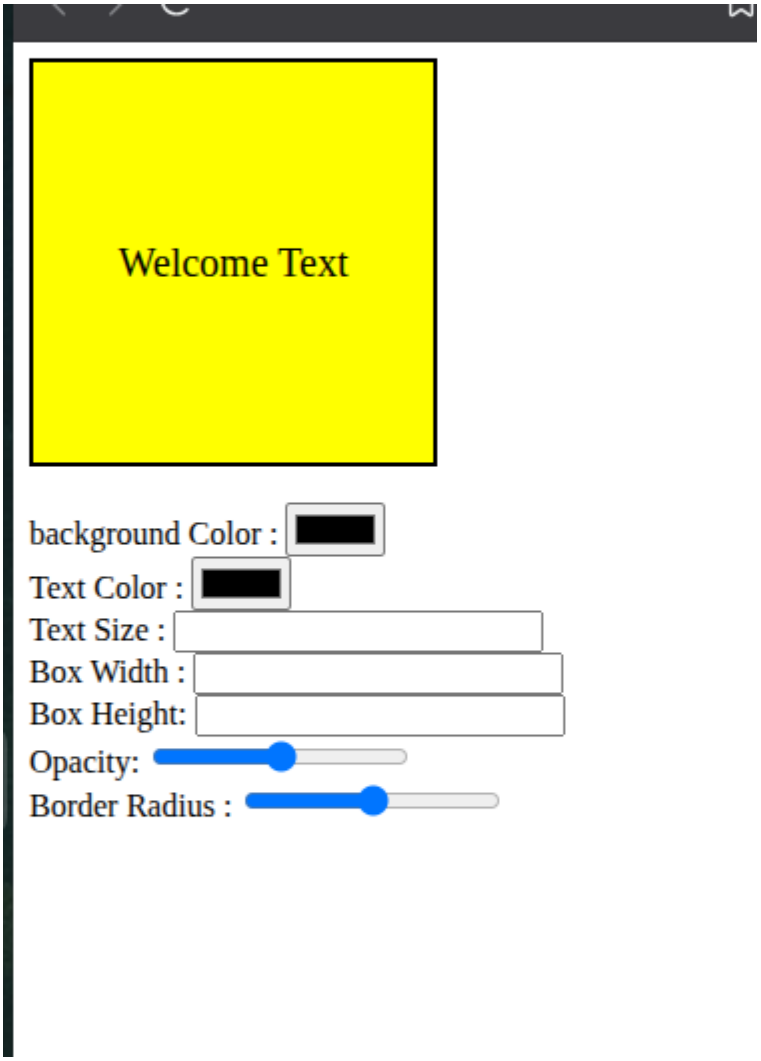
```
#box{
   width: 200px;
   height: 200px;
   background-color: yellow;
   text-align: center;
   line-height: 200px;
   font-size: 20px;
   border: 2px solid black;


}
```

3. stylemod.js

```
function changeStyle() {
   const box = document.getElementById("box");


   box.style.backgroundColor = document.getElementById("bgcolor").value;
   box.style.color = document.getElementById("txtcolor").value;
   box.style.fontSize = document.getElementById("txtsize").value + "px";
   box.style.width = document.getElementById("width").value + "px";
   box.style.height = document.getElementById("height").value + "px";
   box.style.opacity = document.getElementById("opacity").value;
   box.style.borderRadius = document.getElementById("radius").value + "px";
}
```

```
document.querySelectorAll("input").forEach(input => {
  input.addEventListener("change", changeStyle);
  input.addEventListener("keyup", changeStyle);
});
```

Output



Welcome Text

background Color :

Text Color :

Text Size :

Box Width :

Box Height:

Opacity:

Border Radius :

Experiment 2: Car Animation

Code Files

1. `b.html`

```
<!DOCTYPE html>

<html>

  <head>

    <title>

      Car Animantion

    </title>

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

    <script src = "b.js" defer></script>

  </head>

  <body>

    <div id = "danger">Danger</div>

    <img src = "car.png" id = "car">

    <div id = "pillar"></div><br><br><br>


    <button onclick = "startCar()">Start</button>

    <button onclick = "stopCar()">Stop</button>

    <button onclick = "resetCar()">Reset</button>

  </body>

</html>
```

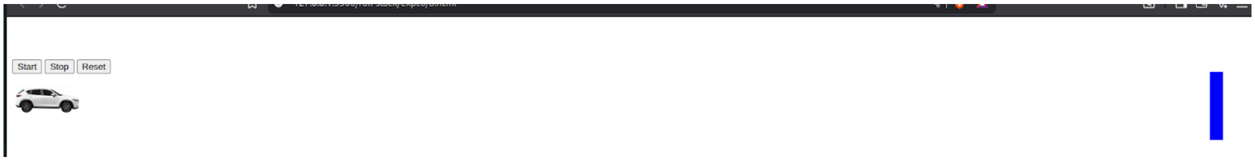
2. `b.css`

```
#car{
   position: absolute;
   left: 10px;
   top: 100px;
   width: 100px;
}

#pillar{
   position : absolute;
   right: 50px;
   top: 80px;
   width: 20px;
   height: 100px;
   background-color: blue;
}

#danger{
   display: none;
   position: absolute;
   top: 50px;
   left: 200px;
   padding: 10px;
   background: red;
   color: white;
   font-weight: bold;
}
```

3. b.js

```
let car = null;
let pos = 10;
let timer;

window.onload = function(){
   car = document.getElementById("car");
};

function startCar(){
   if (!timer) {
      timer = setInterval(moveCar, 20);
   }
}

function moveCar(){
   if(pos >= window.innerWidth - 150){
      clearInterval(timer);
      timer = null;
      document.getElementById("danger").style.display = "block";
   }else{
        pos +=5;

        car.style.left = pos + "px";

    }

}




function stopCar(){

    clearInterval(timer);

    timer = null;

}


function resetCar(){

    clearInterval(timer);

    timer = null;

    pos = 10;

    car.style.left = pos + "px";

    document.getElementById("danger").style.display = "none";

}
```

Output



Result

Thus, the experiments were successfully performed to demonstrate JavaScript style manipulation and animation using HTML, CSS, and JavaScript in separate files.