Week – 5
Introduction to JavaScript

Aim –

To understand the basics of Java Script and learn how to navigate to a particular element using DOM and modify it.

Case Study 1: Updating a Product Price on an E-Commerce Website

You are developing an e-commerce website where product details are displayed dynamically. One of the products listed has a price that needs to be updated due to a discount. Your task is to navigate to the price element using the DOM and modify its value when the "Apply Discount" button is clicked.

Code –

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DOM Navigation - Modify Element</title>
    <style>
      .product {
        border: 2px solid black;
        padding: 20px;
        width: 300px;
        text-align: center;
        margin: 20px auto;
      }
      #price {
        font-size: 20px;
        font-weight: bold;
        color: red;
      }
    </style>
  </head>
  <body>
    <div class="product">
      <h2>Wireless Headphones</h2>
      <p id="price">$100</p>
      <button onclick="applyDiscount()">Apply Discount</button>
    </div>
    <script>
      function applyDiscount() {
        // Navigate to the price element
```

```
      let priceElement = document.getElementById("price");
      // Modify the price content
      priceElement.innerHTML = "$80"; // New discounted price
      // Change style to indicate discount applied
      priceElement.style.color = "green";
      priceElement.style.fontSize = "24px";
    }
  </script>
 </body>
</html>
```

Output -

Case Study 2 : Real-Time Stock Price Update in a Finance Website

You are developing a finance website that displays real-time stock prices for multiple companies. The stock prices change dynamically, and when the "Update Prices" button is clicked, the values for all stocks are updated using document.getElementsByClassName().

Challenge 1: Auto-Update Prices Every 5 Seconds
Challenge 2: Add a "Stop Auto-Update" Button

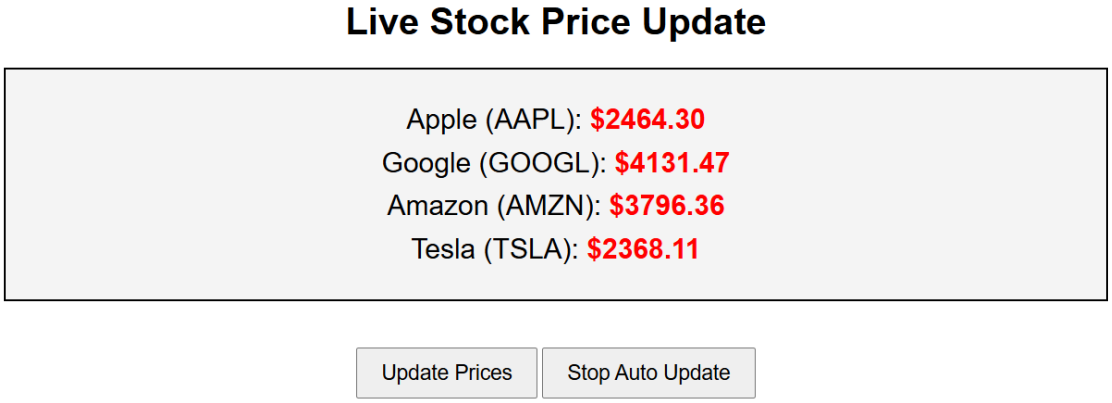
Code –

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Live Stock Price Update</title>
  <style>
   body {
    text-align: center;
    font-family: Arial, sans-serif;
   }
   .stock-container {
    width: 60%;
    margin: 20px auto;
    padding: 20px;
    border: 2px solid black;
```

```
    background-color: #f4f4f4;
   }
   .stock {
    font-size: 24px;
    margin: 10px 0;
   }
   .price {
    color: red;
    font-weight: bold;
   }
   button {
    margin-top: 20px;
    padding: 10px 20px;
    font-size: 18px;
    cursor: pointer;
   }
  </style>
 </head>
 <body>
  <h1>Live Stock Price Update</h1>
  <div class="stock-container">
   <p class="stock">Apple (AAPL): <span class="price">$150</span></p>
   <p class="stock">Google (GOOGL): <span class="price">$2800</span></p>
   <p class="stock">Amazon (AMZN): <span class="price">$3300</span></p>
   <p class="stock">Tesla (TSLA): <span class="price">$900</span></p>
  </div>
  <button onclick="updatePrices()">Update Prices</button>
  <button onclick="stopAutoUpdate()">Stop Auto Update</button>
  <script>
   let autoUpdateInterval;
   function updatePrices() {
    let priceElements = document.getElementsByClassName("price");

    for (let i = 0; i < priceElements.length; i++) {
     let randomPrice = (Math.random() * (5000 - 500) + 500).toFixed(2);

     priceElements[i].innerHTML = "$" + randomPrice;

     priceElements[i].style.color = "green";
     setTimeout(() => {
      priceElements[i].style.color = "red";
     }, 1000);
    }
   }
   function autoUpdate(){
    autoUpdateInterval = setInterval(updatePrices, 5000);
   } function stopAutoUpdate() {
    clearInterval(autoUpdateInterval);
   }
   autoUpdate();
  </script>
```

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

Output -



Case Study 3 : Live Score Update in a Sports Website

You are developing a sports website that displays the live score of a football match. Every time a team scores a goal, the score updates dynamically using the DOM.

Challenge 1: Add a Reset Button
Challenge 2: Add a Match Timer
Challenge 3: Add a Foul Counter

Code –

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Live Football Score</title>
    <style>
      body {
        font-family: Arial, sans-serif;
        text-align: center;
      }
      .scoreboard {
        width: 50%;
        margin: auto;
        padding: 20px;
        border: 2px solid black;
        background-color: #f4f4f4;
      }
      .team {
        font-size: 24px;
        font-weight: bold;
```

```
      margin: 10px;
    }
    .score {
      font-size: 40px;
      color: blue;
    }
    .highlight {
      color: red;
      font-weight: bold;
    }
    button {
      margin: 10px;
      padding: 10px 20px;
      font-size: 18px;
      cursor: pointer;
    }
  </style>
 </head>
 <body>
  <h1>Live Football Score</h1>
  <div class="scoreboard">
   <p class="team">Team A: <span id="scoreA" class="score">0</span></p>
   <p class="team">Team B: <span id="scoreB" class="score">0</span></p>

  </div>
  <button onclick="goalScored('A')">Goal Scored by Team A</button>
  <button onclick="goalScored('B')">Goal Scored by Team B</button>
  <button onclick="reset()">Reset Score</button>
  <div class="timeContainer">Match Time: <span class="timer"
id="matchTimer">0:00</span></div>
  <button onclick="startTimer()">Start Match</button>
  <p class="foul">Fouls Committed by Team A: <span id="foulA"
class="foul">0</span></p>
  <p class="foul">Fouls Committed by Team B: <span id="foulB"
class="foul">0</span></p>
  <button onclick="foulCommited('A')">Foul by Team A</button>
  <button onclick="foulCommited('B')">Foul by Team B</button>
  <script>
   let matchTime = 0;
   function goalScored(team) {
    let scoreElement = document.querySelector("#score" + team);
    let allScores = document.querySelectorAll(".score");

    let currentScore = parseInt(scoreElement.innerHTML);
    scoreElement.innerHTML = currentScore + 1;

    allScores.forEach(score => score.classList.remove("highlight"));

    scoreElement.classList.add("highlight");

    setTimeout(() => {
```

```
        scoreElement.classList.remove("highlight");
      }, 1000);
    }
    function reset(){
      let scores = document.querySelectorAll(".score");
      scores.forEach(score => {score.innerHTML=0;});
    }
    function startTimer() {
      timerInterval = setInterval(function() {
        matchTime++;
        let minutes = Math.floor(matchTime / 60);
        let seconds = matchTime % 60;
        document.getElementById('matchTimer').innerHTML = `${minutes}:${seconds < 10 ?
'0' : ''}${seconds}`;
      }, 1000);
    }
    function foulCommited(team) {
      // Select the correct score element
      let foul = document.querySelector("#foul" + team);
      let allFouls = document.querySelectorAll(".foul");

      // Increase the score by 1
      let foulCount = parseInt(foul.innerHTML);
      foul.innerHTML = foulCount + 1;
    }
  </script>
 </body>
</html>
```

Output -