Course Outcome 4 : Apply JavaScript features such as DOM manipulation, event handling, and callback mechanisms to build responsive client-side applications.

Experiment 07 : Program based on Document Object Model to change the background color of the web page automatically after every 5 seconds.

Problem Statement :

Create a webpage that uses the Document Object Model (DOM) to access and modify the background color of the page. The background color should automatically change to a new color after every 5 seconds without requiring any user action.

Solution :

1. Document Object Model (DOM)

- The **DOM** is a **programming interface** for HTML and XML documents.
- It represents the page as a **tree structure** of nodes, where each element (like <body>, <div>, <h1>) is a node that can be **accessed and modified** using JavaScript.

Example:

document.body.style.backgroundColor = "blue";

- Here, document.body refers to the <body> element in the DOM, and .style.backgroundColor modifies its CSS property directly.

---

2. DOM Manipulation

- Using JavaScript, we can dynamically **modify the style, content, or structure** of a webpage without reloading it.
- For this problem, we manipulate the backgroundColor property of the body.
- DOM manipulation allows web pages to be **interactive and dynamic**, unlike static HTML.

---

3. JavaScript Timer Functions

- JavaScript provides **timing events** to execute code at intervals:

○ setTimeout(function, delay) → runs code **once** after a delay.

○ setInterval(function, delay) → runs code **repeatedly** after every delay interval.

In our case:

setInterval(changeBackgroundColor, 5000);

- executes the changeBackgroundColor() function every **5000 milliseconds = 5 seconds**.

---

4. CSS Object Model (CSSOM) Integration

- When we use document.body.style.backgroundColor, we are working with the **CSSOM** (CSS Object Model).
- The **CSSOM** provides a way for JavaScript to **interact with CSS rules**.
- By using the style property, JavaScript **overrides CSS rules** dynamically, giving us real-time control over the page's appearance.

---

5. Randomization vs. Predefined Colors

There are **two approaches** to color changes:

1. **Predefined Array of Colors** (like in the code earlier) → predictable, controlled cycle.

Random Colors using Math.random() → unpredictable and dynamic. Example:

```
function getRandomColor() {

  let r = Math.floor(Math.random() * 256);

  let g = Math.floor(Math.random() * 256);

  let b = Math.floor(Math.random() * 256);

  return `rgb(${r}, ${g}, ${b})`;

}
```

6. Event Loop & Asynchronous Execution

- JavaScript is **single-threaded**, meaning it executes one task at a time.
- When we use setInterval(), the function is scheduled in the **event queue** and executed asynchronously without blocking the main thread.
- This ensures that the **color keeps changing every 5 seconds** while the page remains responsive.

---

7. Practical Applications

The same concept of **timed DOM updates** can be used in:

- **Slideshows / Carousels** (changing images automatically).
- **Real-time dashboards** (updating stock prices, weather info).
- **Games** (animations, score updates).
- **Attention grabbers** (flashing backgrounds, alerts).