# PRACTICAL: 1

## AIM:

Create a real-time voting system where users can vote on a poll and see the results updated in real-time using only JavaScript, HTML, and CSS.

HTML: A simple poll interface with buttons to vote and display the results.

CSS: Styles the poll and results.

JavaScript:

Defines a vote function to update the local votes.

Updates the vote counts in the UI.

Simulates real-time voting by randomly incrementing votes.

Notes:

Votes object: Keeps track of the current vote counts for each language. It initializes each language with 0 votes.

vote function: This function is called when a button is clicked. It increments the vote count for the selected language and calls updateVotes to refresh the displayed vote counts.

updateVotes function: Updates the displayed vote counts by setting the text content of the spans in the results section to the current vote counts.

## THEORY:

1. HTML (HyperText Markup Language)

HTML is used to create the structure of the voting system. It defines the layout of the poll, including the question, voting buttons, and the section where results are displayed. Each button is associated with a voting option, and result areas are set up to show the current vote counts.

2. CSS (Cascading Style Sheets)

CSS is used to style the poll interface, making it visually appealing and user-friendly. It controls the layout, colors, spacing, and overall appearance of the poll and results sections.

3. JavaScript

JavaScript provides the interactivity and real-time functionality of the voting system. The main concepts used are:

Event Handling: JavaScript listens for button clicks to register votes.

DOM Manipulation: The script updates the displayed vote counts dynamically by changing the content of HTML elements.

State Management: A JavaScript object (e.g., votes) keeps track of the current vote counts for each option.

Functions: Functions like vote() and updateVotes() encapsulate the logic for voting and updating the UI.

setInterval: This function is used to simulate real-time voting by periodically (every 2 seconds) incrementing random vote counts, mimicking votes from other users.

## CODE:

**//INDEX.HTML**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Real-time Voting System</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <div class="container">
    <h1>Vote for Your Favorite Language</h1>
    <div class="buttons">
      <button class="btn" data-lang="javascript">JavaScript</button>
      <button class="btn" data-lang="python">Python</button>
      <button class="btn" data-lang="java">Java</button>
    </div>
    <div class="results">
      <p id="js-result">JavaScript: <span>0</span></p>
      <p id="py-result">Python: <span>0</span></p>
      <p id="java-result">Java: <span>0</span></p>
    </div>
  </div>

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

```
//STYLE.CSS
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
  margin: 0;
}

.container {
  background: white;
  padding: 30px 40px;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0,0,0,0.1);
  text-align: center;
}

h1 {
  margin-bottom: 20px;
}

.buttons button {
  margin: 5px;
  padding: 10px 20px;
  font-size: 16px;
  cursor: pointer;
}

.results p {
  font-size: 18px;
  margin: 10px 0;
}

//SCRIPT.JS
const counts = {
  javascript: 0,
  python: 0,
  java: 0
};

document.querySelectorAll('.btn').forEach(btn => {
  btn.addEventListener('click', function () {
```

```
    const lang = this.getAttribute('data-lang');
    counts[lang]++;
    updateResults();
  });
});

function updateResults() {
  document.querySelector('#js-result span').textContent = counts.javascript;
  document.querySelector('#py-result span').textContent = counts.python;
  document.querySelector('#java-result span').textContent = counts.java;
}
```

**OUTPUT:**


## LATEST APPLICATIONS:

In todays time html css and javascript are great stepping stones to get into the world of learning web development, a lot of simple websites use this tech stack to reduce complexity and cost of hostings.

## LEARNING OUTCOME:

By performing this practical I got a thorough understanding of basic web technologies and terms. And how the frontend and backend connect in a real website

## REFERENCES:

1. JavaScript: https://www.w3schools.com/jsrEF/default.asp
2. Html: https://www.freecodecamp.org/learn/2022/responsive-web-design/learn-html-by-building-a-cat-photo-app/step-66
3. Css: https://www.w3schools.com/css/