**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.

**THEORY:**

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

CSS: Styles the poll and results.

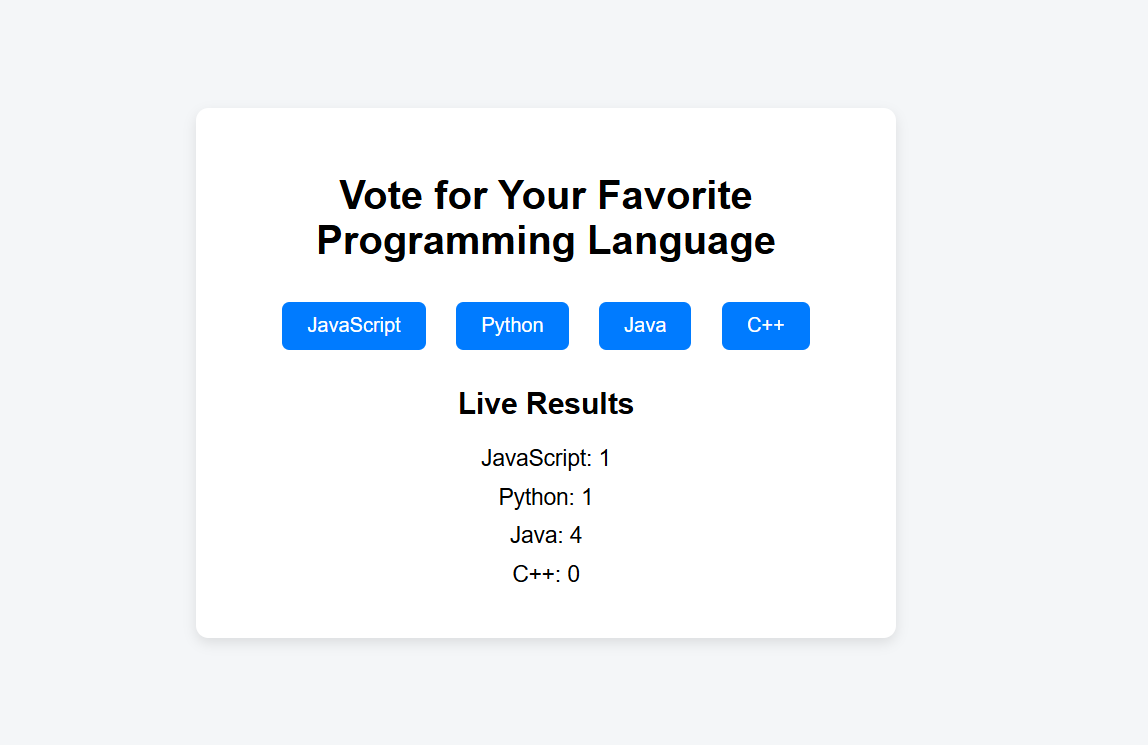
JavaScript: Defines a vote function to update the local votes. Updates the vote counts in the UI.

Simulates real-time voting by randomly incrementing votes.

**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 Programming Language</h1>      <div class="poll">        <button onclick="vote('JavaScript')">JavaScript</button>        <button onclick="vote('Python')">Python</button>        <button onclick="vote('Java')">Java</button>        <button onclick="vote('C++')">C++</button>      </div>      <h2>Live Results</h2>      <div class="results">        <p>JavaScript: <span id="JavaScript">0</span></p>        <p>Python: <span id="Python">0</span></p>        <p>Java: <span id="Java">0</span></p>        <p>C++: <span id="C++">0</span></p>      </div>    </div>    <script src="script.js"></script>  </body>  </html>  **style.css**  body {    font-family: Arial, sans-serif;    background-color: #f4f6f8;    margin: 0;    padding: 0;    height: 100vh;    display: flex;    justify-content: center;    align-items: center;  }  .container {    width: 100%;    max-width: 500px;    background: white;    padding: 30px;    border-radius: 10px;    box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);    text-align: center;  }  .poll button {    margin: 10px;    padding: 10px 20px;    font-size: 16px;    cursor: pointer;    border: none;    border-radius: 6px;    background-color: #007bff;    color: white;    transition: background-color 0.3s;  }  .poll button:hover {    background-color: #0056b3;  }  .results p {    font-size: 18px;    margin: 10px 0;  }  **Script.js**  // Initialize votes  const votes = {    "JavaScript": 0,    "Python": 0,    "Java": 0,    "C++": 0  };  // Vote function  function vote(language) {    if (votes[language] !== undefined) {      votes[language]++;      updateVotes();    }  }  // Update displayed vote counts  function updateVotes() {    for (let lang in votes) {      document.getElementById(lang).textContent = votes[lang];    }  }  // Simulate real-time votes from other users  setInterval(() => {    const languages = Object.keys(votes);    const randomLang = languages[Math.floor(Math.random() \* languages.length)];    votes[randomLang]++;    updateVotes();  }, 2000); |

**OUTPUT:**

****

This project is a simple real-time voting system using HTML, CSS, and JavaScript. Users can vote for programming languages by clicking buttons. Each vote updates the count instantly on the screen. The votes object keeps track of the counts, and the vote() function updates them. The updateVotes() function refreshes the display. A setInterval function simulates real-time votes from other users every 2 seconds, making the results appear live.

**LATEST APPLICATIONS:**

TV Shows – Live audience voting (e.g., Indian Idol).

Online Meetings – Zoom polls, Google Meet.

Events – Apps like Slido for audience interaction.

Social Media – Instagram, YouTube polls.

Education – Kahoot, Quizizz for live quizzes.

**LEARNING OUTCOME:**

Understood how to build interactive web applications using HTML, CSS, and JavaScript.

Learned to update DOM elements dynamically based on user actions.

Gained hands-on experience with JavaScript functions, events, and objects.

Simulated real-time updates using setInterval to mimic live voting.

Improved UI/UX design through basic CSS styling and layout techniques.

**REFERENCES:**

1. [https://www.w3schools.com/](https://www.w3schools.com/%20) – JavaScript Introduction
2. <https://www.geeksforgeeks.org/>– GeeksforGeeks