**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.
* setInterval: Simulates real-time updates by randomly incrementing the vote counts every 2 seconds. This mimics votes coming in from other users in real time.

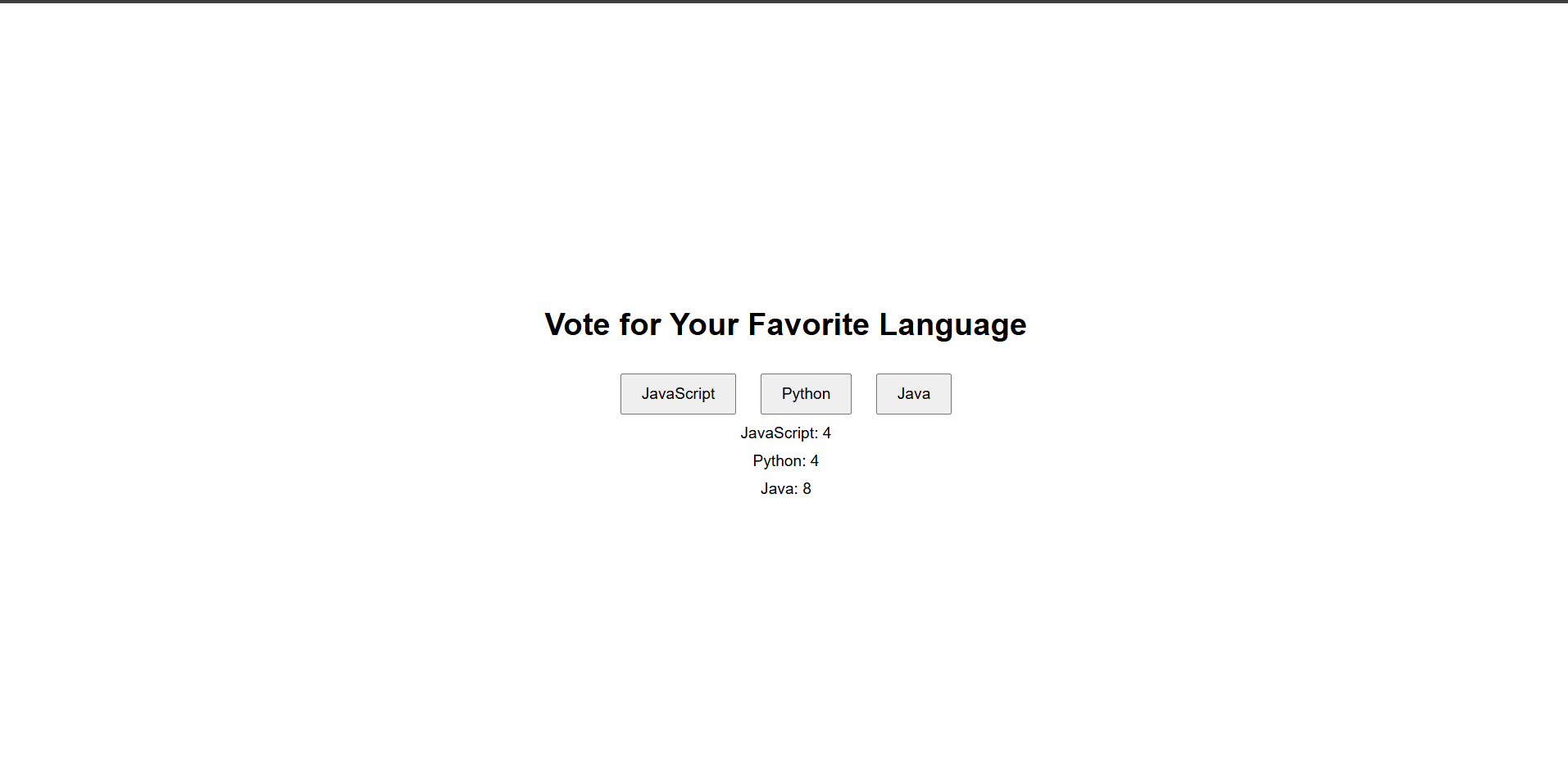
**THEORY:**

This real-time voting system simulates a live polling application where users can vote and view updated results instantly. It uses HTML to create the voting interface, CSS to style the layout, and JavaScript to manage logic and interactivity. A votes object holds and updates vote counts. The vote() function handles user input, and setInterval() simulates real-time updates by randomly increasing votes. This project demonstrates real-time UI updates using only front-end technologies.

**CODE:**

|  |
| --- |
| **HTML:**  <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <title>Real-Time Voting</title>  <link rel="stylesheet" href="style.css">  </head>  <body>  <div class="container">  <h1>Vote for Your Favorite Language</h1>  <div class="buttons">  <button onclick="vote('JavaScript')">JavaScript</button>  <button onclick="vote('Python')">Python</button>  <button onclick="vote('Java')">Java</button>  </div>  <div class="results">  <span id="JavaScript">JavaScript: 0</span>  <span id="Python">Python: 0</span>  <span id="Java">Java: 0</span>  </div>  </div>  <script src="script.js"></script>  </body>  </html>  **CSS:**  body {    margin**:** 0**;**    height**:** 100vh**;**    font-family**:** sans-serif**;**    display**:** flex**;**    justify-content**:** center**;**    align-items**:** center**;**  }  *.container* {    text-align**:** center**;**  }  *.results*{    display**:** flex**;**    flex-direction**:** column**;**    gap **:** 10px**;**  }  button {    margin**:** 10px**;**    padding**:** 10px 20px**;**    font-size**:** 16px**;**  }  **JS:**  const votes **=** {    JavaScript**:** 0**,**    Python**:** 0**,**    Java**:** 0  }**;**  function vote(**language**) {    votes[language]**++;**    updateVotes()**;**  }  function updateVotes() {  **for** (let lang **in** votes) {      document**.**getElementById(lang)**.***textContent* **=** `${lang}: ${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:**

****

**LATEST APPLICATIONS:**

* **Live Event Polling** – Used in webinars and conferences for audience engagement.
* **Online Competitions** – Voting systems in contests like reality shows or e-sports tournaments.
* **Opinion Surveys** – News websites use polls to gather public opinion in real time.
* **Classroom Quizzes** – Teachers use polling tools to check student understanding instantly.
* **Streaming Platforms** – Platforms like Twitch use real-time votes for audience-driven decisions.
* **Interactive Advertisements** – Brands use voting features in ads for market feedback.

**LEARNING OUTCOME:**

* Understood how to manage and update dynamic data using JavaScript.
* Learned to handle user input through event listeners.
* Implemented real-time simulations using setInterval.
* Gained experience in DOM manipulation to reflect real-time changes.
* Strengthened front-end development skills by building an interactive application.

**REFERENCES:**

1. MDN Web Docs – setInterval() – <https://developer.mozilla.org/en-US/docs/Web/API/setInterval>
2. JavaScript Event Handling – <https://developer.mozilla.org/en-US/docs/Web/Events>
3. W3Schools – DOM Manipulation Tutorial – <https://www.w3schools.com/js/js_htmldom.asp>