JavaScript Assignment - 04

Name – Soumitra Anil Kode

Roll no. – 42134 , BE- 06, Batch - P6

DOP :

Git Repo Link :

Source Code :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Experiment 4 - String Comparison (Modern Style)</title>
  <style>
    body {
      font-family: 'Inter', 'Segoe UI', Arial, sans-serif;
      background: linear-gradient(120deg, #e0eafc 0%, #cfdef3 100%);
      margin: 0;
      min-height: 100vh;
      display: flex;
      align-items: center;
      justify-content: center;
    }
    .container {
      backdrop-filter: blur(8px);
      background: rgba(255, 255, 255, 0.95);
      box-shadow: 0 8px 32px #cbdaeb55, 0 1.5px 7px #b0aad444;
      border-radius: 18px;
      max-width: 420px;
      width: 97vw;
      padding: 38px 30px 28px 30px;
      position: relative;
      animation: floatIn 0.85s cubic-bezier(.91,.03,.33,1.02);
    }
    @keyframes floatIn {
      from { opacity: 0; transform: translateY(30px) scale(.96);}
      to { opacity: 1; transform: none;}
    }
    .student-info {
      font-size: 15px;
      color: #3678bb;
      margin-bottom: 16px;
      background: #f2fcff;
      border-radius: 7px;
      padding: 10px 13px;
      display: flex;
  align-items: center;
  gap: 8px;
}
h1 {
  margin: 0 0 7px 0;
  color: #31a4b1;
  letter-spacing: 0.02em;
  font-size: 1.39em;
  font-weight: 700;
}
.theory {
  background: #f6f8fc;
  border-left: 5px solid #3ec6c0;
  border-radius: 7px;
  color: #70562e;
  font-size: 1em;
  margin-bottom: 19px;
  padding: 14px 15px 13px 17px;
  line-height: 1.49;
}
label {
  display: block;
  margin-top: 17px;
  font-weight: 570;
  color: #1a3658;
  font-size: 1em;
}
input[type="text"] {
  width: 100%;
  padding: 10px;
  margin-top: 7px;
  border-radius: 6px;
  border: 2px solid #e2eef7;
  font-size: 1.09em;
  background: #fafdffcc;
  transition: border-color 0.22s;
}
input[type="text"]:focus {
  border-color: #42bdd5;
  outline: none;
  background: #e0f5fd;
}
.btn-group {
  display: flex;
  flex-wrap: wrap;
  gap: 10px;
  margin-top: 15px;
```

```
      margin-bottom: 10px;
      justify-content: center;
    }
    .btn-group button {
      background: linear-gradient(90deg,#3ec6c0 5%,#6db3f2 95%);
      color: #fff;
      padding: 9px 17px;
      border: none;
      border-radius: 6px;
      font-weight: 600;
      cursor: pointer;
      font-size: 1.01em;
      box-shadow: 0 1.5px 7px #bfe8fb44;
      transition: filter 0.16s,transform 0.13s;
      will-change: filter,transform;
    }
    .btn-group button:hover, .btn-group button:focus {
      filter: brightness(1.1) contrast(1.1);
      transform: scale(1.06);
    }
    #result {
      margin-top: 19px;
      padding: 13px 10px 13px 16px;
      background: #f1fefe;
      border-left: 5px solid #3ec6c0;
      min-height: 35px;
      border-radius: 7px;
      color: #22674c;
      font-weight: 650;
      font-size: 1.07em;
      box-shadow: 0 1.2px 5px #edfffa66;
      overflow-x: auto;
      transition: background 0.15s;
    }
    .fade {
      animation: fadeIn 0.7s;
    }
    @keyframes fadeIn {
      from {opacity: 0;}
      to {opacity: 1;}
    }
    @media (max-width: 600px) {
      .container {max-width: 97vw; padding: 13px 4vw;}
    }
  </style>
</head>
<body>
```

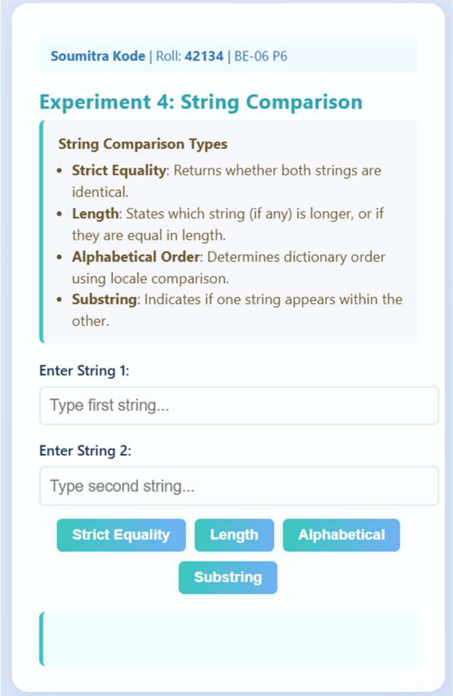
```
  <div class="container">
    <div class="student-info">
      <span><b>Soumitra Kode</b> | Roll: <b>42134</b> | BE-06 P6</span>
    </div>
    <h1>Experiment 4: String Comparison</h1>
    <div class="theory">
      <strong>String Comparison Types</strong><br>
      <ul style="margin:6px 0 0 15px; padding:0;">
        <li><b>Strict Equality</b>: Returns whether both strings are identical.</li>
        <li><b>Length</b>: States which string (if any) is longer, or if they are equal in length.</li>
        <li><b>Alphabetical Order</b>: Determines dictionary order using locale comparison.</li>
        <li><b>Substring</b>: Indicates if one string appears within the other.</li>
      </ul>
    </div>
    <label for="string1">Enter String 1:</label>
    <input type="text" id="string1" placeholder="Type first string...">
    <label for="string2">Enter String 2:</label>
    <input type="text" id="string2" placeholder="Type second string...">
    <div class="btn-group">
      <button onclick="compareStrings('strict')">Strict Equality</button>
      <button onclick="compareStrings('length')">Length</button>
      <button onclick="compareStrings('locale')">Alphabetical</button>
      <button onclick="compareStrings('substring')">Substring</button>
    </div>
    <div id="result"></div>
  </div>
  <script>
    function compareStrings(method) {
      let str1 = document.getElementById("string1").value;
      let str2 = document.getElementById("string2").value;
      let resultBox = document.getElementById("result");
      let result = "";

      if (method === "strict") {
        result = (str1 === str2)
          ? "Both strings are STRICTLY equal."
          : "Strings are NOT strictly equal.";
      }
      else if (method === "length") {
        if (str1.length > str2.length) {
          result = "String 1 is longer than String 2.";
        } else if (str1.length < str2.length) {
          result = "String 2 is longer than String 1.";
        } else {
          result = "Both strings have equal length.";
        }
      }
      else if (method === "locale") {
        let cmp = str1.localeCompare(str2);
        if (cmp < 0) {
          result = "String 1 comes before String 2 alphabetically.";
        } else if (cmp > 0) {
          result = "String 1 comes after String 2 alphabetically.";
        } else {
          result = "Both strings are alphabetically equal.";
        }
      }
      else if (method === "substring") {
        if (str1 && str2 && str1.includes(str2)) {
          result = "String 2 is a substring of String 1.";
        } else if (str1 && str2 && str2.includes(str1)) {
          result = "String 1 is a substring of String 2.";
        } else {
          result = "Neither string is a substring of the other.";
        }
      }

      resultBox.innerHTML = result;
      resultBox.classList.add('fade');
      setTimeout(() => { resultBox.classList.remove('fade'); }, 700);
    }
  </script>
</body>
</html>
```

Output :

A. Initial Web Page Layout –



Soumitra Kode | Roll: **42134** | BE-06 P6

Experiment 4: String Comparison

String Comparison Types

- **Strict Equality**: Returns whether both strings are identical.
- **Length**: States which string (if any) is longer, or if they are equal in length.
- **Alphabetical Order**: Determines dictionary order using locale comparison.
- **Substring**: Indicates if one string appears within the other.

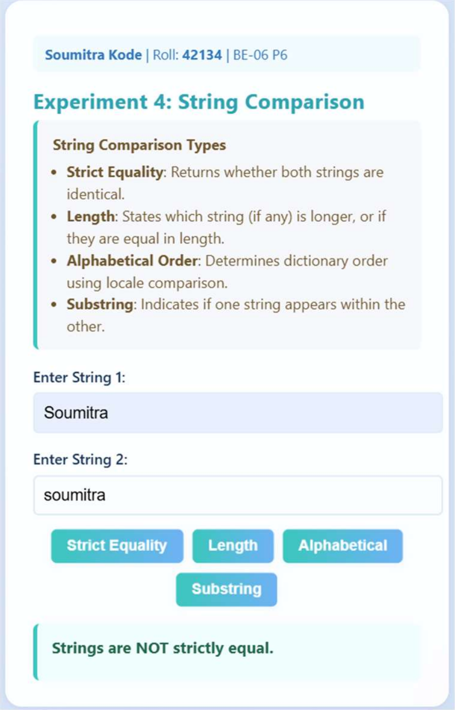
Enter String 1:

Type first string...

Enter String 2:

Type second string...

B. Output for strict equal comparison –



Soumitra Kode | Roll: **42134** | BE-06 P6

Experiment 4: String Comparison

String Comparison Types

- **Strict Equality**: Returns whether both strings are identical.
- **Length**: States which string (if any) is longer, or if they are equal in length.
- **Alphabetical Order**: Determines dictionary order using locale comparison.
- **Substring**: Indicates if one string appears within the other.

Enter String 1:

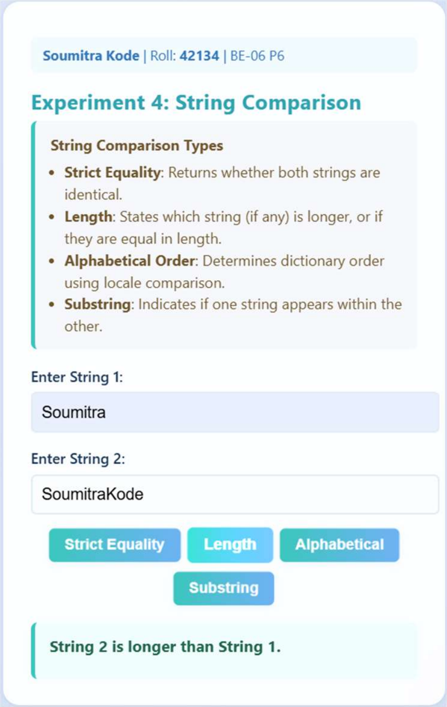
Soumitra

Enter String 2:

soumitra

Strings are NOT strictly equal.

C. Output for comparison of length –



Soumitra Kode | Roll: 42134 | BE-06 P6

Experiment 4: String Comparison

String Comparison Types

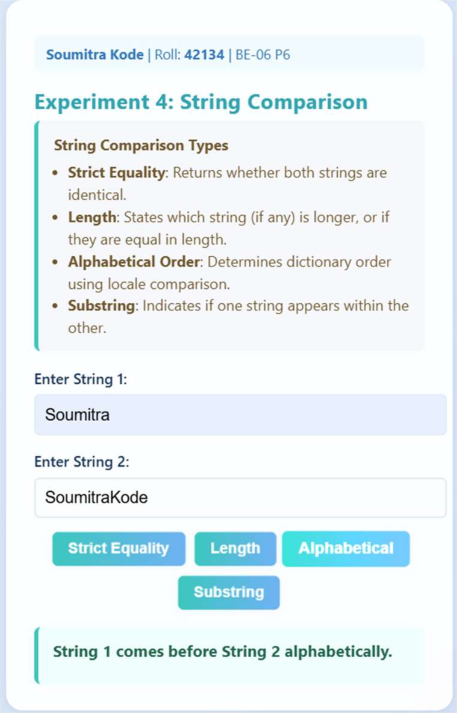
- **Strict Equality**: Returns whether both strings are identical.
- **Length**: States which string (if any) is longer, or if they are equal in length.
- **Alphabetical Order**: Determines dictionary order using locale comparison.
- **Substring**: Indicates if one string appears within the other.

Enter String 1: Soumitra

Enter String 2: SoumitraKode

String 2 is longer than String 1.

D. Output for Alphabetical precedence –



Soumitra Kode | Roll: 42134 | BE-06 P6

Experiment 4: String Comparison

String Comparison Types

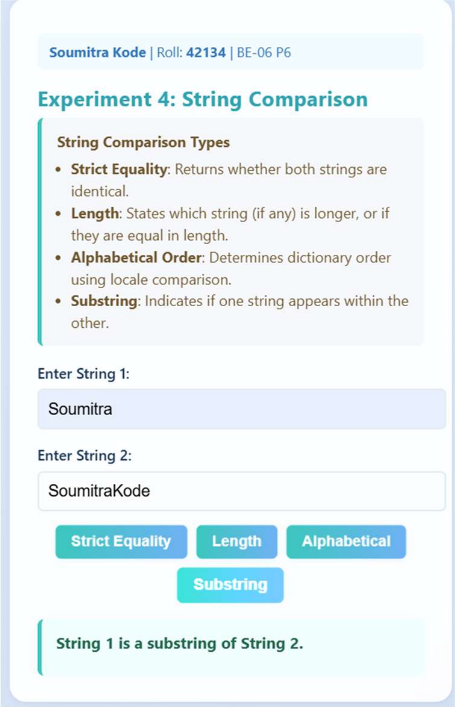
- **Strict Equality**: Returns whether both strings are identical.
- **Length**: States which string (if any) is longer, or if they are equal in length.
- **Alphabetical Order**: Determines dictionary order using locale comparison.
- **Substring**: Indicates if one string appears within the other.

Enter String 1: Soumitra

Enter String 2: SoumitraKode

String 1 comes before String 2 alphabetically.

E. Output for Substring presence –



Soumitra Kode | Roll: **42134** | BE-06 P6

Experiment 4: String Comparison

String Comparison Types

- **Strict Equality**: Returns whether both strings are identical.
- **Length**: States which string (if any) is longer, or if they are equal in length.
- **Alphabetical Order**: Determines dictionary order using locale comparison.
- **Substring**: Indicates if one string appears within the other.

Enter String 1: Soumitra

Enter String 2: SoumitraKode

String 1 is a substring of String 2.