Q31)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Convert First Letter to UpperCase</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
.input {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }

  #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
 </style>
</head>
```

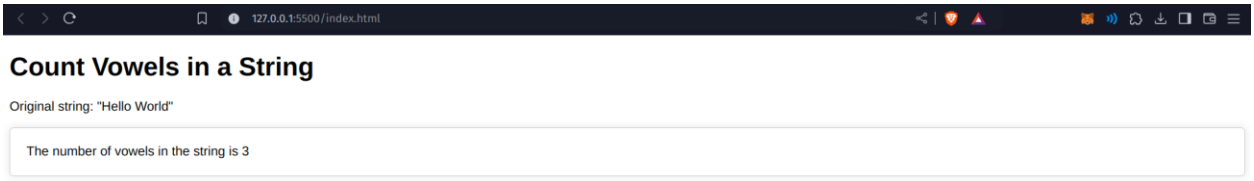
```
<body>
 <h1>Convert First Letter to UpperCase</h1>
<div class = “input”>Original String: hello world</div>
 <div id="output"></div>
 <script>
  let str = "hello world";
  let upperCaseStr = “New String: “+str.charAt(0).toUpperCase() + str.slice(1);
  document.getElementById('output').innerHTML = upperCaseStr;
 </script>
</body>
</html>
```

Q32)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Count Vowels in a String</title>
  <style>
   body {
    font-family: Arial, sans-serif;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
  <h1>Count Vowels in a String</h1>
  <p>Original string: "Hello World"</p>
  <div id="output"></div>
  <script>
   let str = "Hello World";
   let vowels = 'aeiouAEIOU';
   let count = 0;
   for (let i = 0; i < str.length; i++) {
    if (vowels.indexOf(str[i]) !== -1) {
     count++;
    }
```

```
  }
  document.getElementById('output').innerHTML = `The number of vowels in the string is ${count}`;
 </script>
</body>
</html>
```

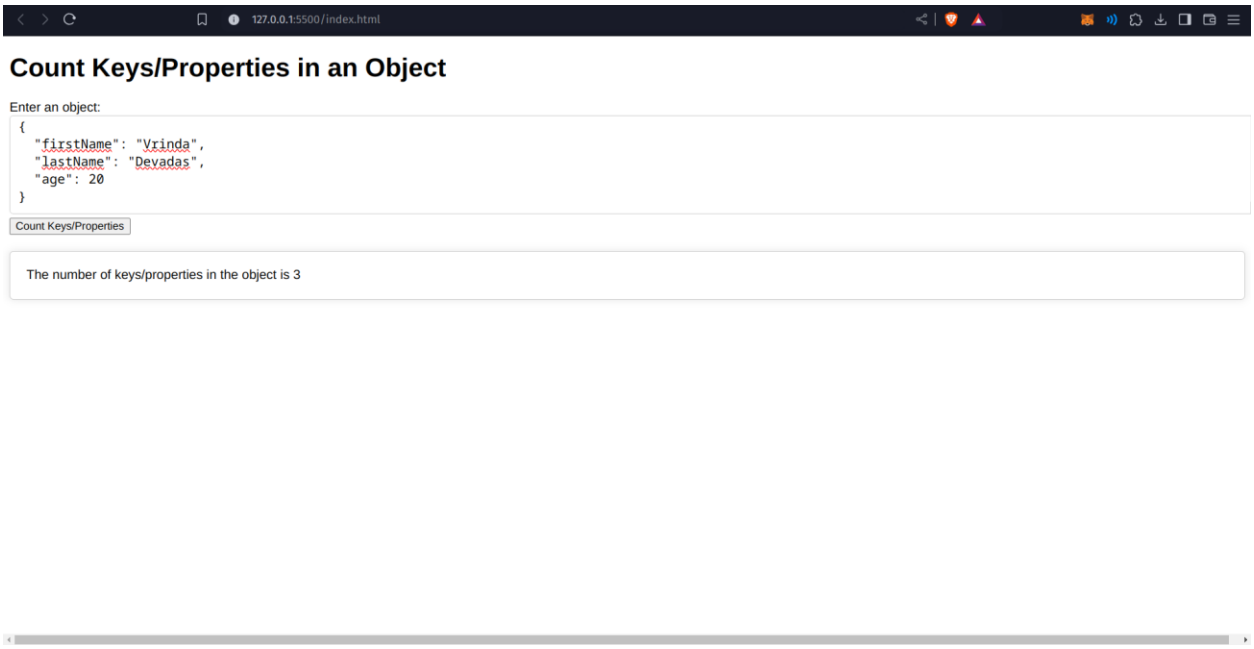


Q33)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Count Keys/Properties in an Object</title>
 <style>
  body {
   font-family: Arial, sans-serif;
```

```
    }
    #input-container {
      margin-bottom: 20px;
    }
    #input-object {
      width: 100%;
      height: 100px;
      padding: 10px;
      font-size: 16px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    #output {
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
  </style>
</head>
<body>
  <h1>Count Keys/Properties in an Object</h1>
  <div id="input-container">
    <label for="input-object">Enter an object:</label>
    <textarea id="input-object" placeholder="e.g. { firstName: 'John', lastName: 'Doe', age: 30 }"></textarea>
```

```
  <button id="count-btn">Count Keys/Properties</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('count-btn').addEventListener('click', countKeys);

  function countKeys() {
   let inputObj = document.getElementById('input-object').value;
   let obj = {};
   try {
    obj = JSON.parse(inputObj);
   } catch (e) {
    alert('Invalid object format. Please enter a valid JSON object.');
    return;
   }
   let count = Object.keys(obj).length;
   document.getElementById('output').innerHTML = `The number of keys/properties in the object is ${count}`;
  }
 </script>
</body>
</html>
```



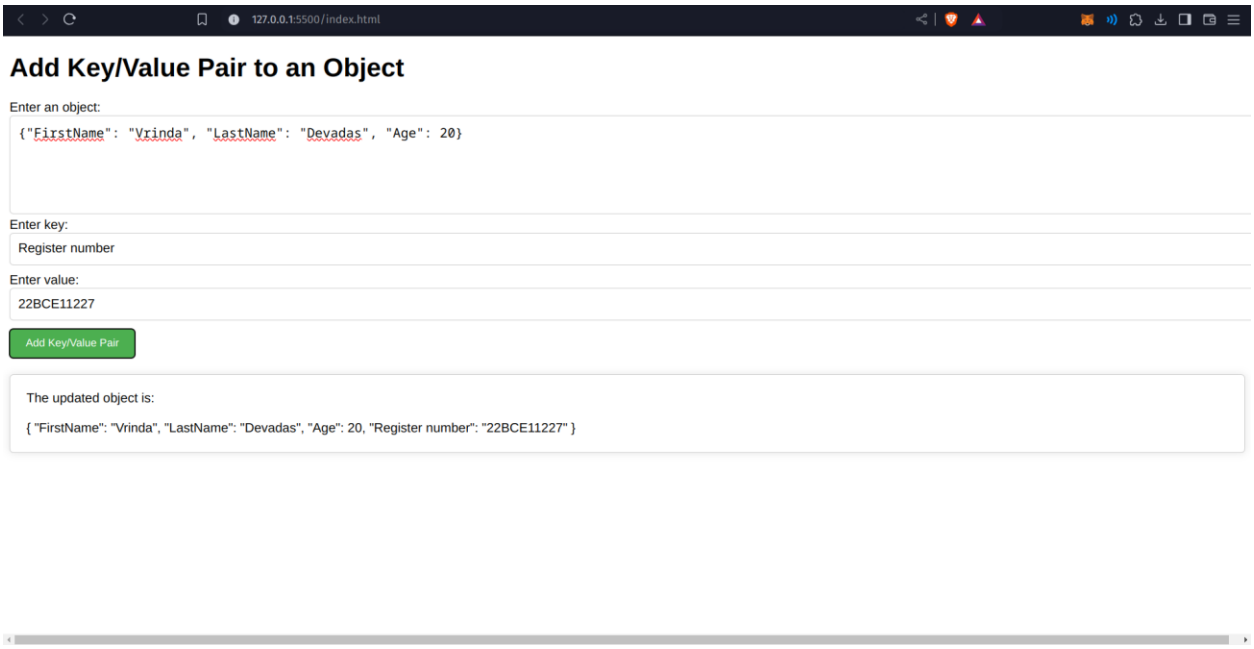
Q34)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Add Key/Value Pair to an Object</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #input-object {
   width: 100%;
```

```
  height: 100px;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#key-input, #value-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
  margin-bottom: 10px;
}
#add-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
#add-btn:hover {
  background-color: #3e8e41;
}
#output {
```

```
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
  <h1>Add Key/Value Pair to an Object</h1>
  <div id="input-container">
   <label for="input-object">Enter an object:</label>
   <textarea id="input-object" placeholder="e.g. { firstName: 'John', lastName:
'Doe' }"></textarea>
   <label for="key-input">Enter key:</label>
   <input type="text" id="key-input" placeholder="e.g. age">
   <label for="value-input">Enter value:</label>
   <input type="text" id="value-input" placeholder="e.g. 30">
   <button id="add-btn">Add Key/Value Pair</button>
  </div>
  <div id="output"></div>

  <script>
   document.getElementById('add-btn').addEventListener('click', addKeyValuePair);

   function addKeyValuePair() {
    let inputObj = document.getElementById('input-object').value;
```

```
    let key = document.getElementById('key-input').value;
    let value = document.getElementById('value-input').value;
    let obj = {};
    try {
      obj = JSON.parse(inputObj);
    } catch (e) {
      alert('Invalid object format. Please enter a valid JSON object.');
      return;
    }
    obj[key] = value;
    document.getElementById('output').innerHTML = `The updated object is: <br><br>
${JSON.stringify(obj, null, 2)}`;
   }
  </script>
</body>
</html>
```



Q35)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Replace All Occurrences of a String</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #input-text, #search-input, #replace-input {
   width: 100%;
```

```
    padding: 10px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 5px;
    margin-bottom: 10px;
   }
   #replace-btn {
    background-color: #4CAF50;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
   }
   #replace-btn:hover {
    background-color: #3e8e41;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
```

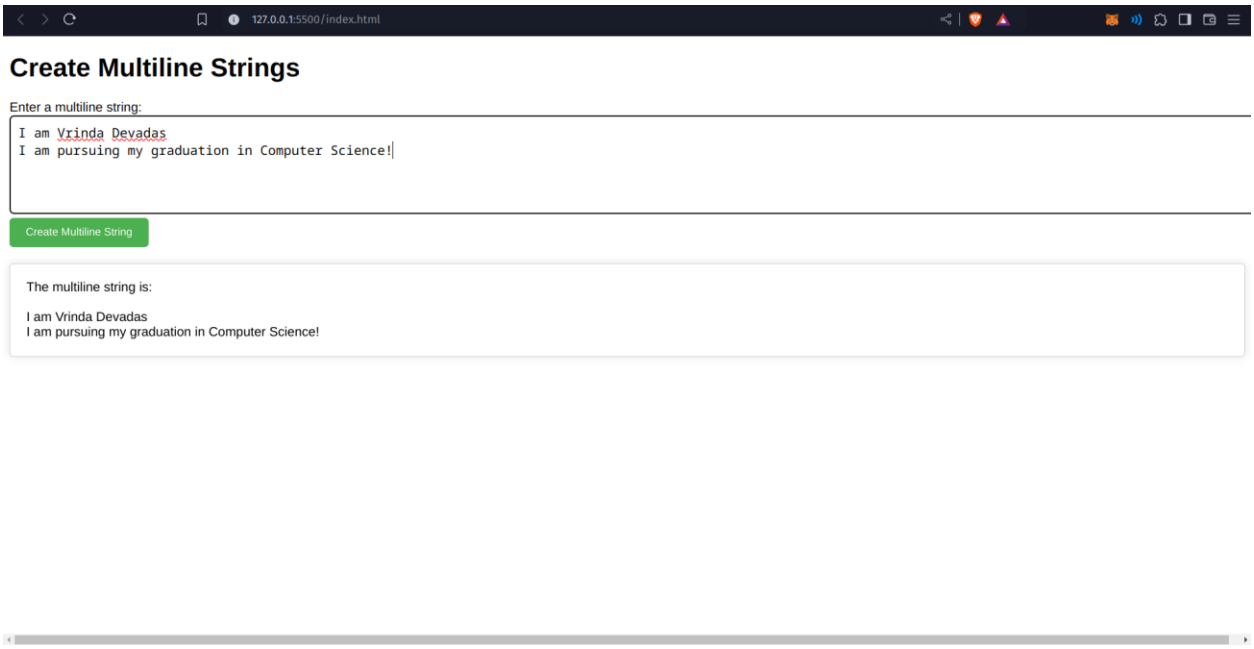
```
 <h1>Replace All Occurrences of a String</h1>
 <div id="input-container">
  <label for="input-text">Enter a text:</label>
  <textarea id="input-text" placeholder="e.g. Hello world, hello again world."></textarea>
  <label for="search-input">Enter a string to search:</label>
  <input type="text" id="search-input" placeholder="e.g. hello">
  <label for="replace-input">Enter a string to replace:</label>
  <input type="text" id="replace-input" placeholder="e.g. hi">
  <button id="replace-btn">Replace</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('replace-btn').addEventListener('click', replaceAll);

  function replaceAll() {
   let inputText = document.getElementById('input-text').value;
   let searchText = document.getElementById('search-input').value;
   let replaceText = document.getElementById('replace-input').value;
   let output = inputText.replace(new RegExp(searchText, 'g'), replaceText);
   document.getElementById('output').innerHTML = `The updated text is: <br><br>
${output}`;
  }
 </script>
</body>
</html>
```

Q36)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Create Multiline Strings</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #input-text {
   width: 100%;
```

```
    height: 100px;
    padding: 10px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 5px;
   }
   #create-btn {
    background-color: #4CAF50;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
   }
   #create-btn:hover {
    background-color: #3e8e41;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
```

```
 <h1>Create Multiline Strings</h1>
 <div id="input-container">
  <label for="input-text">Enter a multiline string:</label>
  <textarea id="input-text" placeholder="e.g. Line 1\nLine 2\nLine 3"></textarea>
  <button id="create-btn">Create Multiline String</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('create-btn').addEventListener('click', createMultilineString);

  function createMultilineString() {
   let inputText = document.getElementById('input-text').value;
   let multilineString = inputText.replace(/\n/g, '<br>');
   document.getElementById('output').innerHTML = `The multiline string is: <br><br>
${multilineString}`;
  }
 </script>
</body>
</html>
```

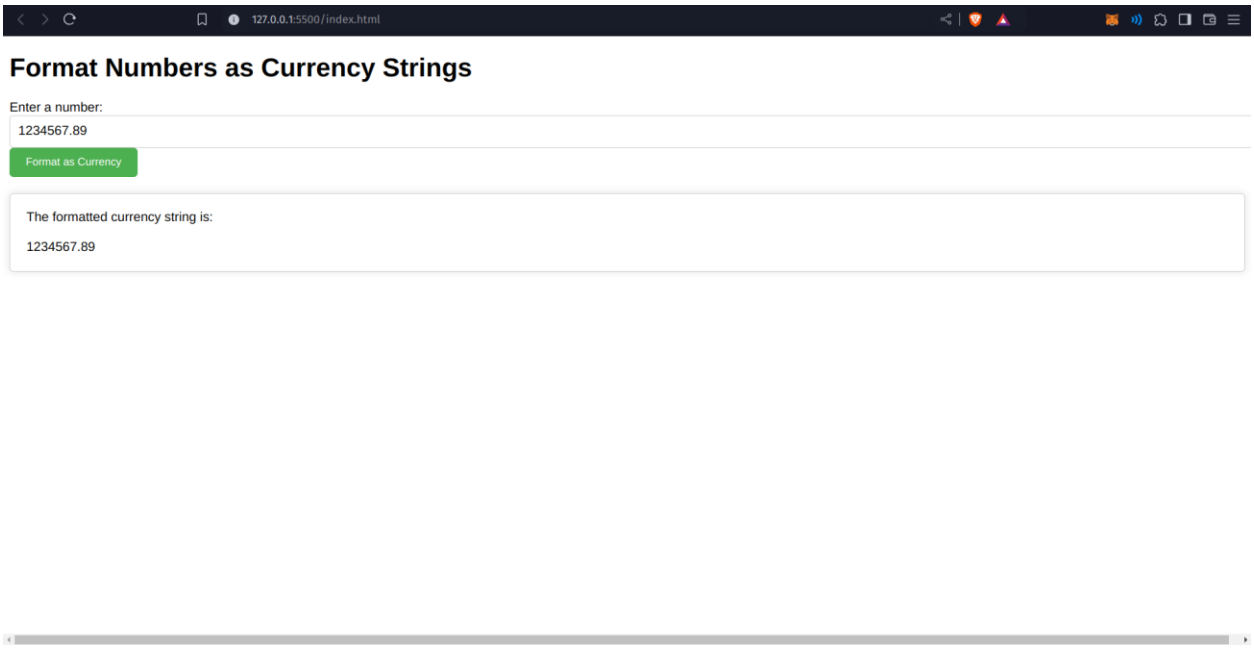


Q37)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Format Numbers as Currency Strings</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #input-number {
   width: 100%;
```

```
    padding: 10px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 5px;
   }
   #format-btn {
    background-color: #4CAF50;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
   }
   #format-btn:hover {
    background-color: #3e8e41;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
  <h1>Format Numbers as Currency Strings</h1>
```

```
 <div id="input-container">
  <label for="input-number">Enter a number:</label>
  <input type="number" id="input-number" placeholder="e.g. 1234567.89">
  <button id="format-btn">Format as Currency</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('format-btn').addEventListener('click', formatAsCurrency);

  function formatAsCurrency() {
   let number = document.getElementById('input-number').value;
   let currencyString = number.toLocaleString('en-US', {
    style: 'currency',
    currency: 'USD',
    minimumFractionDigits: 2
   });
   document.getElementById('output').innerHTML = `The formatted currency string is:
<br><br> ${currencyString}` ;
  }
 </script>
</body>
</html>
```

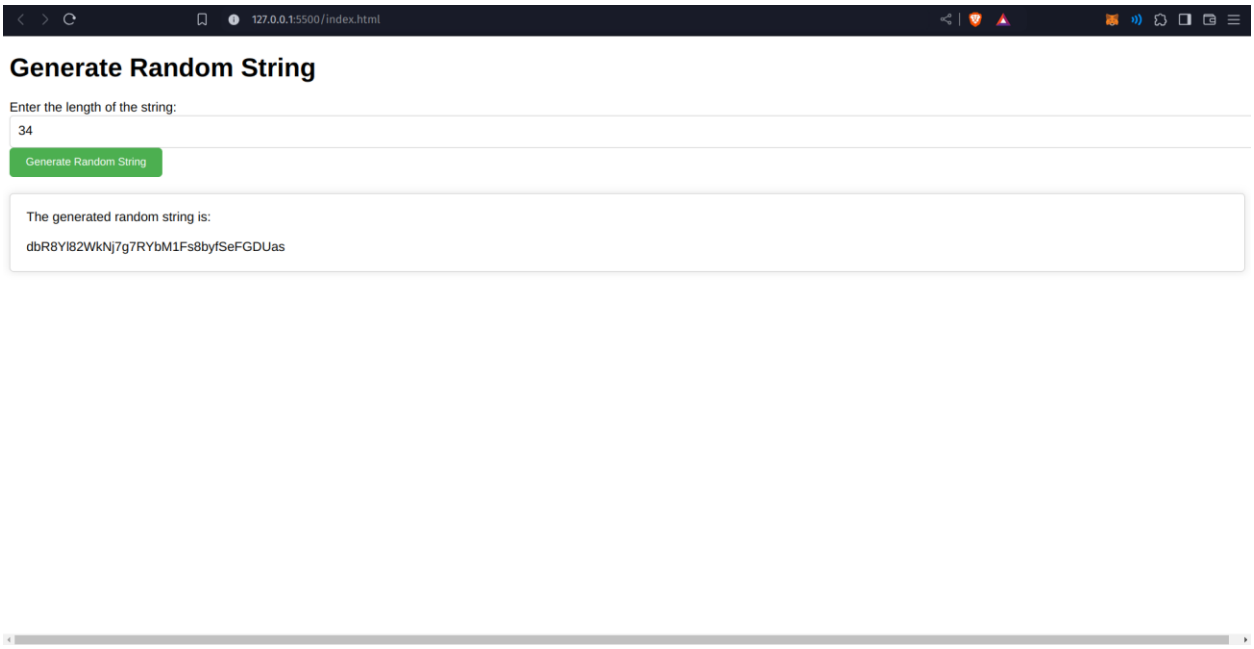


Q38)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Generate Random String</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #length-input {
   width: 100%;
```

```
    padding: 10px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 5px;
   }
   #generate-btn {
    background-color: #4CAF50;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
   }
   #generate-btn:hover {
    background-color: #3e8e41;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
  <h1>Generate Random String</h1>
```

```
 <div id="input-container">
  <label for="length-input">Enter the length of the string:</label>
  <input type="number" id="length-input" placeholder="e.g. 10">
  <button id="generate-btn">Generate Random String</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('generate-btn').addEventListener('click',
generateRandomString);

  function generateRandomString() {
   let length = document.getElementById('length-input').value;
   let characters =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
   let randomString = '';
   for (let i = 0; i < length; i++) {
    randomString += characters.charAt(Math.floor(Math.random() * characters.length));
   }
   document.getElementById('output').innerHTML = `The generated random string is:
<br><br> ${randomString}`;
  }
 </script>
</body>
</html>
```



Q39)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Check if a String Starts With Another String</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #main-string-input {
   width: 100%;
```

```
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#start-string-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#check-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
#check-btn:hover {
  background-color: #3e8e41;
}
#output {
  padding: 20px;
  border: 1px solid #ccc;
```

```
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
  <h1>Check if a String Starts With Another String</h1>
  <div id="input-container">
   <label for="main-string-input">Enter the main string:</label>
   <input type="text" id="main-string-input" placeholder="e.g. Hello World">
   <br><br>
   <label for="start-string-input">Enter the string to check:</label>
   <input type="text" id="start-string-input" placeholder="e.g. Hello">
   <button id="check-btn">Check if String Starts With</button>
  </div>
  <div id="output"></div>

  <script>
   document.getElementById('check-btn').addEventListener('click',
checkIfStringStartsWith);

   function checkIfStringStartsWith() {
    let mainString = document.getElementById('main-string-input').value;
    let startString = document.getElementById('start-string-input').value;
    let result = mainString.startsWith(startString);
    document.getElementById('output').innerHTML = `The result is: <br><br> ${result}`;
```

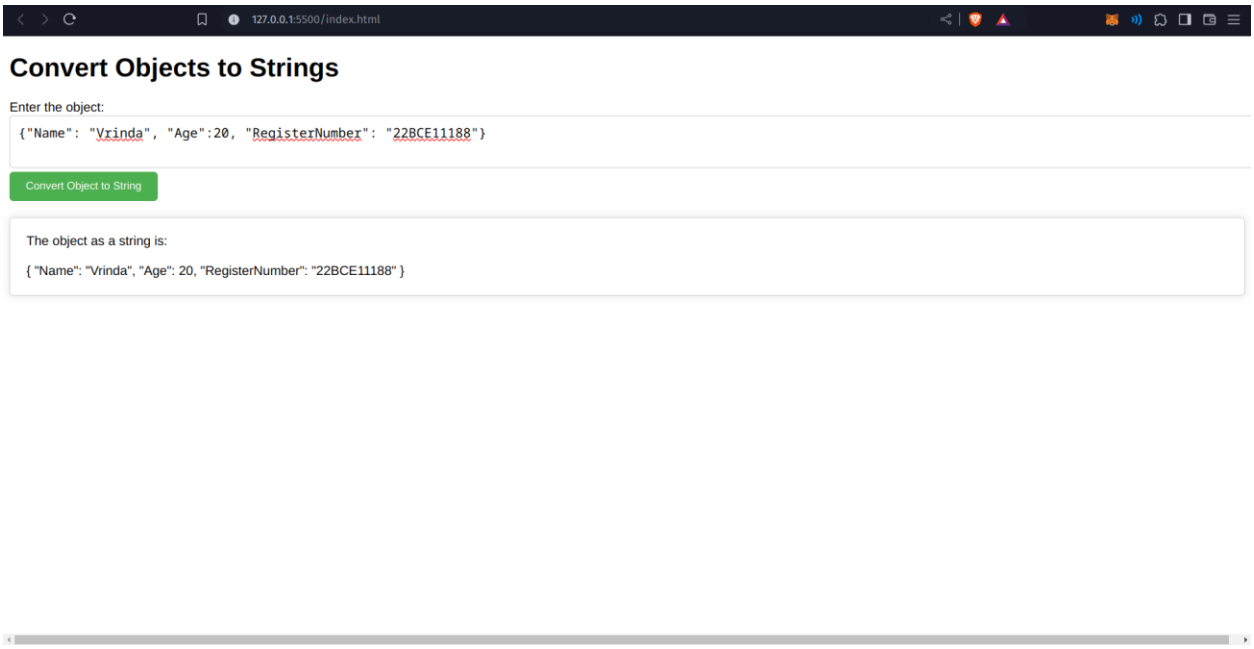
```
  }
 </script>
</body>
</html>
```

Q40)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Convert Objects to Strings</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
```

```
  margin-bottom: 20px;
}
#object-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#convert-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
#convert-btn:hover {
  background-color: #3e8e41;
}
#output {
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
```

```
  </style>
</head>
<body>
  <h1>Convert Objects to Strings</h1>
  <div id="input-container">
    <label for="object-input">Enter the object:</label>
    <textarea id="object-input" placeholder="e.g. { name: 'John', age: 30 }"></textarea>
    <button id="convert-btn">Convert Object to String</button>
  </div>
  <div id="output"></div>

  <script>
    document.getElementById('convert-btn').addEventListener('click',
convertObjectToString);

    function convertObjectToString() {
      let objectInput = document.getElementById('object-input').value;
      let obj = JSON.parse(objectInput);
      let jsonString = JSON.stringify(obj, null, 2);
      document.getElementById('output').innerHTML = `The object as a string is: <br><br>
${jsonString}`;
    }
  </script>
</body>
</html>
```



Q41)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Replace all Instances of a Character in a String</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #string-input {
   width: 100%;
```

```
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#char-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#replace-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#replace-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
```

```
  }
  #replace-btn:hover {
    background-color: #3e8e41;
  }
  #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
 </style>
</head>
<body>
 <h1>Replace all Instances of a Character in a String</h1>
 <div id="input-container">
  <label for="string-input">Enter the string:</label>
  <input type="text" id="string-input" placeholder="e.g. Hello World">
  <br><br>
  <label for="char-input">Enter the character to replace:</label>
  <input type="text" id="char-input" placeholder="e.g. o">
  <br><br>
  <label for="replace-input">Enter the replacement character:</label>
  <input type="text" id="replace-input" placeholder="e.g. x">
  <button id="replace-btn">Replace Character</button>
 </div>
 <div id="output"></div>
```

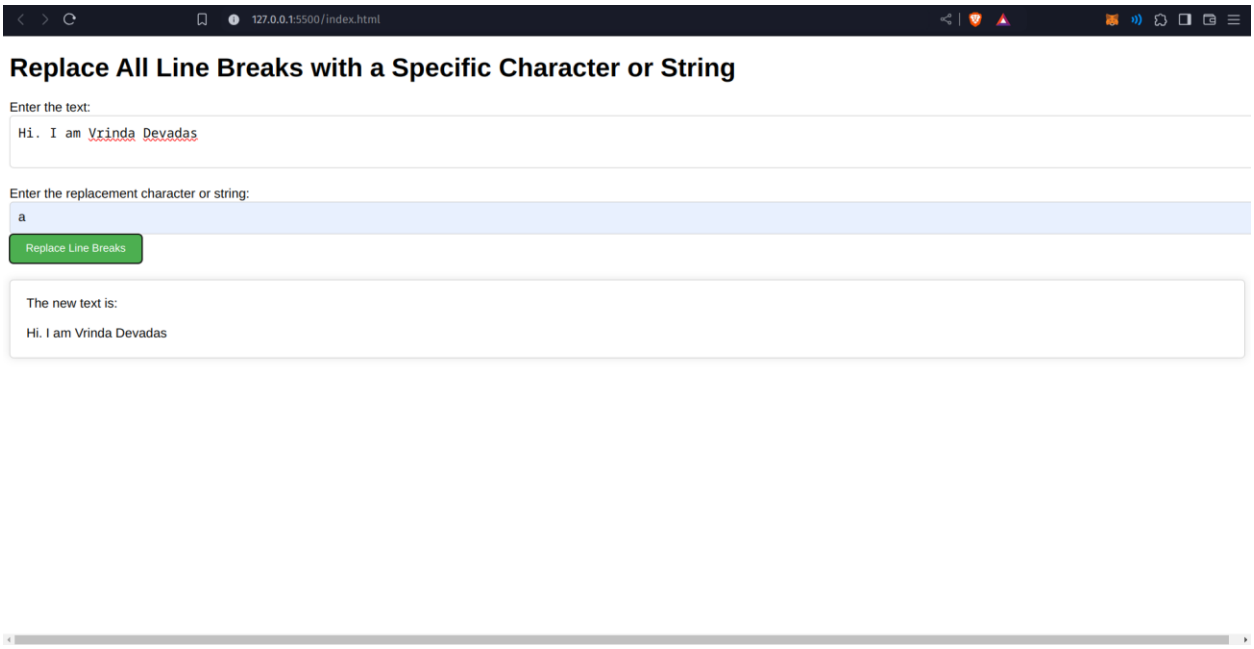
```
 <script>
  document.getElementById('replace-btn').addEventListener('click', replaceCharacter);

  function replaceCharacter() {
    let str = document.getElementById('string-input').value;
    let char = document.getElementById('char-input').value;
    let replaceWith = document.getElementById('replace-input').value;
    let newStr = str.replace(new RegExp(char, 'g'), replaceWith);
    document.getElementById('output').innerHTML = `The new string is: <br><br>
${newStr}`;
  }
 </script>
</body>
</html>
```

Q42)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Replace All Line Breaks with a Specific Character or String</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
  #text-input {
   width: 100%;
   padding: 10px;
   font-size: 16px;
   border: 1px solid #ccc;
   border-radius: 5px;
  }
  #replace-input {
   width: 100%;
   padding: 10px;
   font-size: 16px;
   border: 1px solid #ccc;
```

```
    border-radius: 5px;
   }
   #replace-btn {
    background-color: #4CAF50;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
   }
   #replace-btn:hover {
    background-color: #3e8e41;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
  <h1>Replace All Line Breaks with a Specific Character or String</h1>
  <div id="input-container">
   <label for="text-input">Enter the text:</label>
   <textarea id="text-input" placeholder="e.g. Hello\nWorld"></textarea>
```

```
  <br><br>
  <label for="replace-input">Enter the replacement character or string:</label>
  <input type="text" id="replace-input" placeholder="e.g. |">
  <button id="replace-btn">Replace Line Breaks</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('replace-btn').addEventListener('click', replaceLineBreaks);

  function replaceLineBreaks() {
   let text = document.getElementById('text-input').value;
   let replaceWith = document.getElementById('replace-input').value;
   let newText = text.replace(/\r\n|\n|\r/g, replaceWith);
   document.getElementById('output').innerHTML = `The new text is: <br><br>
${newText}`;
  }
 </script>
</body>
</html>
```

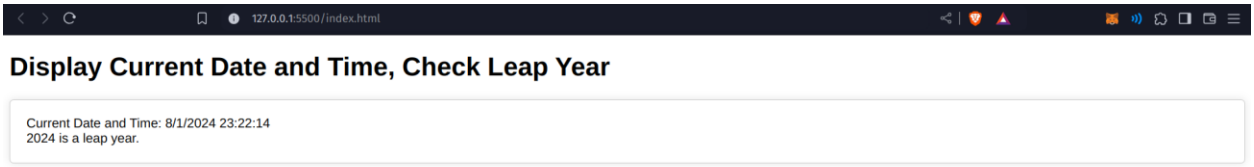


Q43)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Display Current Date and Time, Check Leap Year</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #output {
   padding: 20px;
   border: 1px solid #ccc;
   border-radius: 5px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
```

```
  }
 </style>
</head>
<body>
 <h1>Display Current Date and Time, Check Leap Year</h1>
 <div id="output"></div>

 <script>
  function displayDateTime() {
    let now = new Date();
    let year = now.getFullYear();
    let month = now.getMonth() + 1;
    let day = now.getDate();
    let hour = now.getHours();
    let minute = now.getMinutes();
    let second = now.getSeconds();

    let dateTime = `Current Date and Time: ${month}/${day}/${year}
${hour}:${minute}:${second}`;

    if (isLeapYear(year)) {
      dateTime += `<br> ${year} is a leap year.`;
    } else {
      dateTime += `<br> ${year} is not a leap year.`;
    }
```

```
    document.getElementById('output').innerHTML = dateTime;
  }

  function isLeapYear(year) {
    return (year % 4 === 0 && year % 100 !== 0) || year % 400 === 0;
  }

  displayDateTime();
  setInterval(displayDateTime, 1000); // Update every second
 </script>
</body>
</html>
```



Q44)

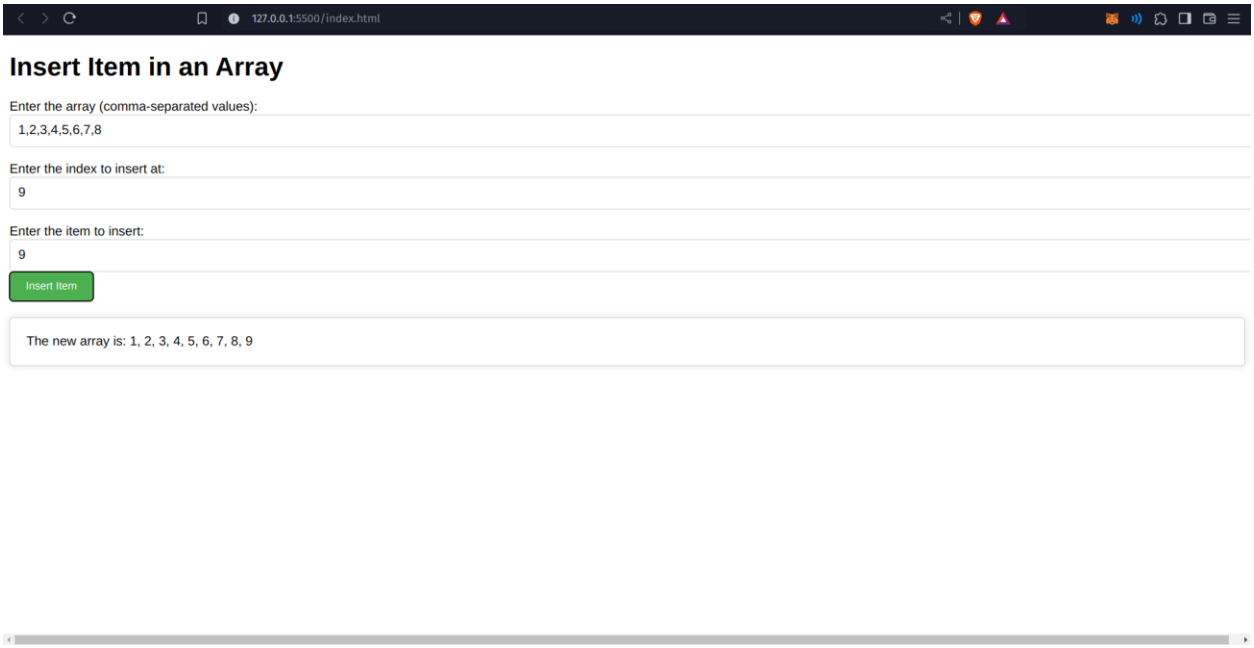
```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Insert Item in an Array</title>
<style>
 body {
  font-family: Arial, sans-serif;
 }
 #input-container {
  margin-bottom: 20px;
 }
 #array-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
 }
 #index-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
 }
 #item-input {
  width: 100%;
```

```
    padding: 10px;
    font-size: 16px;
    border: 1px solid #ccc;
    border-radius: 5px;
   }
   #insert-btn {
    background-color: #4CAF50;
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
   }
   #insert-btn:hover {
    background-color: #3e8e41;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
  </style>
</head>
<body>
  <h1>Insert Item in an Array</h1>
```

```
<div id="input-container">
  <label for="array-input">Enter the array (comma-separated values):</label>
  <input type="text" id="array-input" placeholder="e.g. 1,2,3,4,5">
  <br><br>
  <label for="index-input">Enter the index to insert at:</label>
  <input type="number" id="index-input" placeholder="e.g. 2">
  <br><br>
  <label for="item-input">Enter the item to insert:</label>
  <input type="text" id="item-input" placeholder="e.g. 10">
  <button id="insert-btn">Insert Item</button>
</div>
<div id="output"></div>

<script>
  document.getElementById('insert-btn').addEventListener('click', insertItem);

  function insertItem() {
    let arrayInput = document.getElementById('array-input').value;
    let indexInput = parseInt(document.getElementById('index-input').value);
    let itemInput = document.getElementById('item-input').value;

    let array = arrayInput.split(',');
    array.splice(indexInput, 0, itemInput);

    let output = `The new array is: ${array.join(', ')}`;
    document.getElementById('output').innerHTML = output;
```

```
  }
 </script>
</body>
</html>
```



Q45)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Add Element to Start of an Array</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
```

```
  margin-bottom: 20px;
}
#array-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#item-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#add-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
#add-btn:hover {
  background-color: #3e8e41;
```

```
  }
  #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
 </style>
</head>
<body>
 <h1>Add Element to Start of an Array</h1>
 <div id="input-container">
  <label for="array-input">Enter the array (comma-separated values):</label>
  <input type="text" id="array-input" placeholder="e.g. 1,2,3,4,5">
  <br><br>
  <label for="item-input">Enter the item to add:</label>
  <input type="text" id="item-input" placeholder="e.g. 10">
  <button id="add-btn">Add Item</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('add-btn').addEventListener('click', addToStart);

  function addToStart() {
    let arrayInput = document.getElementById('array-input').value;
```

```
    let itemInput = document.getElementById('item-input').value;

    let array = arrayInput.split(',');
    array.unshift(itemInput);

    let output = `The new array is: ${array.join(', ')}`;
    document.getElementById('output').innerHTML = output;
  }
 </script>
</body>
</html>
```

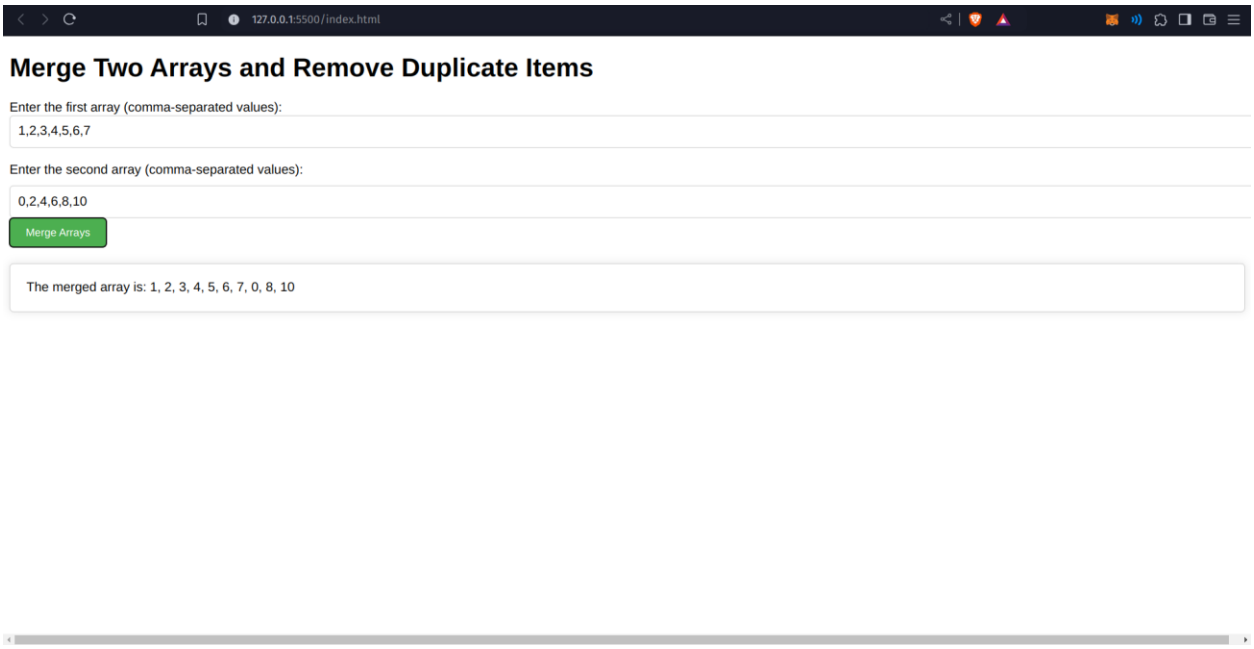
Q46)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Merge Two Arrays and Remove Duplicate Items</title>
<style>
 body {
  font-family: Arial, sans-serif;
 }
 #input-container {
  margin-bottom: 20px;
 }
 #array1-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
 }
 #array2-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
  margin-top: 10px;
 }
 #merge-btn {
  background-color: #4CAF50;
```

```
    color: #fff;
    padding: 10px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
  }
  #merge-btn:hover {
    background-color: #3e8e41;
  }
  #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
 </style>
</head>
<body>
 <h1>Merge Two Arrays and Remove Duplicate Items</h1>
 <div id="input-container">
  <label for="array1-input">Enter the first array (comma-separated values):</label>
  <input type="text" id="array1-input" placeholder="e.g. 1,2,3,4,5">
  <br><br>
  <label for="array2-input">Enter the second array (comma-separated values):</label>
  <input type="text" id="array2-input" placeholder="e.g. 4,5,6,7,8">
  <button id="merge-btn">Merge Arrays</button>
```

```
  </div>
  <div id="output"></div>

  <script>
    document.getElementById('merge-btn').addEventListener('click',
mergeAndRemoveDuplicates);

    function mergeAndRemoveDuplicates() {
      let array1Input = document.getElementById('array1-input').value;
      let array2Input = document.getElementById('array2-input').value;

      let array1 = array1Input.split(',').map(Number);
      let array2 = array2Input.split(',').map(Number);

      let mergedArray = [...new Set([...array1, ...array2])];

      let output = `The merged array is: ${mergedArray.join(', ')}`;
      document.getElementById('output').innerHTML = output;
    }
  </script>
</body>
</html>
```



Q47)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Sort Array of Objects by Property Values</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  #input-container {
   margin-bottom: 20px;
  }
```

```
#array-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#property-input {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
  margin-top: 10px;
}
#sort-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
#sort-btn:hover {
  background-color: #3e8e41;
}
```

```
  #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
 </style>
</head>
<body>
 <h1>Sort Array of Objects by Property Values</h1>
 <div id="input-container">
  <label for="array-input">Enter the array of objects (JSON format):</label>
  <textarea id="array-input" placeholder="e.g. [{name: 'John', age: 25}, {name: 'Jane', age: 30}, {name: 'Bob', age: 20}]"></textarea>
  <br><br>
  <label for="property-input">Enter the property to sort by:</label>
  <input type="text" id="property-input" placeholder="e.g. age">
  <button id="sort-btn">Sort Array</button>
 </div>
 <div id="output"></div>

 <script>
  document.getElementById('sort-btn').addEventListener('click', sortByProperty);

  function sortByProperty() {
   let arrayInput = document.getElementById('array-input').value;
```

```
    let propertyInput = document.getElementById('property-input').value;

    let array = JSON.parse(arrayInput);

    array.sort((a, b) => {
     if (a[propertyInput] < b[propertyInput]) return -1;
     if (a[propertyInput] > b[propertyInput]) return 1;
     return 0;
    });

    let output = `The sorted array is: ${JSON.stringify(array)}`;
    document.getElementById('output').innerHTML = output;
   }
  </script>
</body>
</html>
```

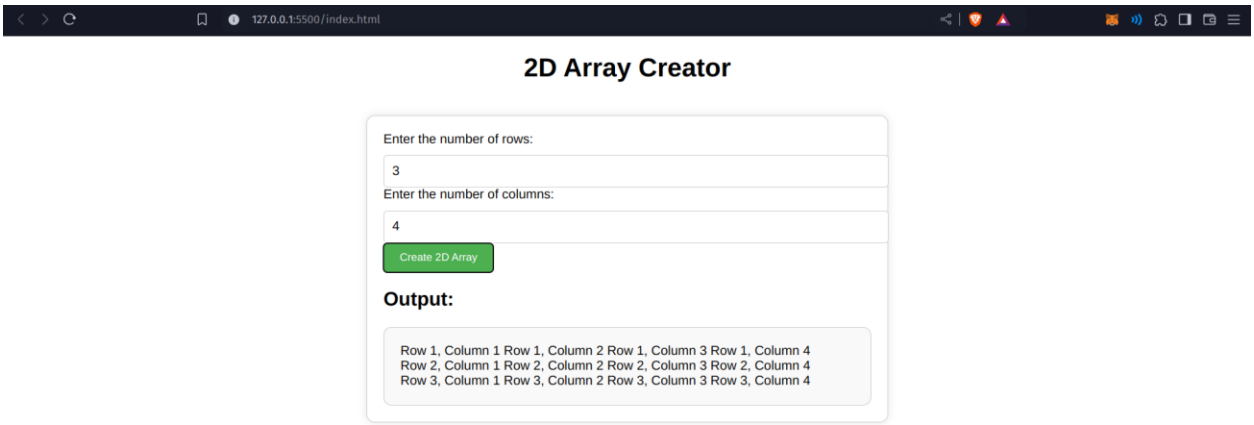
Q48)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>2D Array Creator</title>
 <style>
  body {
   font-family: Arial, sans-serif;
   margin: 20px;
  }
  .container {
   max-width: 600px;
   margin: 40px auto;
   padding: 20px;
   border: 1px solid #ccc;
   border-radius: 10px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  .input-container {
   margin-bottom: 20px;
  }
  label {
   display: block;
   margin-bottom: 10px;
  }
```

```
input[type="number"] {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#create-array-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
#create-array-btn:hover {
  background-color: #3e8e41;
}
.output-container {
  margin-top: 20px;
}
#output {
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 10px;
  background-color: #f9f9f9;
```

```
  }
 </style>
</head>
<body>
 <h1 style="text-align: center;">2D Array Creator</h1>
 <div class="container">
  <div class="input-container">
   <label for="rows">Enter the number of rows:</label>
   <input type="number" id="rows" value="3">
   <label for="columns">Enter the number of columns:</label>
   <input type="number" id="columns" value="4">
   <button id="create-array-btn">Create 2D Array</button>
  </div>
  <div class="output-container">
   <h2>Output:</h2>
   <div id="output"></div>
  </div>
 </div>

 <script>
  document.getElementById('create-array-btn').addEventListener('click', create2DArray);

  function create2DArray() {
   let rows = parseInt(document.getElementById('rows').value);
   let columns = parseInt(document.getElementById('columns').value);
   let twoDimensionalArray = new Array(rows);
```

```
    for (let i = 0; i < rows; i++) {
      twoDimensionalArray[i] = new Array(columns);
    }

    for (let i = 0; i < rows; i++) {
      for (let j = 0; j < columns; j++) {
        twoDimensionalArray[i][j] = `Row ${i + 1}, Column ${j + 1}`;
      }
    }

    let output = '';
    for (let i = 0; i < rows; i++) {
      for (let j = 0; j < columns; j++) {
        output += `${twoDimensionalArray[i][j]} `;
      }
      output += '<br>';
    }

    document.getElementById('output').innerHTML = output;
  }
 </script>
</body>
</html>
```



Q49)

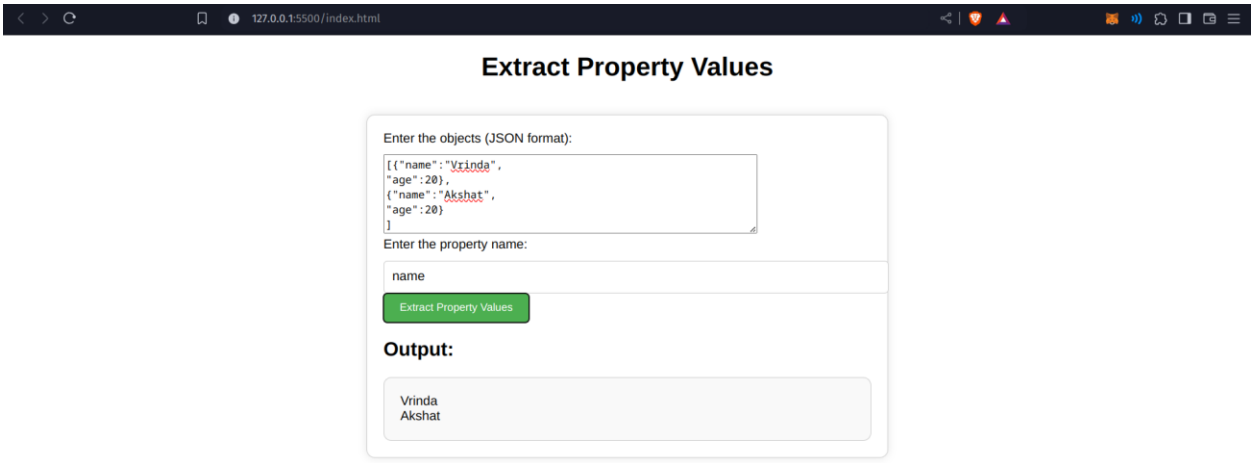
```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Extract Property Values</title>
 <style>
  body {
   font-family: Arial, sans-serif;
   margin: 20px;
  }
  .container {
   max-width: 600px;
   margin: 40px auto;
```

```
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.input-container {
  margin-bottom: 20px;
}
label {
  display: block;
  margin-bottom: 10px;
}
input[type="text"] {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#extract-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
```

```
    }
    #extract-btn:hover {
      background-color: #3e8e41;
    }
    .output-container {
      margin-top: 20px;
    }
    #output {
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 10px;
      background-color: #f9f9f9;
    }
  </style>
</head>
<body>
  <h1 style="text-align: center;">Extract Property Values</h1>
  <div class="container">
    <div class="input-container">
      <label for="objects">Enter the objects (JSON format):</label>
      <textarea id="objects" rows="5" cols="50"></textarea>
      <label for="property">Enter the property name:</label>
      <input type="text" id="property" value="name">
      <button id="extract-btn">Extract Property Values</button>
    </div>
    <div class="output-container">
```

```
    <h2>Output:</h2>
    <div id="output"></div>
   </div>
  </div>

  <script>
   document.getElementById('extract-btn').addEventListener('click',
extractPropertyValues);

   function extractPropertyValues() {
    let objectsText = document.getElementById('objects').value;
    let objects = JSON.parse(objectsText);
    let property = document.getElementById('property').value;
    let result = [];
    for (let obj of objects) {
     if (obj.hasOwnProperty(property)) {
      result.push(obj[property]);
     }
    }
    let output = '';
    for (let value of result) {
     output += `${value}<br>`;
    }
    document.getElementById('output').innerHTML = output;
   }
  </script>
```

```
</body>

</html>
```



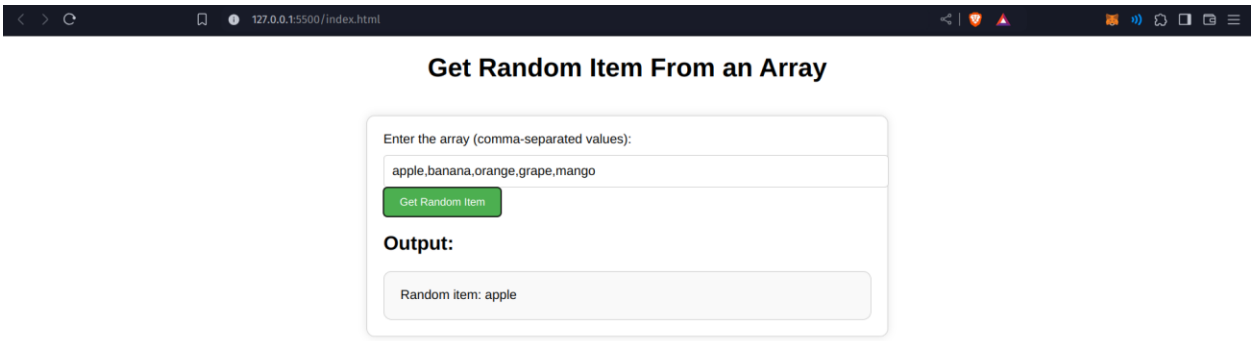
Q50)

```
<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

 <meta name="viewport" content="width=device-width, initial-scale=1.0">

 <title>Get Random Item From an Array</title>

 <style>

  body {

   font-family: Arial, sans-serif;

   margin: 20px;

  }

  .container {

   max-width: 600px;
```

```
  margin: 40px auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.input-container {
  margin-bottom: 20px;
}
label {
  display: block;
  margin-bottom: 10px;
}
input[type="text"] {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#get-random-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
```

```
    cursor: pointer;
   }
   #get-random-btn:hover {
    background-color: #3e8e41;
   }
   .output-container {
    margin-top: 20px;
   }
   #output {
    padding: 20px;
    border: 1px solid #ccc;
    border-radius: 10px;
    background-color: #f9f9f9;
   }
  </style>
</head>
<body>
  <h1 style="text-align: center;">Get Random Item From an Array</h1>
  <div class="container">
   <div class="input-container">
    <label for="array">Enter the array (comma-separated values):</label>
    <input type="text" id="array" value="apple,banana,orange,grape,mango">
    <button id="get-random-btn">Get Random Item</button>
   </div>
   <div class="output-container">
    <h2>Output:</h2>
```

```
    <div id="output"></div>
  </div>
 </div>

 <script>
  document.getElementById('get-random-btn').addEventListener('click', getRandomItem);

  function getRandomItem() {
   let arrayText = document.getElementById('array').value;
   let array = arrayText.split(',');
   let randomIndex = Math.floor(Math.random() * array.length);
   let randomItem = array[randomIndex];
   document.getElementById('output').innerHTML = ` Random item: ${randomItem}` ;
  }
 </script>
</body>
</html>
```



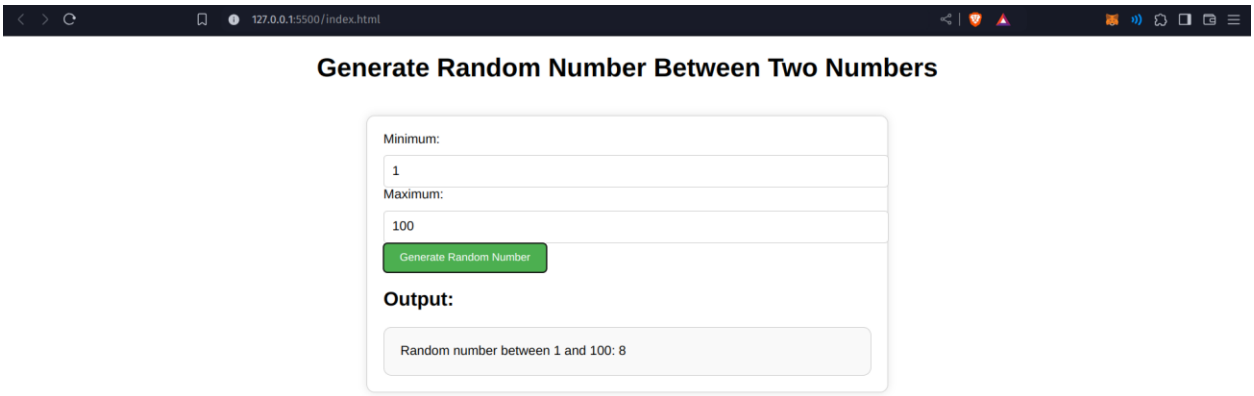
Q51)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Generate Random Number Between Two Numbers</title>
 <style>
  body {
   font-family: Arial, sans-serif;
   margin: 20px;
  }
  .container {
   max-width: 600px;
   margin: 40px auto;
   padding: 20px;
```

```
  border: 1px solid #ccc;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.input-container {
  margin-bottom: 20px;
}
label {
  display: block;
  margin-bottom: 10px;
}
input[type="number"] {
  width: 100%;
  padding: 10px;
  font-size: 16px;
  border: 1px solid #ccc;
  border-radius: 5px;
}
#generate-btn {
  background-color: #4CAF50;
  color: #fff;
  padding: 10px 20px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
```

```
    #generate-btn:hover {
      background-color: #3e8e41;
    }
    .output-container {
      margin-top: 20px;
    }
    #output {
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 10px;
      background-color: #f9f9f9;
    }
  </style>
</head>
<body>
  <h1 style="text-align: center;">Generate Random Number Between Two Numbers</h1>
  <div class="container">
    <div class="input-container">
      <label for="min">Minimum:</label>
      <input type="number" id="min" value="1">
      <label for="max">Maximum:</label>
      <input type="number" id="max" value="100">
      <button id="generate-btn">Generate Random Number</button>
    </div>
    <div class="output-container">
      <h2>Output:</h2>
```

```
    <div id="output"></div>
  </div>
 </div>

 <script>
  document.getElementById('generate-btn').addEventListener('click',
generateRandomNumber);

  function generateRandomNumber() {
    let min = parseInt(document.getElementById('min').value);
    let max = parseInt(document.getElementById('max').value);
    let randomNumber = Math.floor(Math.random() * (max - min + 1)) + min;
    document.getElementById('output').innerHTML = ` Random number between ${min}
and ${max}: ${randomNumber}` ;
  }
 </script>
</body>
</html>
```



Q52)

<!-- index.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Current URL</title>

<style>

/* Add some basic styling to make it look nice */

body {

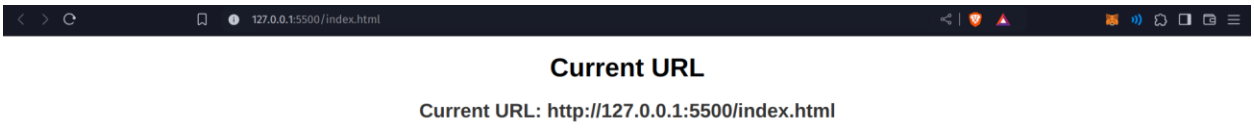
font-family: Arial, sans-serif;

text-align: center;

}

#current-url {

```
        font-size: 24px;
        font-weight: bold;
        color: #333;
      }
    </style>
</head>
<body>
    <h1>Current URL</h1>
    <div id="current-url"></div>

    <script>
      // Get the current URL
      let currentUrl = window.location.href;

      // Display the current URL on the webpage
      document.getElementById("current-url").innerText = `Current URL: ${currentUrl}`;
    </script>
</body>
</html>
```

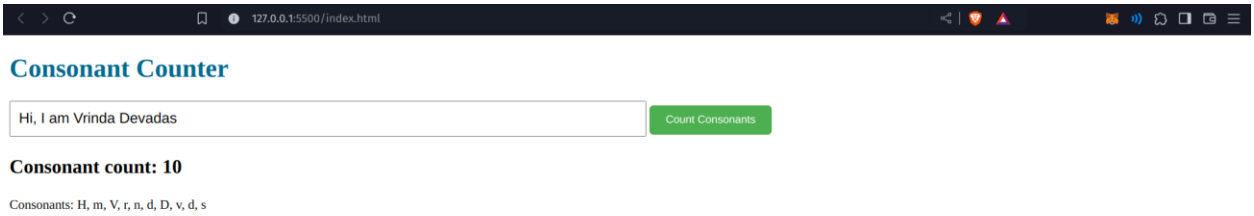


Q53)

```
<!-- index.html -->


<!DOCTYPE html>

<html lang="en">

<head>

 <meta charset="UTF-8">

 <meta name="viewport" content="width=device-width, initial-scale=1.0">

 <title>Consonant Counter</title>

</head>

<body>

 <h1 style="color: #00698f;">Consonant Counter</h1>

 <input type="text" id="input-field" placeholder="Enter a string" style="width: 50%;
padding: 10px; font-size: 18px;">

 <button id="count-button" style="background-color: #4CAF50; color: #fff; padding: 10px
20px; border: none; border-radius: 5px; cursor: pointer;">Count Consonants</button>
```

```
<div id="result-container" style="margin-top: 20px;">
  <p id="consonant-count" style="font-size: 24px; font-weight: bold;"></p>
  <p id="consonants-list"></p>
</div>

<script>
  const consonants = 'bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ';
  const inputField = document.getElementById('input-field');
  const countButton = document.getElementById('count-button');
  const consonantCountElement = document.getElementById('consonant-count');
  const consonantsListElement = document.getElementById('consonants-list');

  countButton.addEventListener('click', () => {
    const inputString = inputField.value;
    let consonantCount = 0;
    let consonantsList = [];

    for (let i = 0; i < inputString.length; i++) {
      if (consonants.includes(inputString[i])) {
        consonantCount++;
        consonantsList.push(inputString[i]);
      }
    }

    consonantCountElement.textContent = `Consonant count: ${consonantCount}`;
    consonantsListElement.textContent = `Consonants: ${consonantsList.join(', ')}`;
```

```
  });
 </script>
</body>
</html>
```

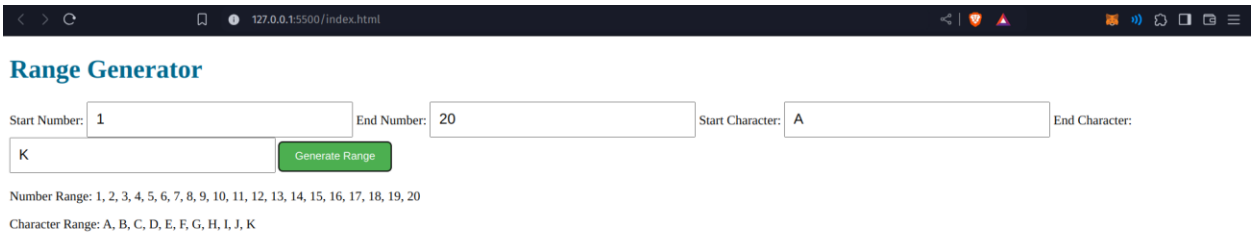


Q54)

```
<!-- index.html -->


<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Range Generator</title>
</head>
<body>
 <h1 style="color: #00698f;">Range Generator</h1>
```

```
  <label for="start-num">Start Number:</label>
  <input type="number" id="start-num" value="1" style="width: 20%; padding: 10px; font-size: 18px;">
  <label for="end-num">End Number:</label>
  <input type="number" id="end-num" value="10" style="width: 20%; padding: 10px; font-size: 18px;">
  <label for="start-char">Start Character:</label>
  <input type="text" id="start-char" value="A" style="width: 20%; padding: 10px; font-size: 18px;">
  <label for="end-char">End Character:</label>
  <input type="text" id="end-char" value="Z" style="width: 20%; padding: 10px; font-size: 18px;">
  <button id="generate-button" style="background-color: #4CAF50; color: #fff; padding: 10px 20px; border: none; border-radius: 5px; cursor: pointer;">Generate Range</button>
  <div id="result-container" style="margin-top: 20px;">
    <p id="number-range"></p>
    <p id="char-range"></p>
  </div>

  <script>
    const generateButton = document.getElementById('generate-button');
    const startNumInput = document.getElementById('start-num');
    const endNumInput = document.getElementById('end-num');
    const startCharInput = document.getElementById('start-char');
    const endCharInput = document.getElementById('end-char');
    const numberRangeElement = document.getElementById('number-range');
    const charRangeElement = document.getElementById('char-range');
```

```
  generateButton.addEventListener('click', () => {
    const startNum = parseInt(startNumInput.value);
    const endNum = parseInt(endNumInput.value);
    const startChar = startCharInput.value.charCodeAt(0);
    const endChar = endCharInput.value.charCodeAt(0);

    let numberRange = [];
    for (let i = startNum; i <= endNum; i++) {
      numberRange.push(i);
    }

    let charRange = [];
    for (let i = startChar; i <= endChar; i++) {
      charRange.push(String.fromCharCode(i));
    }

    numberRangeElement.textContent = `Number Range: ${numberRange.join(', ')}`;
    charRangeElement.textContent = `Character Range: ${charRange.join(', ')}`;
  });
 </script>
</body>
</html>
```



Q55)

```
// Function to calculate area
function calculateArea() {
  // Check the number of arguments
  if (arguments.length === 1) {
    // Circle area
    let radius = arguments[0];
    return Math.PI * radius * radius;
  } else if (arguments.length === 2) {
    // Rectangle area
    let length = arguments[0];
    let width = arguments[1];
    return length * width;
  } else if (arguments.length === 3) {
    // Triangle area
    let base = arguments[0];
```

```
    let height = arguments[1];
    let side = arguments[2];
    return 0.5 * base * height;
  } else {
    throw new Error("Invalid number of arguments");
  }
}

// Test the function
console.log(calculateArea(5)); // Circle area
console.log(calculateArea(4, 5)); // Rectangle area
console.log(calculateArea(3, 4, 5)); // Triangle area
```

Q56)

```
<!-- index.html -->


<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Stack and Queue Implementation</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  .container {
   width: 80%;
   margin: 40px auto;
   padding: 20px;
   border: 1px solid #ddd;
   border-radius: 10px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  .input-field {
   margin-bottom: 20px;
  }
  .input-field label {
```

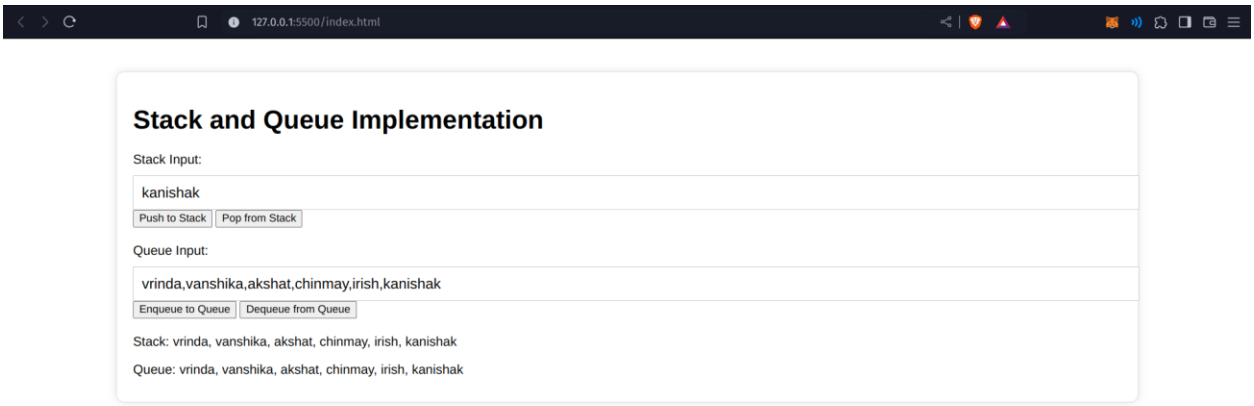
```
    display: block;
    margin-bottom: 10px;
   }
   .input-field input {
    width: 100%;
    padding: 10px;
    font-size: 18px;
    border: 1px solid #ccc;
   }
   .output-field {
    margin-top: 20px;
   }
   .output-field p {
    margin-bottom: 10px;
   }
  </style>
</head>
<body>
  <div class="container">
   <h1>Stack and Queue Implementation</h1>
   <div class="input-field">
    <label for="stack-input">Stack Input:</label>
    <input type="text" id="stack-input" placeholder="Enter elements separated by
commas">
    <button id="stack-push-button">Push to Stack</button>
    <button id="stack-pop-button">Pop from Stack</button>
```

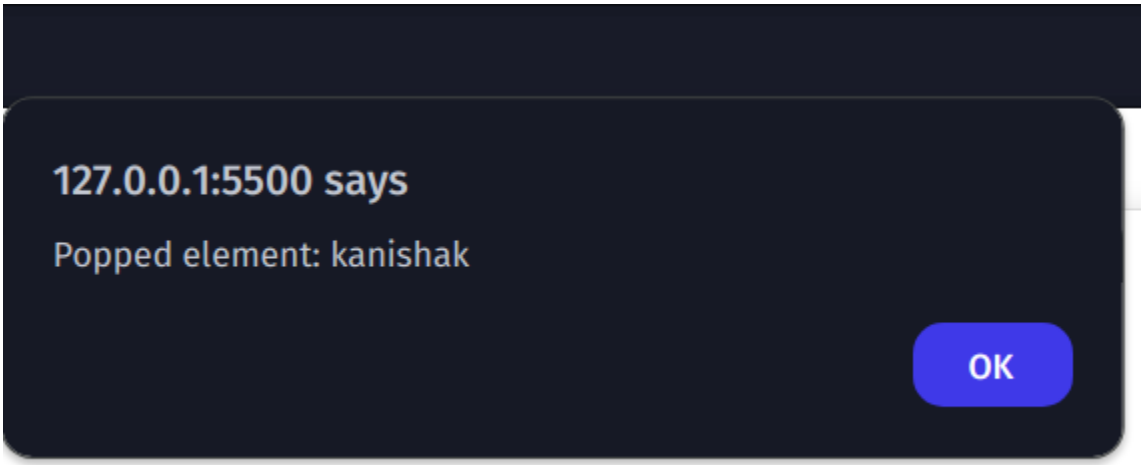
```
  </div>
  <div class="input-field">
   <label for="queue-input">Queue Input:</label>
   <input type="text" id="queue-input" placeholder="Enter elements separated by commas">
   <button id="queue-enqueue-button">Enqueue to Queue</button>
   <button id="queue-dequeue-button">Dequeue from Queue</button>
  </div>
  <div class="output-field">
   <p id="stack-output"></p>
   <p id="queue-output"></p>
  </div>
 </div>

 <script>
  // Stack implementation
  class Stack {
   constructor() {
    this.elements = [];
   }
   push(element) {
    this.elements.push(element);
   }
   pop() {
    return this.elements.pop();
   }
```

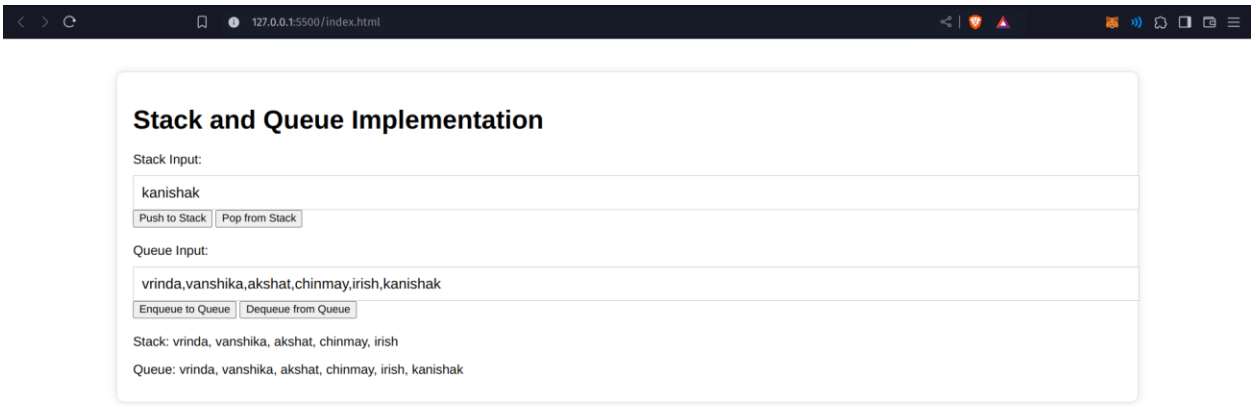
```
  peek() {
    return this.elements[this.elements.length - 1];
  }
  isEmpty() {
    return this.elements.length === 0;
  }
}

// Queue implementation
class Queue {
  constructor() {
    this.elements = [];
  }
  enqueue(element) {
    this.elements.push(element);
  }
  dequeue() {
    return this.elements.shift();
  }
  peek() {
    return this.elements[0];
  }
  isEmpty() {
    return this.elements.length === 0;
  }
}
```

```
  // Initialize stack and queue
  const stack = new Stack();
  const queue = new Queue();

  // Event listeners
  document.getElementById('stack-push-button').addEventListener('click', () => {
    const input = document.getElementById('stack-input').value.trim();
    const elements = input.split(',');
    elements.forEach(element => stack.push(element));
    document.getElementById('stack-output').textContent = `Stack: ${stack.elements.join(', ')}`;
  });

  document.getElementById('stack-pop-button').addEventListener('click', () => {
    if (!stack.isEmpty()) {
      const poppedElement = stack.pop();
      document.getElementById('stack-output').textContent = `Stack: ${stack.elements.join(', ')}`;
      alert(`Popped element: ${poppedElement}`);
    } else {
      alert('Stack is empty');
    }
  });

  document.getElementById('queue-enqueue-button').addEventListener('click', () => {
```

```
    const input = document.getElementById('queue-input').value.trim();
    const elements = input.split(',');
    elements.forEach(element => queue.enqueue(element));
    document.getElementById('queue-output').textContent = `Queue: ${queue.elements.join(', ')}`;
  });

  document.getElementById('queue-dequeue-button').addEventListener('click', () => {
    if (!queue.isEmpty()) {
      const dequeuedElement = queue.dequeue();
      document.getElementById('queue-output').textContent = `Queue: ${queue.elements.join(', ')}`;
      alert(`Dequeued element: ${dequeuedElement}`);
    } else {
      alert('Queue is empty');
    }
  });
 </script>
</body>
</html>
```


127.0.0.1:5500/index.html
Stack and Queue Implementation
Stack Input:
kanishak
Push to Stack
Pop from Stack
Queue Input:
vrinda,vanshika,akshat,chinmay,irish,kanishak
Enqueue to Queue
Dequeue from Queue
Stack: vrinda, vanshika, akshat, chinmay, irish, kanishak
Queue: vrinda, vanshika, akshat, chinmay, irish, kanishak


127.0.0.1:5500 says
Popped element: kanishak
OK



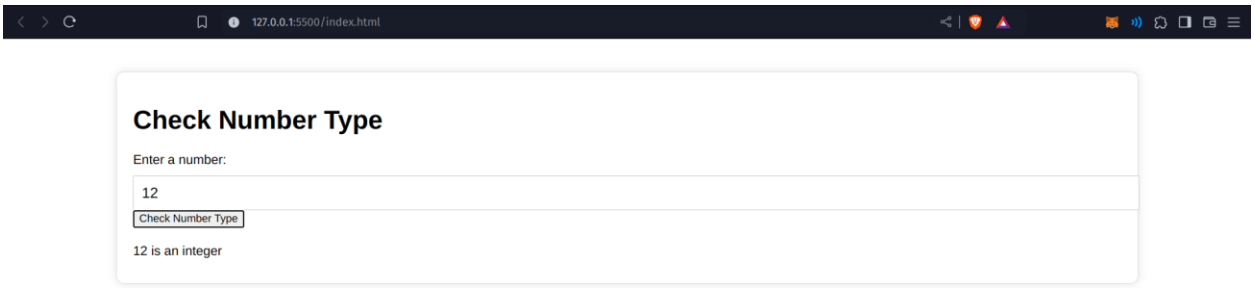
Q57)

```
 <!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Check Number Type</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
  .container {
   width: 80%;
   margin: 40px auto;
   padding: 20px;
   border: 1px solid #ddd;
```

```
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   }
   .input-field {
    margin-bottom: 20px;
   }
   .input-field label {
    display: block;
    margin-bottom: 10px;
   }
   .input-field input {
    width: 100%;
    padding: 10px;
    font-size: 18px;
    border: 1px solid #ccc;
   }
   .output-field {
    margin-top: 20px;
   }
   .output-field p {
    margin-bottom: 10px;
   }
  </style>
</head>
<body>
  <div class="container">
```

```
    <h1>Check Number Type</h1>
    <div class="input-field">
      <label for="number-input">Enter a number:</label>
      <input type="number" id="number-input" placeholder="Enter a number">
      <button id="check-button">Check Number Type</button>
    </div>
    <div class="output-field">
      <p id="output"></p>
    </div>
  </div>

  <script>
// Function to check if a number is a float or an integer
function checkNumberType(num) {
  if (Number.isInteger(num)) {
    return `${num} is an integer`;
  } else {
    return `${num} is a float`;
  }
}

// Test the function
console.log(checkNumberType(10)); // Output: 10 is an integer
console.log(checkNumberType(10.5)); // Output: 10.5 is a float
console.log(checkNumberType(20)); // Output: 20 is an integer
console.log(checkNumberType(30.25)); // Output: 30.25 is a float
```

```
  document.getElementById('check-button').addEventListener('click', () => {
    const num = parseFloat(document.getElementById('number-input').value);
    const result = checkNumberType(num);
    document.getElementById('output').textContent = result;
  });
 </script>
</body>
</html>
```



Q58)

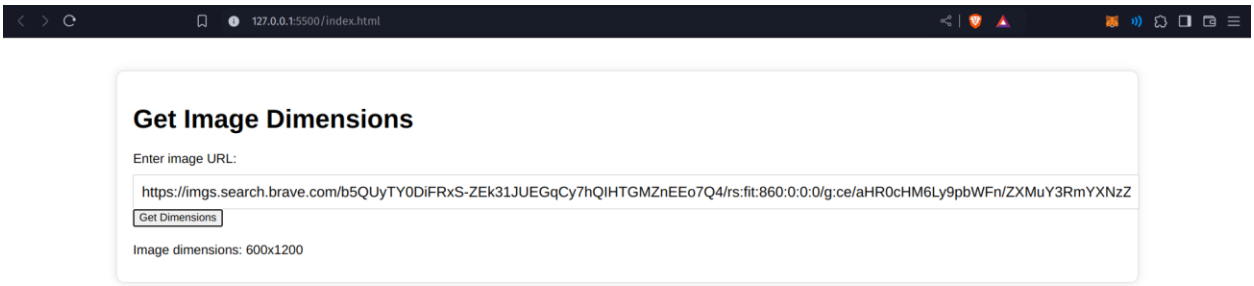
```
 <!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Get Image Dimensions</title>
```

```
<style>
  body {
    font-family: Arial, sans-serif;
  }
  .container {
    width: 80%;
    margin: 40px auto;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  .input-field {
    margin-bottom: 20px;
  }
  .input-field label {
    display: block;
    margin-bottom: 10px;
  }
  .input-field input {
    width: 100%;
    padding: 10px;
    font-size: 18px;
    border: 1px solid #ccc;
  }
  .output-field {
```

```
    margin-top: 20px;
   }
   .output-field p {
    margin-bottom: 10px;
   }
  </style>
</head>
<body>
  <div class="container">
   <h1>Get Image Dimensions</h1>
   <div class="input-field">
    <label for="image-url-input">Enter image URL:</label>
    <input type="text" id="image-url-input" placeholder="Enter image URL">
    <button id="get-dimensions-button">Get Dimensions</button>
   </div>
   <div class="output-field">
    <p id="output"></p>
   </div>
  </div>

  <script>
function getImageDimensions(imageUrl) {
  return new Promise((resolve, reject) => {
   const img = new Image();
   img.onload = function() {
    resolve({ width: img.width, height: img.height });
```

```
  };
  img.onerror = function() {
    reject(new Error(` Failed to load image: ${imageUrl}` ));
  };
  img.src = imageUrl;
 });
}

// Test the function
getImageDimensions('https://example.com/image.jpg')
 .then(dimensions => {
  console.log(` Image dimensions: ${dimensions.width}x${dimensions.height}` );
 })
 .catch(error => {
  console.error(error);
 });


  document.getElementById('get-dimensions-button').addEventListener('click', () => {
    const imageUrl = document.getElementById('image-url-input').value.trim();
    getImageDimensions(imageUrl)
      .then(dimensions => {
       document.getElementById('output').textContent = ` Image dimensions:
${dimensions.width}x${dimensions.height}` ;
      })
      .catch(error => {
       document.getElementById('output').textContent = ` Error: ${error.message}` ;
```

```
    });
  });
 </script>
</body>
</html>
```



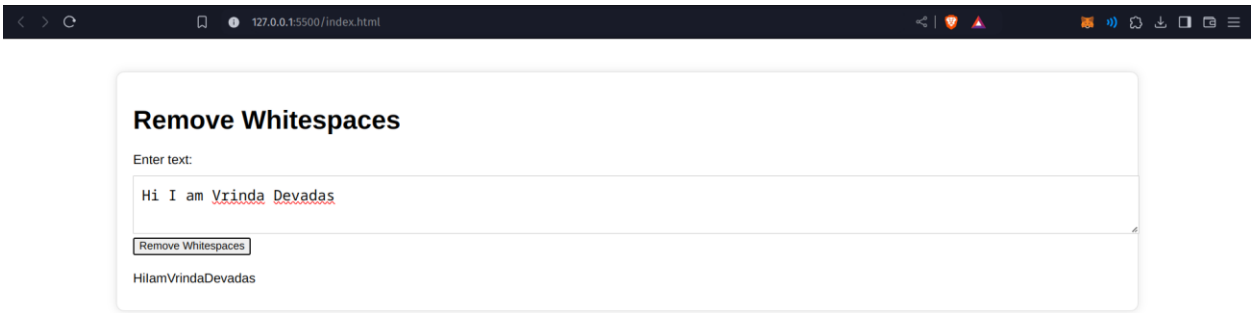
Q59)

```
 <!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Remove Whitespaces</title>
 <style>
  body {
   font-family: Arial, sans-serif;
  }
```

```
.container {
  width: 80%;
  margin: 40px auto;
  padding: 20px;
  border: 1px solid #ddd;
  border-radius: 10px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.input-field {
  margin-bottom: 20px;
}
.input-field label {
  display: block;
  margin-bottom: 10px;
}
.input-field textarea {
  width: 100%;
  padding: 10px;
  font-size: 18px;
  border: 1px solid #ccc;
}
.output-field {
  margin-top: 20px;
}
.output-field p {
  margin-bottom: 10px;
```

```
  }
 </style>
</head>
<body>
 <div class="container">
  <h1>Remove Whitespaces</h1>
  <div class="input-field">
   <label for="text-input">Enter text:</label>
   <textarea id="text-input" placeholder="Enter text"></textarea>
   <button id="remove-whitespaces-button">Remove Whitespaces</button>
  </div>
  <div class="output-field">
   <p id="output"></p>
  </div>
 </div>

 <script>
function removeWhitespaces(text) {
 return text.replace(/\s+/g, '');
}
 const originalText = "Hello   World!   This   is   a   test.";
const newText = removeWhitespaces(originalText);
console.log(newText); // Output: "HelloWorld!Thisisatest."

  document.getElementById('remove-whitespaces-button').addEventListener('click', () =>
{
```

```
    const originalText = document.getElementById('text-input').value.trim();
    const newText = removeWhitespaces(originalText);
    document.getElementById('output').textContent = newText;
  });
 </script>
</body>
</html>
```



Q60)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>Convert Date to Number</title>
```

```
<style>
  body {
    font-family: Arial, sans-serif;
  }
  .container {
    width: 80%;
    margin: 40px auto;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 10px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  }
  .input-field {
    margin-bottom: 20px;
  }
  .input-field label {
    display: block;
    margin-bottom: 10px;
  }
  .input-field input {
    width: 100%;
    padding: 10px;
    font-size: 18px;
    border: 1px solid #ccc;
  }
  .output-field {
```

```
    margin-top: 20px;
   }
   .output-field p {
    margin-bottom: 10px;
   }
  </style>
</head>
<body>
  <div class="container">
   <h1>Convert Date to Number</h1>
   <div class="input-field">
    <label for="date-input">Enter date (YYYY-MM-DDTHH:MM:SS.SSSZ):</label>
    <input type="text" id="date-input" placeholder="Enter date">
    <button id="convert-button">Convert to Number</button>
   </div>
   <div class="output-field">
    <p id="output"></p>
   </div>
  </div>

  <script>
function dateToNumber(date) {
  return date.getTime();
}
 const date = new Date('2022-07-25T14:30:00.000Z');
const number = dateToNumber(date);
```

```
console.log(number); // Output: 1658755800000



   document.getElementById('convert-button').addEventListener('click', () => {
     const dateInput = document.getElementById('date-input').value.trim();
     const date = new Date(dateInput);
     if (isNaN(date.getTime())) {
       document.getElementById('output').textContent = 'Invalid date format';
     } else {
       const number = dateToNumber(date);
       document.getElementById('output').textContent = ` Date converted to number: ${number}` ;
     }
   });
  </script>
</body>
</html>
```

Convert Date to Number

Enter date (YYYY-MM-DDTHH:MM:SS.SSSZ):

2024-02-24

Convert to Number

Date converted to number: 1708732800000