Assignment 03

SUBMITTED BY:

SAI KIRAN

12213149

IT-B(08)

NUMBER CONVERTER:

HTML, JS CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Decimal Converter</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>Decimal to Binary, Hexadecimal, and Octal Converter</h1>
        <input type="number" id="decimal" placeholder="Enter a decimal number">
        <button onclick="convertNumber()">Convert</button>

        <div class="output">
            <p>Binary: <span id="binary"></span></p>
            <p>Hexadecimal: <span id="hexa"></span></p>
            <p>Octal: <span id="octal"></span></p>
        </div>
    </div>
    <script>
    function convertNumber() {
    let decimal = document.getElementById('decimal').value;

    if (decimal === '') {
        alert('Please enter a vaild decimal number');
        return;
    }

    let binary = parseInt(decimal, 10).toString(2);
    let hex = parseInt(decimal, 10).toString(16).toUpperCase();
    let octal = parseInt(decimal, 10).toString(8);

    document.getElementById('binary').textContent = binary;
    document.getElementById('hexa').textContent = hex;
    document.getElementById('octal').textContent = octal;
}
```

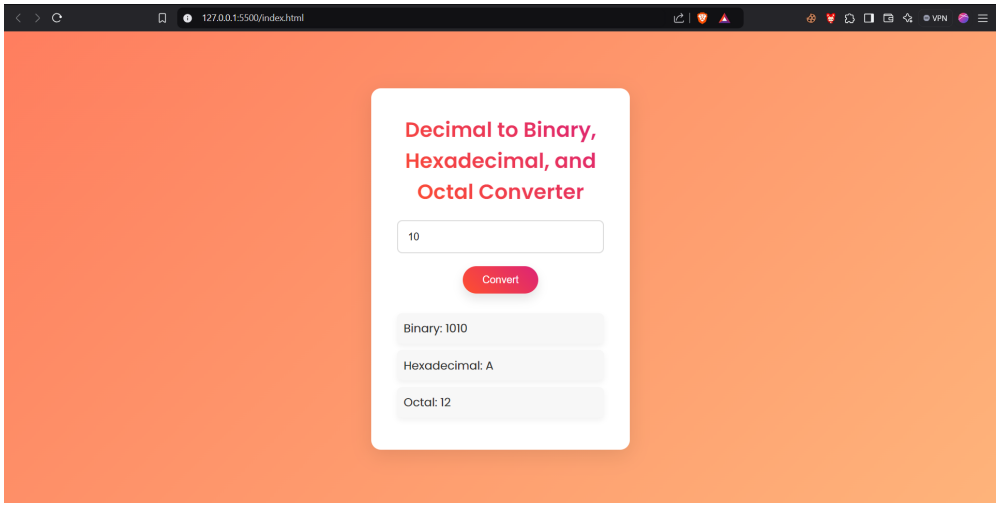
```
    </script>
</body>
</html>
```

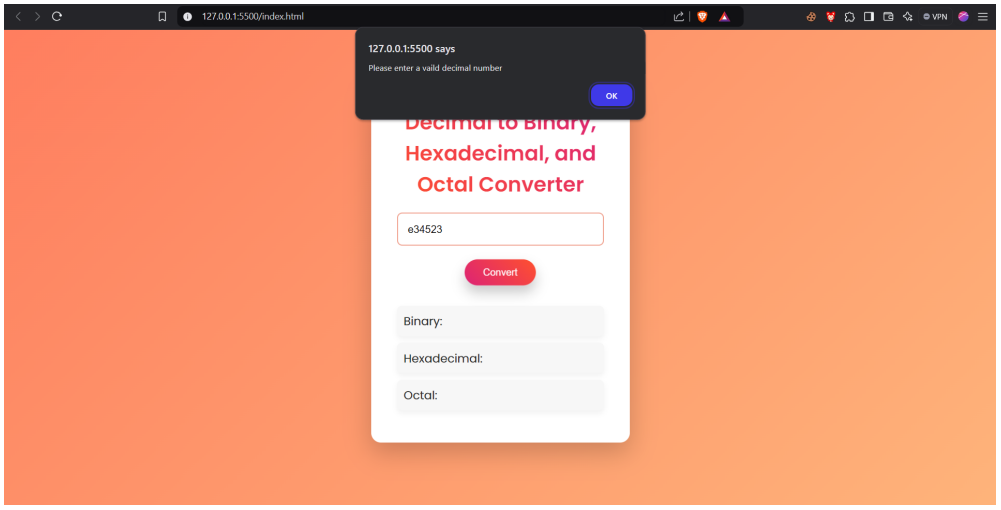
CSS CODE:

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: 'Poppins', sans-serif;
  background: linear-gradient(135deg, #ff7e5f, #feb47b);
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  color: #333;
}

.container {
  background: #fff;
  padding: 40px;
  border-radius: 15px;
  box-shadow: 0 8px 30px rgba(0, 0, 0, 0.1);
  width: 100%;
  max-width: 400px;
  text-align: center;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}
.container:hover {
  transform: translateY(-10px);
  box-shadow: 0 15px 40px rgba(0, 0, 0, 0.2);
}


h1 {
  font-size: 2rem;
  color: #333;
  margin-bottom: 20px;
  font-weight: 600;
  background: -webkit-linear-gradient(45deg, #ff512f, #dd2476);
  background-clip: text;
  -webkit-text-fill-color: transparent;
}

input {
  width: 100%;
  padding: 15px;
  font-size: 16px;
  border: 2px solid #ddd;
  border-radius: 8px;
  outline: none;
```

```
  transition: border-color 0.3s ease;
  margin-bottom: 20px;
}
input:focus {
  border-color: #ff7e5f;
}

button {
  background: linear-gradient(45deg, #ff512f, #dd2476);
  color: white;
  font-size: 16px;
  padding: 12px 30px;
  border: none;
  border-radius: 50px;
  cursor: pointer;
  transition: background 0.3s ease, transform 0.3s ease;
  box-shadow: 0 8px 15px rgba(0, 0, 0, 0.1);
}

button:hover {
  background: linear-gradient(45deg, #dd2476, #ff512f);
  transform: translateY(-3px);
  box-shadow: 0 12px 20px rgba(0, 0, 0, 0.2);
}

button:active {
  transform: scale(0.95);
}

.output {
  margin-top: 30px;
  font-size: 18px;
  text-align: left;
}

.output p {
  background: #f7f7f7;
  padding: 10px;
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.05);
  margin-bottom: 10px;
  transition: background 0.3s ease;
}

.output p:hover {
  background: #ffe0e0;
}

@media (max-width: 768px) {
  .container {
      padding: 30px;
      width: 90%;
  }
}
```

OUTPUT:





HTML VALIDATOR:

HTML, JS CODE:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>HTML Validator</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h1>HTML Validator</h1>
        <textarea id="html-input" placeholder="Enter your HTML code here..."></textarea>
        <button id="validate-btn">Validate</button>
        <div class="result" id="validation-result"></div>
```

```
    </div>
    <script>
    document.getElementById('validate-btn').addEventListener('click', validateHTML);

    function validateHTML() {
    const htmlInput = document.getElementById('html-input').value;
    const resultDiv = document.getElementById('validation-result');
    resultDiv.style.display = 'none';
    resultDiv.classList.remove('error', 'success');

    if (htmlInput.trim() === '') {
        resultDiv.innerHTML = 'Please enter some HTML to validate.';
        resultDiv.classList.add('error');
        resultDiv.style.display = 'block';
        return;
    }

    //creating a DOM tree
    let parser = new DOMParser();
    let doc = parser.parseFromString(htmlInput, 'text/html');
    let parserErrors = doc.querySelectorAll('parsererror');

    if (parserErrors.length > 0) {
        resultDiv.innerHTML = 'Invalid HTML syntax detected!';
        resultDiv.classList.add('error');
        resultDiv.style.display = 'block';
        return;
    }

    let unclosedTagResult = hasUnclosedTags(htmlInput);
    if (unclosedTagResult) {
        resultDiv.innerHTML = `Error: Unclosed tag found on line ${unclosedTagResult.lin
        resultDiv.classList.add('error');
        resultDiv.style.display = 'block';
    } else {
        resultDiv.innerHTML = 'No errors found in the HTML!';
        resultDiv.classList.add('success');
        resultDiv.style.display = 'block';
    }
}

    // to return line number
    function hasUnclosedTags(html) {
    const lines = html.split('\n');
    const tagPattern = /<\/?([a-z][a-z0-9]*)\b[^>]*>/gi;
    let tags = [];
    let match;

    for (let lineNumber = 0; lineNumber < lines.length; lineNumber++) {
        const currentLine = lines[lineNumber];
        while ((match = tagPattern.exec(currentLine)) !== null) {
            let tag = match[1];
            if (match[0].startsWith('</')) {
                if (tags.length === 0 || tags[tags.length - 1].tag !== tag) {
                    return { tag, line: lineNumber + 1 };
```

```
                }
                tags.pop();
            } else {
                tags.push({ tag, line: lineNumber + 1 });
            }
        }
    }

    if (tags.length !== 0) {
        return tags[0];
    }
    return null;
}


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

CSS CODE:

```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: 'Poppins', sans-serif;
  background: linear-gradient(135deg, #6a11cb, #2575fc);
  height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  color: #333;
  background-attachment: fixed;
}

.container {
  background: white;
  padding: 40px;
  border-radius: 20px;
  box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);
  width: 100%;
  max-width: 500px;
  text-align: center;
  transition: transform 0.3s ease, box-shadow 0.3s ease;
}

h1 {
  font-size: 2.5rem;
  color: #333;
  margin-bottom: 20px;
  font-weight: 700;
  background: linear-gradient(45deg, #ff512f, #dd2476);
```

```
  background-clip: text;
  -webkit-text-fill-color: transparent;
}

textarea {
  width: 100%;
  height: 150px;
  padding: 15px;
  font-size: 16px;
  border: 2px solid #ddd;
  border-radius: 10px;
  margin-bottom: 20px;
  outline: none;
  transition: border-color 0.3s ease, box-shadow 0.3s ease;
  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.1);
  resize: none;
}

textarea:focus {
  border-color: #6a11cb;
  box-shadow: 0 0 10px rgba(106, 17, 203, 0.5);
}

button {
  background: linear-gradient(45deg, #ff512f, #dd2476);
  color: white;
  font-size: 16px;
  padding: 12px 30px;
  border: none;
  border-radius: 50px;
  cursor: pointer;
  transition: background 0.3s ease, transform 0.3s ease;
  box-shadow: 0 8px 15px rgba(0, 0, 0, 0.1);
}

button:hover {
  background: linear-gradient(45deg, #dd2476, #ff512f);
  transform: translateY(-3px);
  box-shadow: 0 12px 20px rgba(0, 0, 0, 0.2);
}

button:active {
  transform: scale(0.95);
}

.result {
  margin-top: 20px;
  font-size: 18px;
  color: #fff;
  padding: 15px;
  background-color: #2ecc71;
  border-radius: 10px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
  display: none;
}
```

```css
.result.error {
  background-color: #e74c3c;
}

@media (max-width: 768px) {
  .container {
      padding: 20px;
      width: 90%;
  }

  h1 {
      font-size: 2rem;
  }

  textarea {
      height: 120px;
  }
}
```

OUTPUT: