Fraud Detection System for Online Transactions

code=
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width,
initial-scale=1.0">
   <title>Fraud Detection System</title>
   <style>
      /* Embedded CSS */
      body
      {
         font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
         background-color: #f0f2f5;
         margin: 0;
         padding: 0;
         color: #333;
      }
      .container
      {
         width: 90%;
         max-width: 600px;
         margin: 50px auto;
         padding: 30px;
         background-color: #fff;
         border-radius: 12px;
         box-shadow: 0 8px 16px rgba(0, 0, 0, 0.1);
         transition: transform 0.3s ease-in-out;
      }

      .container:hover
      {
         transform: translateY(-5px);
      }
```

```
h1
{
   text-align: center;
   color: #007bff;
   margin-bottom: 20px;
   font-size: 2rem;
   font-weight: 600;
}

form
{
   display: flex;
   flex-direction: column;
   gap: 20px;
}

label
{
   font-size: 18px;
   font-weight: 500;
   color: #555;
}

input, select
{
   padding: 12px;
   font-size: 16px;
   border: 1px solid #ccc;
   border-radius: 8px;
   box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.1);
   transition: border-color 0.3s ease-in-out;
}

input:focus, select:focus
{
   border-color: #007bff;
```

```
            outline: none;
        }

        button
        {
            padding: 12px;
            font-size: 16px;
            border: none;
            border-radius: 8px;
            background-color: #28a745;
            color: white;
            cursor: pointer;
            transition: background-color 0.3s ease-in-out, transform 0.3s
ease-in-out;
        }

        button:hover
        {
            background-color: #218838;
            transform: scale(1.02);
        }

        button:disabled
        {
            background-color: #6c757d;
            cursor: not-allowed;
        }

        #result
        {
            margin-top: 20px;
            text-align: center;
            display: none;
        }

        #loading
        {
            display: none;x
            margin-top: 20px;
            text-align: center;
        }

        #loading img
        {
            width: 60px;
            height: 60px;
            animation: spin 1s linear infinite;
        }

        @keyframes spin
        {
            0% { transform: rotate(0deg); }
            100% { transform: rotate(360deg); }
        }

        #resultText
        {
            font-size: 20px;
            font-weight: 700;
            color: #007bff;
        }

        .error
        {
            color: #dc3545;
            font-size: 14px;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Fraud Detection System</h1>
        <form id="transactionForm">
            <label for="transactionId">Transaction ID:</label>
            <input type="text" id="transactionId" name="transactionId"
required>
            <span id="idError" class="error"></span>

            <label for="amount">Transaction Amount:</label>
            <input type="number" id="amount" name="amount" required>
            <span id="amountError" class="error"></span>

            <label for="transactionType">Transaction Type:</label>
            <select id="transactionType" name="transactionType" required>
                <option value="purchase">Purchase</option>
                <option value="withdrawal">Withdrawal</option>
                <option value="transfer">Transfer</option>
            </select>

            <button type="submit">Check for Fraud</button>
        </form>

        <div id="loading">
            <p>Checking for fraud...</p>
            <img src="https://i.gifer.com/ZZ5H.gif" alt="Loading">
        </div>

        <div id="result">
            <p>Transaction Result: <span id="resultText">Awaiting
Input</span></p>
        </div>
    </div>

    <script>
        // Get form and elements
        const form = document.getElementById('transactionForm');
        const transactionIdInput =
document.getElementById('transactionId');
        const amountInput = document.getElementById('amount');
        const resultText = document.getElementById('resultText');
        const loading = document.getElementById('loading');
        const resultDiv = document.getElementById('result');
        const idError = document.getElementById('idError');
        const amountError = document.getElementById('amountError');

        // Validate form inputs
        function validateForm() {
            let valid = true;

            if (!transactionIdInput.value) {
                idError.textContent = 'Transaction ID is required.';
                valid = false;
            } else {
                idError.textContent = '';
            }

            if (!amountInput.value || amountInput.value <= 0) {
                amountError.textContent = 'Amount must be greater than 0.';
                valid = false;
            } else {
                amountError.textContent = '';
            }

            return valid;
        }

        // Simulate fraud detection logic with async function (to be replaced
by actual backend API call)
        async function checkForFraud(transactionId, amount,
transactionType) {
            return new Promise((resolve) => {
                setTimeout(() => {
                    if (amount > 1000 && transactionType === 'withdrawal') {
                        resolve('Suspicious Activity Detected!');
                    } else {
                        resolve('Transaction is Legitimate.');
                    }
                }, 2000);
            });
        }

        // Form submission handler
        form.addEventListener('submit', async function(event) {
            event.preventDefault();

            // Validate form
            if (!validateForm()) {
                return;
            }

            // Get input values
            const transactionId = transactionIdInput.value;
            const amount = parseFloat(amountInput.value);
            const transactionType =
document.getElementById('transactionType').value;

            // Display loading state
            resultDiv.style.display = 'none';
            loading.style.display = 'block';

            // Simulate fraud check
            const fraudResult = await checkForFraud(transactionId, amount,
transactionType);

            // Display result
            loading.style.display = 'none';
            resultDiv.style.display = 'block';
            resultText.textContent = fraudResult;
        });
    </script>
</body>
</html>
```