<!DOCTYPE html>

<html lang="id">

<head>

    <meta charset="UTF-8">

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

    <title>SIA APP</title>

    <style>

        body {

            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

            max-width: 900px;

            margin: 0 auto;

            padding: 20px;

            background-color: #f5f5f5;

        }

        .container {

            background-color: white;

            border-radius: 10px;

            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

            padding: 20px;

        }

        h1 {

            color: #2c3e50;

            text-align: center;

            margin-bottom: 30px;

        }

        h2 {

            color: #34495e;

            margin-top: 30px;

            border-bottom: 2px solid #eee;

            padding-bottom: 10px;

        }

        .form-group {

            margin-bottom: 15px;

        }

        label {

            display: block;

            margin-bottom: 5px;

            font-weight: bold;

        }

        input, select {

            width: 100%;

            padding: 10px;

            border: 1px solid #ddd;

            border-radius: 5px;

            box-sizing: border-box;

        }

        button {

            background-color: #3498db;

            color: white;

            border: none;

            padding: 10px 15px;

            border-radius: 5px;

            cursor: pointer;

            font-size: 16px;

            width: 100%;

            margin-top: 10px;

        }

        button:hover {

            background-color: #2980b9;

        }

        table {

            width: 100%;

            border-collapse: collapse;

            margin-top: 20px;

        }

        th, td {

            padding: 12px;

            text-align: left;

            border-bottom: 1px solid #ddd;

        }

        th {

            background-color: #f2f2f2;

            font-weight: bold;

        }

        tr:hover {

            background-color: #f5f5f5;

        }

        .delete-btn {

            background-color: #e74c3c;

            color: white;

            border: none;

            padding: 5px 10px;

            border-radius: 3px;

            cursor: pointer;

        }

        .delete-btn:hover {

            background-color: #c0392b;

        }

        .summary {

            margin-top: 30px;

            background-color: #f8f9fa;

            padding: 15px;

            border-radius: 5px;

            border-left: 5px solid #3498db;

        }

        .summary h3 {

            margin-top: 0;

            color: #2c3e50;

        }

        .export-btn {

            background-color: #27ae60;

            margin-top: 10px;

            margin-bottom: 10px;

        }

        .export-btn:hover {

            background-color: #219653;

        }

        .report-btn {

            background-color: #9b59b6;

            margin-top: 10px;

            margin-bottom: 10px;

        }

        .report-btn:hover {

            background-color: #8e44ad;

        }

        .balance {

            font-size: 18px;

            font-weight: bold;

            text-align: right;

        }

        .positive {

            color: #27ae60;

        }

        .negative {

            color: #e74c3c;

        }

        .credit {

            color: #e74c3c;

        }

        .debit {

            color: #27ae60;

        }

        .tab {

            overflow: hidden;

            border: 1px solid #ccc;

            background-color: #f1f1f1;

            border-radius: 5px 5px 0 0;

        }

        .tab button {

            background-color: inherit;

            float: left;

            border: none;

            outline: none;

            cursor: pointer;

            padding: 14px 16px;

            transition: 0.3s;

            font-size: 15px;

            margin-top: 0;

            width: auto;

            color: #333;

        }

        .tab button:hover {

            background-color: #ddd;

        }

        .tab button.active {

            background-color: #3498db;

            color: white;

        }

        .tabcontent {

            display: none;

            padding: 20px;

            border: 1px solid #ccc;

            border-top: none;

            border-radius: 0 0 5px 5px;

            animation: fadeEffect 1s;

        }

        @keyframes fadeEffect {

            from {opacity: 0;}

            to {opacity: 1;}

        }

        .category-select {

            margin-bottom: 15px;

        }

        .report-section {

            margin-bottom: 20px;

            padding: 15px;

            background-color: #fff;

            border-radius: 5px;

            box-shadow: 0 1px 3px rgba(0,0,0,0.1);

        }

        .report-title {

            border-bottom: 2px solid #3498db;

            padding-bottom: 10px;

            margin-bottom: 15px;

            color: #2c3e50;

        }

        #reports-container {

            margin-top: 20px;

        }

    </style>

</head>

<body>

    <div class="container">

        <h1>Aplikasi Laporan Keuangan</h1>

        <div class="tab">

            <button class="tablinks active" onclick="openTab(event, 'input-tab')">Input Transaksi</button>

            <button class="tablinks" onclick="openTab(event, 'reports-tab')">Laporan Keuangan</button>

        </div>

        <div id="input-tab" class="tabcontent" style="display: block;">

            <div class="form-group">

                <label for="date">Tanggal:</label>

                <input type="date" id="date" required>

            </div>

            <div class="form-group">

                <label for="description">Deskripsi:</label>

                <input type="text" id="description" placeholder="Masukkan deskripsi transaksi" required>

            </div>

            <div class="form-group">

                <label for="category">Kategori:</label>

                <select id="category" required>

                    <option value="pendapatan">Pendapatan</option>

                    <option value="beban">Beban</option>

                    <option value="aset">Aset</option>

                    <option value="kewajiban">Kewajiban</option>

                    <option value="modal">Modal</option>

                </select>

            </div>

            <div class="form-group">

                <label for="type">Jenis Transaksi:</label>

                <select id="type" required>

                    <option value="debit">Debit (Masuk)</option>

                    <option value="credit">Kredit (Keluar)</option>

                </select>

            </div>

            <div class="form-group">

                <label for="amount">Jumlah (Rp):</label>

                <input type="number" id="amount" placeholder="Masukkan jumlah transaksi" min="0" required>

            </div>

            <button id="add-transaction">Tambah Transaksi</button>

            <div id="transaction-list">

                <h2>Daftar Transaksi</h2>

                <table id="transactions-table">

                    <thead>

                        <tr>

                            <th>Tanggal</th>

                            <th>Deskripsi</th>

                            <th>Kategori</th>

                            <th>Jenis</th>

                            <th>Jumlah (Rp)</th>

                            <th>Aksi</th>

                        </tr>

                    </thead>

                    <tbody id="transactions-body">

                        <!-- Transaksi akan ditambahkan di sini -->

                    </tbody>

                </table>

            </div>

            <div class="summary">

                <h3>Ringkasan Keuangan</h3>

                <p>Total Debit: <span id="total-debit">Rp 0</span></p>

                <p>Total Kredit: <span id="total-credit">Rp 0</span></p>

                <p class="balance">Saldo: <span id="balance">Rp 0</span></p>

            </div>

            <button id="export-excel" class="export-btn">Ekspor Data ke Excel</button>

        </div>

        <div id="reports-tab" class="tabcontent">

            <div class="category-select">

                <label for="report-period">Periode Laporan:</label>

                <select id="report-period">

                    <option value="all">Semua Transaksi</option>

                    <option value="this-month">Bulan Ini</option>

                    <option value="last-month">Bulan Lalu</option>

                    <option value="this-year">Tahun Ini</option>

                </select>

            </div>

            <button id="generate-reports" class="report-btn">Generate Laporan</button>

            <div id="reports-container">

                <div class="report-section">

                    <h3 class="report-title">Laporan Laba Rugi</h3>

                    <div id="profit-loss-report"></div>

                </div>

                <div class="report-section">

                    <h3 class="report-title">Laporan Perubahan Modal</h3>

                    <div id="capital-change-report"></div>

                </div>

                <div class="report-section">

                    <h3 class="report-title">Laporan Arus Kas</h3>

                    <div id="cash-flow-report"></div>

                </div>

                <div class="report-section">

                    <h3 class="report-title">Neraca Keuangan</h3>

                    <div id="balance-sheet-report"></div>

                </div>

            </div>

            <button id="export-reports" class="export-btn">Ekspor Laporan ke Excel</button>

        </div>

    </div>

    <script>

        document.addEventListener('DOMContentLoaded', function() {

            // Set default date to today

            document.getElementById('date').valueAsDate = new Date();

            // Initialize transactions array

            let transactions = JSON.parse(localStorage.getItem('transactions')) || [];

            // Function to format currency

            function formatCurrency(amount) {

                return new Intl.NumberFormat('id-ID').format(amount);

            }

            // Function to format date for display

            function formatDate(dateString) {

                const date = new Date(dateString);

                return date.toLocaleDateString('id-ID', {

                    day: '2-digit',

                    month: '2-digit',

                    year: 'numeric'

                });

            }

            // Function to open tab

            window.openTab = function(evt, tabName) {

                const tabcontent = document.getElementsByClassName("tabcontent");

                for (let i = 0; i < tabcontent.length; i++) {

                    tabcontent[i].style.display = "none";

                }

                const tablinks = document.getElementsByClassName("tablinks");

                for (let i = 0; i < tablinks.length; i++) {

                    tablinks[i].className = tablinks[i].className.replace(" active", "");

                }

                document.getElementById(tabName).style.display = "block";

                evt.currentTarget.className += " active";

                if (tabName === 'reports-tab') {

                    generateReports();

                }

            }

            // Helper function to capitalize first letter

            function capitalizeFirstLetter(string) {

                return string.charAt(0).toUpperCase() + string.slice(1);

            }

            // Function to render transactions

            function renderTransactions() {

                const tbody = document.getElementById('transactions-body');

                tbody.innerHTML = '';

                transactions.forEach((transaction, index) => {

                    const row = document.createElement('tr');

                    const dateCell = document.createElement('td');

                    dateCell.textContent = formatDate(transaction.date);

                    row.appendChild(dateCell);

                    const descCell = document.createElement('td');

                    descCell.textContent = transaction.description;

                    row.appendChild(descCell);

                    const categoryCell = document.createElement('td');

                    categoryCell.textContent = transaction.category ? capitalizeFirstLetter(transaction.category) : 'Umum';

                    row.appendChild(categoryCell);

                    const typeCell = document.createElement('td');

                    typeCell.textContent = transaction.type === 'debit' ? 'Debit (Masuk)' : 'Kredit (Keluar)';

                    typeCell.className = transaction.type;

                    row.appendChild(typeCell);

                    const amountCell = document.createElement('td');

                    amountCell.textContent = 'Rp ' + formatCurrency(transaction.amount);

                    row.appendChild(amountCell);

                    const actionCell = document.createElement('td');

                    const deleteBtn = document.createElement('button');

                    deleteBtn.textContent = 'Hapus';

                    deleteBtn.className = 'delete-btn';

                    deleteBtn.onclick = function() {

                        deleteTransaction(index);

                    };

                    actionCell.appendChild(deleteBtn);

                    row.appendChild(actionCell);

                    tbody.appendChild(row);

                });

                updateSummary();

            }

            // Function to update summary

            function updateSummary() {

                let totalDebit = 0;

                let totalCredit = 0;

                transactions.forEach(transaction => {

                    if (transaction.type === 'debit') {

                        totalDebit += parseFloat(transaction.amount);

                    } else {

                        totalCredit += parseFloat(transaction.amount);

                    }

                });

                const balance = totalDebit - totalCredit;

                document.getElementById('total-debit').textContent = 'Rp ' + formatCurrency(totalDebit);

                document.getElementById('total-credit').textContent = 'Rp ' + formatCurrency(totalCredit);

                const balanceElement = document.getElementById('balance');

                balanceElement.textContent = 'Rp ' + formatCurrency(Math.abs(balance));

                balanceElement.className = balance >= 0 ? 'positive' : 'negative';

                if (balance < 0) {

                    balanceElement.textContent = '- ' + balanceElement.textContent;

                }

                // Save to localStorage

                localStorage.setItem('transactions', JSON.stringify(transactions));

            }

            // Function to add transaction

            function addTransaction() {

                const date = document.getElementById('date').value;

                const description = document.getElementById('description').value;

                const category = document.getElementById('category').value;

                const type = document.getElementById('type').value;

                const amount = document.getElementById('amount').value;

                if (!date || !description || !category || !type || !amount || isNaN(amount) || parseFloat(amount) <= 0) {

                    alert('Mohon lengkapi semua data transaksi dengan benar.');

                    return;

                }

                transactions.push({

                    date, description, category, type, amount: parseFloat(amount)

                });

                renderTransactions();

                document.getElementById('description').value = '';

                document.getElementById('amount').value = '';

            }

            // Function to delete transaction

            function deleteTransaction(index) {

                if (confirm('Yakin ingin menghapus transaksi ini?')) {

                    transactions.splice(index, 1);

                    renderTransactions();

                }

            }

            // Function to filter transactions by period

            function filterTransactionsByPeriod(period) {

                const now = new Date();

                if (period === 'all') return transactions;

                if (period === 'this-month') {

                    return transactions.filter(t => {

                        const d = new Date(t.date);

                        return d.getMonth() === now.getMonth() && d.getFullYear() === now.getFullYear();

                    });

                }

                if (period === 'last-month') {

                    let lastMonth = now.getMonth() - 1;

                    let year = now.getFullYear();

                    if (lastMonth < 0) {

                        lastMonth = 11;

                        year -= 1;

                    }

                    return transactions.filter(t => {

                        const d = new Date(t.date);

                        return d.getMonth() === lastMonth && d.getFullYear() === year;

                    });

                }

                if (period === 'this-year') {

                    return transactions.filter(t => {

                        const d = new Date(t.date);

                        return d.getFullYear() === now.getFullYear();

                    });

                }

                return transactions;

            }

            // Function to generate reports

            function generateReports() {

                const period = document.getElementById('report-period').value;

                const filtered = filterTransactionsByPeriod(period);

                // Laporan Laba Rugi

                let totalPendapatan = 0, totalBeban = 0;

                filtered.forEach(trx => {

                    if (trx.category === 'pendapatan' && trx.type === 'debit') totalPendapatan += parseFloat(trx.amount);

                    if (trx.category === 'beban' && trx.type === 'credit') totalBeban += parseFloat(trx.amount);

                });

                document.getElementById('profit-loss-report').innerHTML =

                    `<p>Total Pendapatan: <b>Rp ${formatCurrency(totalPendapatan)}</b></p>

                     <p>Total Beban: <b>Rp ${formatCurrency(totalBeban)}</b></p>

                     <p><b>Laba/Rugi Bersih: Rp ${formatCurrency(totalPendapatan - totalBeban)}</b></p>`;

                // Laporan Perubahan Modal (sederhana)

                let modalAkhir = 0;

                filtered.forEach(trx => {

                    if (trx.category === 'modal' && trx.type === 'debit') modalAkhir += parseFloat(trx.amount);

                    if (trx.category === 'modal' && trx.type === 'credit') modalAkhir -= parseFloat(trx.amount);

                });

                document.getElementById('capital-change-report').innerHTML =

                    `<p>Modal Akhir: <b>Rp ${formatCurrency(modalAkhir)}</b></p>`;

                // Laporan Arus Kas

                let kasMasuk = 0, kasKeluar = 0;

                filtered.forEach(trx => {

                    if (trx.type === 'debit') kasMasuk += parseFloat(trx.amount);

                    if (trx.type === 'credit') kasKeluar += parseFloat(trx.amount);

                });

                document.getElementById('cash-flow-report').innerHTML =

                    `<p>Kas Masuk: <b>Rp ${formatCurrency(kasMasuk)}</b></p>

                     <p>Kas Keluar: <b>Rp ${formatCurrency(kasKeluar)}</b></p>

                     <p><b>Saldo Kas: Rp ${formatCurrency(kasMasuk - kasKeluar)}</b></p>`;

                // Neraca Keuangan (sederhana)

                let aset = 0, kewajiban = 0, modal = 0;

                filtered.forEach(trx => {

                    if (trx.category === 'aset' && trx.type === 'debit') aset += parseFloat(trx.amount);

                    if (trx.category === 'aset' && trx.type === 'credit') aset -= parseFloat(trx.amount);

                    if (trx.category === 'kewajiban' && trx.type === 'credit') kewajiban += parseFloat(trx.amount);

                    if (trx.category === 'kewajiban' && trx.type === 'debit') kewajiban -= parseFloat(trx.amount);

                    if (trx.category === 'modal' && trx.type === 'debit') modal += parseFloat(trx.amount);

                    if (trx.category === 'modal' && trx.type === 'credit') modal -= parseFloat(trx.amount);

                });

                document.getElementById('balance-sheet-report').innerHTML =

                    `<p>Aset: <b>Rp ${formatCurrency(aset)}</b></p>

                     <p>Kewajiban: <b>Rp ${formatCurrency(kewajiban)}</b></p>

                     <p>Modal: <b>Rp ${formatCurrency(modal)}</b></p>

                     <p><b>Total Aset = Kewajiban + Modal: Rp ${formatCurrency(kewajiban + modal)}</b></p>`;

            }

            // Fungsi ekspor transaksi ke CSV (Excel)

            function exportTransactionsToCSV() {

                if (transactions.length === 0) {

                    alert("Tidak ada data transaksi untuk diekspor.");

                    return;

                }

                const header = ["Tanggal", "Deskripsi", "Kategori", "Jenis", "Jumlah (Rp)"];

                    let totalDebit = 0;

                    let totalCredit = 0;

            // Buat baris transaksi

                 const rows = transactions.map(trx => {

            // Hitung total debit dan kredit

                if (trx.type === "debit") {

                    totalDebit += parseFloat(trx.amount);

                } else {

                    totalCredit += parseFloat(trx.amount);

                }

                return [

                    formatDate(trx.date),

                    `"${trx.description.replace(/"/g, '""')}"`, // handle koma/tanda kutip

                    capitalizeFirstLetter(trx.category),

                    trx.type === "debit" ? "Debit (Masuk)" : "Kredit (Keluar)",

                    trx.amount

                ];

            });

            // Gabungkan header dan data transaksi

                    let csvContent = header.join(",") + "\n" + rows.map(e => e.join(",")).join("\n");

            // Tambahkan baris kosong

                    csvContent += "\n,,,,\n";

            // Tambahkan baris total

                    csvContent += `,,,Total Debit,${totalDebit}\n`;

                    csvContent += `,,,Total Kredit,${totalCredit}\n`;

                    csvContent += `,,,Saldo,${totalDebit - totalCredit}\n`;

            // Encode dan download

                    const encodedUri = "data:text/csv;charset=utf-8," + encodeURIComponent(csvContent);

                    const link = document.createElement("a");

                    link.setAttribute("href", encodedUri);

                    link.setAttribute("download", "data\_transaksi.csv");

                    document.body.appendChild(link);

                    link.click();

                    document.body.removeChild(link);

            };

            // Event listeners

            document.getElementById('add-transaction').onclick = addTransaction;

            document.getElementById('generate-reports').onclick = generateReports;

            document.getElementById('export-excel').onclick = exportTransactionsToCSV;

            document.getElementById('export-reports').onclick = exportReportsToCSV;

            function exportReportsToCSV() {

    const period = document.getElementById('report-period').value;

    const filtered = filterTransactionsByPeriod(period);

    // Ambil data laporan sesuai yang tampil di tab laporan

    const periodSelect = document.getElementById('report-period');

    const periodText = periodSelect.options[periodSelect.selectedIndex].text;

    const periodValue = periodSelect.value;

    constfiltered = filterTransactionsByPeriod(periodValue);

    // Laporan Laba Rugi

    let totalPendapatan = 0, totalBeban = 0;

    filtered.forEach(trx => {

        if (trx.category === 'pendapatan' && trx.type === 'debit') totalPendapatan += parseFloat(trx.amount);

        if (trx.category === 'beban' && trx.type === 'credit') totalBeban += parseFloat(trx.amount);

    });

    const labaRugi = totalPendapatan - totalBeban;

    // Laporan Perubahan Modal

    let modalMasuk = 0, modalKeluar = 0, modalAkhir = 0;

    filtered.forEach(trx => {

        if (trx.category === 'modal' && trx.type === 'debit') {

            modalMasuk += parseFloat(trx.amount);

            modalAkhir += parseFloat(trx.amount);

        }

        if (trx.category === 'modal' && trx.type === 'credit') {

            modalKeluar += parseFloat(trx.amount);

            modalAkhir -= parseFloat(trx.amount);

        }

    });

    // Laporan Arus Kas

    let kasMasuk = 0, kasKeluar = 0;

    filtered.forEach(trx => {

        if (trx.type === 'debit') kasMasuk += parseFloat(trx.amount);

        if (trx.type === 'credit') kasKeluar += parseFloat(trx.amount);

    });

    const saldoKas = kasMasuk - kasKeluar;

    // Neraca Keuangan (Aktiva & Pasiva)

    let aset = 0, kewajiban = 0, modal = 0;

    filtered.forEach(trx => {

        if (trx.category === 'aset' && trx.type === 'debit') aset += parseFloat(trx.amount);

        if (trx.category === 'aset' && trx.type === 'credit') aset -= parseFloat(trx.amount);

        if (trx.category === 'kewajiban' && trx.type === 'credit') kewajiban += parseFloat(trx.amount);

        if (trx.category === 'kewajiban' && trx.type === 'debit') kewajiban -= parseFloat(trx.amount);

        if (trx.category === 'modal' && trx.type === 'debit') modal += parseFloat(trx.amount);

        if (trx.category === 'modal' && trx.type === 'credit') modal -= parseFloat(trx.amount);

    });

    // Jika tidak ada data laporan, tampilkan pesan

    if (

        filtered.length === 0 ||

        (totalPendapatan === 0 && totalBeban === 0 && modalAkhir === 0 && kasMasuk === 0 && kasKeluar === 0 && aset === 0 && kewajiban === 0 && modal === 0)

    ) {

        alert("Tidak ada laporan keuangan untuk diekspor.");

        return;

    }

    // Siapkan data CSV dengan format rapi

    let csvContent = "data:text/csv;charset=utf-8,";

    csvContent += "LAPORAN KEUANGAN\n";

    csvContent += "Periode:," + periodText + "\n";

    csvContent += "Tanggal Ekspor:," + (new Date()).toLocaleString('id-ID') + "\n\n";

    // Laporan Laba Rugi

    csvContent += "Laporan Laba Rugi\n";

    csvContent += "Keterangan,Nominal (Rp)\n";

    csvContent += `Total Pendapatan,${totalPendapatan}\n`;

    csvContent += `Total Beban,${totalBeban}\n`;

    csvContent += `Laba/Rugi Bersih,${labaRugi}\n\n`;

    // Laporan Perubahan Modal

    csvContent += "Laporan Perubahan Modal\n";

    csvContent += "Keterangan,Nominal (Rp)\n";

    csvContent += `Modal Masuk,${modalMasuk}\n`;

    csvContent += `Modal Keluar,${modalKeluar}\n`;

    csvContent += `Modal Akhir,${modalAkhir}\n\n`;

    // Laporan Arus Kas

    csvContent += "Laporan Arus Kas\n";

    csvContent += "Keterangan,Nominal (Rp)\n";

    csvContent += `Kas Masuk,${kasMasuk}\n`;

    csvContent += `Kas Keluar,${kasKeluar}\n`;

    csvContent += `Saldo Kas,${saldoKas}\n\n`;

    // Neraca Keuangan (Aktiva & Pasiva)

    csvContent += "Neraca Keuangan\n";

    csvContent += "AKTIVA,,PASIVA,\n";

    csvContent += "Aset,Nominal,Kewajiban,Nominal\n";

    csvContent += `Aset,${aset},Kewajiban,${kewajiban}\n`;

    csvContent += `,,Modal,${modal}\n`;

    csvContent += `Total Aktiva,${aset},Total Pasiva,${kewajiban + modal}\n\n`;

    // Download file CSV

    const encodedUri = encodeURI(csvContent);

    const link = document.createElement("a");

    link.setAttribute("href", encodedUri);

    link.setAttribute("download", "laporan\_keuangan.csv");

    document.body.appendChild(link);

    link.click();

    document.body.removeChild(link);

}

            // Render initial data

            renderTransactions();

});

    </script>

</body>

</html>