**PERSONAL EXPANSES TRACKER AND ADVISORY SYSTEM**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Personal Expense Tracker</title>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<style>

/\* Color Variables \*/

:root {

--primary-color: #1d3557;

--secondary-color: #457b9d;

--accent-color: #2a9d8f;

--danger-color: #e63946;

--light-color: #f8f9fa;

--dark-color: #212529;

--text-color: #333;

--text-light: #6c757d;

--white: #ffffff;

--shadow: 0 4px 6px rgba(0, 0, 0, 0.1);

--shadow-hover: 0 6px 8px rgba(0, 0, 0, 0.15);

--transition: all 0.3s ease;

}

/\* Base Styles \*/

\* {

box-sizing: border-box;

margin: 0;

padding: 0;

}

body {

font-family: 'Poppins', sans-serif;

margin: 0;

padding: 0;

background: linear-gradient(rgba(255, 255, 255, 0.85), rgba(255, 255, 255, 0.85)),

url('https://images.unsplash.com/photo-1554224155-6726b3ff858f?ixlib=rb-4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D&auto=format&fit=crop&w=2000&q=80');

background-size: cover;

background-attachment: fixed;

background-position: center;

color: var(--text-color);

min-height: 100vh;

line-height: 1.6;

}

h1, h2, h3 {

font-weight: 600;

color: var(--primary-color);

margin-bottom: 1rem;

}

p {

color: var(--text-light);

margin-bottom: 1.5rem;

}

/\* Layout Components \*/

.navbar {

background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));

color: var(--white);

padding: 1rem 2rem;

position: fixed;

top: 0;

width: 100%;

z-index: 1000;

box-shadow: var(--shadow);

display: flex;

justify-content: center;

}

.navbar-inner {

display: flex;

max-width: 1200px;

width: 100%;

justify-content: space-between;

align-items: center;

}

.navbar-links {

display: flex;

}

.navbar a {

color: var(--white);

text-decoration: none;

padding: 0.75rem 1.5rem;

font-size: 1rem;

font-weight: 500;

border-radius: 4px;

transition: var(--transition);

margin: 0 0.5rem;

}

.navbar a:hover {

background-color: rgba(255, 255, 255, 0.15);

transform: translateY(-2px);

}

.container {

max-width: 900px;

margin: 100px auto;

padding: 2.5rem;

background-color: rgba(255, 255, 255, 0.95);

border-radius: 12px;

box-shadow: var(--shadow);

border: 1px solid rgba(0, 0, 0, 0.1);

backdrop-filter: blur(2px);

-webkit-backdrop-filter: blur(2px);

}

.container h1 {

text-align: center;

font-size: 2.25rem;

}

.container h2 {

font-size: 1.75rem;

margin-bottom: 1.5rem;

}

.container h3 {

font-size: 1.25rem;

margin-bottom: 1rem;

}

/\* Form Elements \*/

.form-group {

margin-bottom: 1.5rem;

}

label {

display: block;

margin-bottom: 0.5rem;

font-weight: 500;

color: var(--text-color);

}

input[type="text"],

input[type="password"],

input[type="number"],

input[type="date"],

select,

textarea {

width: 100%;

padding: 0.75rem 1rem;

border: 2px solid #e9ecef;

border-radius: 6px;

font-size: 1rem;

transition: var(--transition);

background-color: rgba(255, 255, 255, 0.9);

font-family: 'Poppins', sans-serif;

}

input[type="text"]:focus,

input[type="password"]:focus,

input[type="number"]:focus,

input[type="date"]:focus,

select:focus,

textarea:focus {

border-color: var(--primary-color);

outline: none;

background-color: var(--white);

box-shadow: 0 0 0 3px rgba(29, 53, 87, 0.2);

}

button, .btn {

padding: 0.75rem 1.5rem;

background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));

color: var(--white);

border: none;

font-size: 1rem;

font-weight: 600;

border-radius: 30px;

cursor: pointer;

transition: var(--transition);

box-shadow: var(--shadow);

display: inline-block;

text-align: center;

}

button:hover, .btn:hover {

background: linear-gradient(135deg, var(--secondary-color), var(--primary-color));

transform: translateY(-2px);

box-shadow: var(--shadow-hover);

}

.btn-accent {

background: linear-gradient(135deg, var(--accent-color), var(--primary-color));

}

.btn-accent:hover {

background: linear-gradient(135deg, var(--primary-color), var(--accent-color));

}

.btn-danger {

background: linear-gradient(135deg, var(--danger-color), #c1121f);

}

.btn-danger:hover {

background: linear-gradient(135deg, #c1121f, var(--danger-color));

}

/\* Table Styling \*/

table {

width: 100%;

margin: 1.5rem 0;

border-collapse: collapse;

background-color: var(--white);

border-radius: 8px;

overflow: hidden;

box-shadow: var(--shadow);

}

th, td {

padding: 1rem;

text-align: left;

border-bottom: 1px solid #e9ecef;

}

th {

background-color: var(--primary-color);

color: var(--white);

font-weight: 600;

}

tr:nth-child(even) {

background-color: rgba(240, 240, 240, 0.8);

}

tr:hover {

background-color: rgba(200, 230, 255, 0.6);

}

/\* Utility Classes \*/

.text-center {

text-align: center;

}

.mt-3 {

margin-top: 1rem;

}

.mb-3 {

margin-bottom: 1rem;

}

.hidden {

display: none;

}

/\* Specific Components \*/

.graph-container {

margin-top: 2rem;

width: 100%;

height: 300px;

background-color: var(--white);

border-radius: 10px;

padding: 1rem;

box-shadow: var(--shadow);

}

.overview {

display: grid;

grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));

gap: 1.5rem;

margin-bottom: 2rem;

}

.overview-card {

background-color: var(--white);

padding: 1.5rem;

border-radius: 10px;

text-align: center;

box-shadow: var(--shadow);

transition: var(--transition);

}

.overview-card:hover {

transform: translateY(-5px);

box-shadow: var(--shadow-hover);

}

.overview-card h3 {

font-size: 1rem;

color: var(--text-light);

margin-bottom: 0.5rem;

}

.overview-card p {

font-size: 1.5rem;

font-weight: 600;

color: var(--primary-color);

margin: 0;

}

/\* Progress bar for savings \*/

.progress-container {

width: 100%;

background-color: #e9ecef;

border-radius: 5px;

margin: 0.5rem 0;

overflow: hidden;

}

.progress-bar {

height: 20px;

background: linear-gradient(90deg, var(--accent-color), var(--primary-color));

border-radius: 5px;

text-align: center;

color: var(--white);

font-size: 0.75rem;

line-height: 20px;

transition: width 0.5s ease;

}

/\* Footer \*/

.footer {

background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));

color: var(--white);

text-align: center;

padding: 1rem;

position: fixed;

bottom: 0;

width: 100%;

box-shadow: 0 -4px 6px rgba(0, 0, 0, 0.1);

font-size: 0.875rem;

}

/\* Animation for page transitions \*/

@keyframes fadeIn {

from { opacity: 0; transform: translateY(10px); }

to { opacity: 1; transform: translateY(0); }

}

.container:not(.hidden) {

animation: fadeIn 0.5s ease-out;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.navbar {

padding: 0.75rem 1rem;

}

.navbar-inner {

flex-direction: column;

}

.navbar-links {

margin-top: 0.5rem;

flex-wrap: wrap;

justify-content: center;

}

.navbar a {

padding: 0.5rem 1rem;

margin: 0.25rem;

font-size: 0.875rem;

}

.container {

padding: 1.5rem;

margin: 80px 1rem;

}

.overview {

grid-template-columns: 1fr;

}

button, .btn {

width: 100%;

padding: 1rem;

}

}

</style>

</head>

<body>

<!-- Navbar -->

<div id="navbar" class="navbar">

<div class="navbar-inner">

<a href="javascript:void(0);" onclick="showPage('home')" class="logo">ExpenseTracker</a>

<div class="navbar-links">

<a href="javascript:void(0);" onclick="showPage('income')" class="logged-in">Set Income</a>

<a href="javascript:void(0);" onclick="showPage('budget')" class="logged-in">Budget</a>

<a href="javascript:void(0);" onclick="showPage('expense')" class="logged-in">Expenses</a>

<a href="javascript:void(0);" onclick="showPage('summary')" class="logged-in">Summary</a>

<a href="javascript:void(0);" onclick="showPage('savings')" class="logged-in">Savings</a>

<a href="javascript:void(0);" onclick="logout()" class="logged-in">Logout</a>

</div>

</div>

</div>

<!-- Home Page -->

<div id="home" class="container">

<h1>Welcome to ExpenseTracker</h1>

<p class="text-center">Take control of your finances with our intuitive expense tracking platform</p>

<div class="text-center" style="margin-top: 2rem;">

<button onclick="showPage('register')" class="btn">Register</button>

<button onclick="showPage('login')" class="btn" style="margin-left: 1rem;">Login</button>

</div>

</div>

<!-- Register Page -->

<div id="register" class="container hidden">

<h2>Create Your Account</h2>

<form id="registrationForm">

<div class="form-group">

<input type="text" id="username" placeholder="Username" required>

</div>

<div class="form-group">

<input type="password" id="password" placeholder="Password" required>

</div>

<button type="button" onclick="registerUser()" class="btn">Register</button>

</form>

<p class="text-center mt-3">Already have an account? <a href="javascript:void(0);" onclick="showPage('login')" style="color: var(--secondary-color); font-weight: 600;">Login here</a></p>

</div>

<!-- Login Page -->

<div id="login" class="container hidden">

<h2>Welcome Back</h2>

<form id="loginForm">

<div class="form-group">

<input type="text" id="loginUsername" placeholder="Username" required>

</div>

<div class="form-group">

<input type="password" id="loginPassword" placeholder="Password" required>

</div>

<button type="button" onclick="loginUser()" class="btn">Login</button>

</form>

<p class="text-center mt-3">Don't have an account? <a href="javascript:void(0);" onclick="showPage('register')" style="color: var(--secondary-color); font-weight: 600;">Register here</a></p>

</div>

<!-- Income Page -->

<div id="income" class="container hidden">

<h2>Set Your Monthly Income</h2>

<form id="incomeForm">

<div class="form-group">

<input type="text" id="incomeSource" placeholder="Income Source (e.g., Salary)" required>

</div>

<div class="form-group">

<input type="number" id="incomeAmount" placeholder="Amount (INR)" required>

</div>

<button type="button" onclick="addIncomeSource()" class="btn">Add Income Source</button>

</form>

<h3>Income Sources</h3>

<table id="incomeTable">

<thead>

<tr>

<th>Source</th>

<th>Amount (INR)</th>

<th>Action</th>

</tr>

</thead>

<tbody></tbody>

</table>

</div>

<!-- Budget Page -->

<div id="budget" class="container hidden">

<h2>Set Your Monthly Budget</h2>

<p>Set spending limits for different categories to stay on track with your finances.</p>

<form id="budgetForm">

<div class="form-group">

<input type="number" id="foodBudget" placeholder="Food Budget (INR)" required>

</div>

<div class="form-group">

<input type="number" id="transportBudget" placeholder="Transport Budget (INR)" required>

</div>

<div class="form-group">

<input type="number" id="utilitiesBudget" placeholder="Utilities Budget (INR)" required>

</div>

<div class="form-group">

<input type="number" id="entertainmentBudget" placeholder="Entertainment Budget (INR)" required>

</div>

<div class="form-group">

<input type="number" id="otherBudget" placeholder="Other Budget (INR)" required>

</div>

<button type="button" onclick="setCategoryBudgets()" class="btn">Set Budgets</button>

</form>

</div>

<!-- Expense Page -->

<div id="expense" class="container hidden">

<h2>Track Your Expenses</h2>

<form id="expenseForm">

<div class="form-group">

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

</div>

<div class="form-group">

<input type="text" id="expenseName" placeholder="Expense Name" required>

</div>

<div class="form-group">

<input type="number" id="expenseAmount" placeholder="Amount (INR)" required>

</div>

<div class="form-group">

<select id="expenseCategory" required>

<option value="">Select Category</option>

<option value="Food">Food</option>

<option value="Transport">Transport</option>

<option value="Utilities">Utilities</option>

<option value="Entertainment">Entertainment</option>

<option value="Other">Other</option>

</select>

</div>

<div class="form-group">

<input type="text" id="expenseNote" placeholder="Note (Optional)">

</div>

<div class="form-group">

<input type="text" id="expenseTag" placeholder="Tag (e.g., Urgent, Recurring)">

</div>

<button type="button" onclick="addExpense()" class="btn">Add Expense</button>

</form>

<h3>Your Expenses</h3>

<div class="form-group">

<label for="filterCategory">Filter by Category:</label>

<select id="filterCategory" onchange="filterExpenses()">

<option value="">All Categories</option>

<option value="Food">Food</option>

<option value="Transport">Transport</option>

<option value="Utilities">Utilities</option>

<option value="Entertainment">Entertainment</option>

<option value="Other">Other</option>

</select>

</div>

<table id="expenseTable">

<thead>

<tr>

<th>Date</th>

<th>Expense Name</th>

<th>Amount (INR)</th>

<th>Category</th>

<th>Note</th>

<th>Tag</th>

</tr>

</thead>

<tbody></tbody>

</table>

<button onclick="exportExpenses()" class="btn btn-accent mt-3">Export Expenses as CSV</button>

</div>

<!-- Summary Page -->

<div id="summary" class="container hidden">

<h2>Financial Summary</h2>

<div class="overview">

<div class="overview-card">

<h3>Total Income</h3>

<p id="totalIncome">₹0.00</p>

</div>

<div class="overview-card">

<h3>Total Expenses</h3>

<p id="totalExpenses">₹0.00</p>

</div>

<div class="overview-card">

<h3>Remaining Balance</h3>

<p id="remainingBalance">₹0.00</p>

</div>

<div class="overview-card">

<h3>Spent This Month</h3>

<p id="spentThisMonth">₹0.00</p>

</div>

</div>

<div class="graph-container">

<canvas id="expenseChart"></canvas>

</div>

</div>

<!-- Savings Goals Page -->

<div id="savings" class="container hidden">

<h2>Savings Goals</h2>

<p>Set financial goals and track your progress toward achieving them.</p>

<form id="savingsForm">

<div class="form-group">

<input type="text" id="goalName" placeholder="Goal Name" required>

</div>

<div class="form-group">

<input type="number" id="goalAmount" placeholder="Target Amount (INR)" required>

</div>

<div class="form-group">

<select id="goalCategory" required>

<option value="">Select Category</option>

<option value="Emergency Fund">Emergency Fund</option>

<option value="Vacation">Vacation</option>

<option value="New Car">New Car</option>

<option value="Home Purchase">Home Purchase</option>

<option value="Other">Other</option>

</select>

</div>

<button type="button" onclick="addSavingsGoal()" class="btn btn-accent">Add Goal</button>

</form>

<h3>Your Savings Goals</h3>

<table id="savingsTable">

<thead>

<tr>

<th>Goal Name</th>

<th>Target Amount</th>

<th>Category</th>

<th>Progress</th>

<th>Add Money</th>

</tr>

</thead>

<tbody></tbody>

</table>

</div>

<!-- Footer -->

<div class="footer">

<p>&copy; 2025 ExpenseTracker. All rights reserved.</p>

</div>

<script>

// JavaScript to manage pages, user authentication, expenses, etc.

let expenseChart;

function showPage(page) {

const pages = document.querySelectorAll('.container');

pages.forEach(p => p.classList.add('hidden'));

document.getElementById(page).classList.remove('hidden');

if (page === 'summary') {

updateSummary();

renderExpenseChart();

} else if (page === 'income') {

updateIncomeTable();

} else if (page === 'savings') {

updateSavingsTable();

} else if (page === 'expense') {

updateExpenseTable();

}

}

function registerUser() {

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

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

if (username && password) {

localStorage.setItem('user', JSON.stringify({ username, password }));

alert("User registered successfully!");

showPage('login');

} else {

alert("Please fill in all fields.");

}

}

function loginUser() {

const username = document.getElementById('loginUsername').value;

const password = document.getElementById('loginPassword').value;

const user = JSON.parse(localStorage.getItem('user'));

if (user && user.username === username && user.password === password) {

// Clear previous user data

localStorage.removeItem('incomeSources');

localStorage.removeItem('categoryBudgets');

localStorage.removeItem('expenses');

localStorage.removeItem('savingsGoals');

// Set login status

localStorage.setItem('isLoggedIn', true);

alert("Login successful!");

showPage('income');

updateNavbar();

} else {

alert("Invalid credentials.");

}

}

function addIncomeSource() {

const source = document.getElementById('incomeSource').value;

const amount = parseFloat(document.getElementById('incomeAmount').value) || 0;

if (!source || isNaN(amount) || amount <= 0) {

alert("Please fill in all fields correctly.");

return;

}

const incomeSources = JSON.parse(localStorage.getItem('incomeSources')) || [];

incomeSources.push({ source, amount });

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

updateIncomeTable();

// Clear form

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

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

}

function updateIncomeTable() {

const incomeSources = JSON.parse(localStorage.getItem('incomeSources')) || [];

const tableBody = document.getElementById('incomeTable').getElementsByTagName('tbody')[0];

tableBody.innerHTML = '';

incomeSources.forEach((income, index) => {

const row = tableBody.insertRow();

row.innerHTML = `

<td>${income.source}</td>

<td>₹${income.amount.toFixed(2)}</td>

<td><button onclick="deleteIncomeSource(${index})" class="btn btn-danger" style="padding: 0.5rem 1rem;">Delete</button></td>

`;

});

}

function deleteIncomeSource(index) {

const incomeSources = JSON.parse(localStorage.getItem('incomeSources')) || [];

incomeSources.splice(index, 1);

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

updateIncomeTable();

}

function setCategoryBudgets() {

const budgets = {

Food: parseFloat(document.getElementById('foodBudget').value) || 0,

Transport: parseFloat(document.getElementById('transportBudget').value) || 0,

Utilities: parseFloat(document.getElementById('utilitiesBudget').value) || 0,

Entertainment: parseFloat(document.getElementById('entertainmentBudget').value) || 0,

Other: parseFloat(document.getElementById('otherBudget').value) || 0,

};

localStorage.setItem('categoryBudgets', JSON.stringify(budgets));

alert("Budgets set successfully!");

showPage('expense');

}

function addExpense() {

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

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

const expenseAmount = parseFloat(document.getElementById('expenseAmount').value) || 0;

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

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

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

if (!date || !expenseName || isNaN(expenseAmount) || expenseAmount <= 0 || !expenseCategory) {

alert("Please fill in all required fields correctly.");

return;

}

const expenses = JSON.parse(localStorage.getItem('expenses')) || [];

expenses.push({ date, expenseName, expenseAmount, expenseCategory, expenseNote, expenseTag });

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

updateExpenseTable();

checkBudget(expenseCategory, expenseAmount);

// Clear form

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

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

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

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

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

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

showPage('summary');

}

function updateExpenseTable() {

const expenses = JSON.parse(localStorage.getItem('expenses')) || [];

const tableBody = document.getElementById('expenseTable').getElementsByTagName('tbody')[0];

tableBody.innerHTML = '';

expenses.forEach(expense => {

const row = tableBody.insertRow();

row.innerHTML = `

<td>${expense.date}</td>

<td>${expense.expenseName}</td>

<td>₹${expense.expenseAmount.toFixed(2)}</td>

<td>${expense.expenseCategory}</td>

<td>${expense.expenseNote || '-'}</td>

<td>${expense.expenseTag || '-'}</td>

`;

});

}

function filterExpenses() {

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

const expenses = JSON.parse(localStorage.getItem('expenses')) || [];

const filteredExpenses = category ? expenses.filter(expense => expense.expenseCategory === category) : expenses;

const tableBody = document.getElementById('expenseTable').getElementsByTagName('tbody')[0];

tableBody.innerHTML = '';

filteredExpenses.forEach(expense => {

const row = tableBody.insertRow();

row.innerHTML = `

<td>${expense.date}</td>

<td>${expense.expenseName}</td>

<td>₹${expense.expenseAmount.toFixed(2)}</td>

<td>${expense.expenseCategory}</td>

<td>${expense.expenseNote || '-'}</td>

<td>${expense.expenseTag || '-'}</td>

`;

});

}

function exportExpenses() {

const expenses = JSON.parse(localStorage.getItem('expenses')) || [];

if (expenses.length === 0) {

alert("No expenses to export!");

return;

}

// Create CSV header

const headers = Object.keys(expenses[0]);

let csvContent = "data:text/csv;charset=utf-8," + headers.join(",") + "\n";

// Add expense data

csvContent += expenses.map(expense =>

headers.map(header => expense[header]).join(",")

).join("\n");

const encodedUri = encodeURI(csvContent);

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

link.setAttribute("href", encodedUri);

link.setAttribute("download", "expenses.csv");

document.body.appendChild(link);

link.click();

document.body.removeChild(link);

}

function addSavingsGoal() {

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

const goalAmount = parseFloat(document.getElementById('goalAmount').value) || 0;

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

if (!goalName || isNaN(goalAmount) || goalAmount <= 0 || !goalCategory) {

alert("Please fill in all fields correctly.");

return;

}

const savingsGoals = JSON.parse(localStorage.getItem('savingsGoals')) || [];

savingsGoals.push({

goalName,

goalAmount,

savedAmount: 0,

category: goalCategory,

createdAt: new Date().toISOString()

});

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

updateSavingsTable();

// Clear form

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

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

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

}

function updateSavingsTable() {

const savingsGoals = JSON.parse(localStorage.getItem('savingsGoals')) || [];

const tableBody = document.getElementById('savingsTable').getElementsByTagName('tbody')[0];

tableBody.innerHTML = '';

savingsGoals.forEach((goal, index) => {

const progress = (goal.savedAmount / goal.goalAmount) \* 100;

const progressBar = progress > 0 ?

`<div class="progress-container">

<div class="progress-bar" style="width: ${Math.min(progress, 100)}%">

${progress.toFixed(1)}%

</div>

</div>` : 'Not started';

const row = tableBody.insertRow();

row.innerHTML = `

<td>${goal.goalName}</td>

<td>₹${goal.goalAmount.toFixed(2)}</td>

<td>${goal.category}</td>

<td>

₹${goal.savedAmount.toFixed(2)} of ₹${goal.goalAmount.toFixed(2)}

${progressBar}

</td>

<td>

<input type="number" id="addAmount${index}" placeholder="Amount" min="0" style="width: 80px; padding: 8px; margin-right: 5px;" />

<button onclick="addMoneyToGoal(${index})" class="btn btn-accent" style="padding: 8px 15px;">Add</button>

</td>

`;

});

}

function addMoneyToGoal(index) {

const amountInput = document.getElementById(`addAmount${index}`);

const amountToAdd = parseFloat(amountInput.value) || 0;

if (amountToAdd <= 0) {

alert("Please enter a valid amount to add.");

return;

}

const savingsGoals = JSON.parse(localStorage.getItem('savingsGoals')) || [];

savingsGoals[index].savedAmount += amountToAdd;

// Check if goal is reached

if (savingsGoals[index].savedAmount >= savingsGoals[index].goalAmount) {

alert(`Congratulations! You've reached your "${savingsGoals[index].goalName}" goal!`);

}

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

updateSavingsTable();

amountInput.value = '';

}

function updateSummary() {

const incomeSources = JSON.parse(localStorage.getItem('incomeSources')) || [];

const totalIncome = incomeSources.reduce((sum, income) => sum + income.amount, 0);

const expenses = JSON.parse(localStorage.getItem('expenses')) || [];

const totalExpenses = expenses.reduce((sum, expense) => sum + expense.expenseAmount, 0);

const remainingBalance = totalIncome - totalExpenses;

document.getElementById('totalIncome').textContent = `₹${totalIncome.toFixed(2)}`;

document.getElementById('totalExpenses').textContent = `₹${totalExpenses.toFixed(2)}`;

document.getElementById('remainingBalance').textContent = `₹${remainingBalance.toFixed(2)}`;

document.getElementById('spentThisMonth').textContent = `₹${totalExpenses.toFixed(2)}`;

}

function checkBudget(category, amount) {

const budgets = JSON.parse(localStorage.getItem('categoryBudgets'));

if (!budgets) return;

const expenses = JSON.parse(localStorage.getItem('expenses')) || [];

const categoryExpenses = expenses.filter(expense => expense.expenseCategory === category)

.reduce((sum, expense) => sum + expense.expenseAmount, 0);

if (categoryExpenses > budgets[category]) {

alert(`Budget exceeded for ${category}! Consider reducing expenses in this category.`);

} else if (categoryExpenses + amount > budgets[category] \* 0.8) {

alert(`Warning: You're approaching your ${category} budget limit (${((categoryExpenses + amount) / budgets[category] \* 100).toFixed(0)}%).`);

}

}

function renderExpenseChart() {

const expenses = JSON.parse(localStorage.getItem('expenses')) || [];

const categories = ['Food', 'Transport', 'Utilities', 'Entertainment', 'Other'];

const data = categories.map(category => {

return expenses.filter(expense => expense.expenseCategory === category)

.reduce((sum, expense) => sum + expense.expenseAmount, 0);

});

const budgets = JSON.parse(localStorage.getItem('categoryBudgets')) || {};

const budgetData = categories.map(category => budgets[category] || 0);

const ctx = document.getElementById('expenseChart').getContext('2d');

if (expenseChart) {

expenseChart.destroy();

}

expenseChart = new Chart(ctx, {

type: 'bar',

data: {

labels: categories,

datasets: [

{

label: 'Actual Spending',

data: data,

backgroundColor: '#1d3557',

borderColor: '#1d3557',

borderWidth: 1

},

{

label: 'Budget',

data: budgetData,

type: 'line',

borderColor: '#2a9d8f',

borderWidth: 2,

fill: false,

pointBackgroundColor: '#2a9d8f'

}

]

},

options: {

responsive: true,

scales: {

y: {

beginAtZero: true,

title: {

display: true,

text: 'Amount (₹)'

}

}

},

plugins: {

title: {

display: true,

text: 'Spending vs Budget by Category',

font: {

size: 16

}

},

legend: {

position: 'top'

}

}

}

});

}

function logout() {

localStorage.removeItem('incomeSources');

localStorage.removeItem('categoryBudgets');

localStorage.removeItem('expenses');

localStorage.removeItem('savingsGoals');

localStorage.removeItem('isLoggedIn');

alert("Logged out successfully!");

showPage('home');

updateNavbar();

}

function updateNavbar() {

const isLoggedIn = localStorage.getItem('isLoggedIn');

const loggedInLinks = document.querySelectorAll('.logged-in');

loggedInLinks.forEach(link => {

link.style.display = isLoggedIn ? 'inline-block' : 'none';

});

}

// Initialize

if (localStorage.getItem('isLoggedIn')) {

showPage('income');

updateNavbar();

} else {

showPage('home');

updateNavbar();

}

</script>

</body>

</html>