Emergency Fund Calculator

Description

A

ALI ASGAR

FINANCIAL SAFETY ENGINE

EMERGENCY FUND PLANNER

How Financially Ready is Your Family?

Find your emergency-fund target, understand why you need it, and build a practical monthly action plan.

About 60â??90 seconds

10+ Years in Financial Advisory â?¢ Educational Planning Tool â?¢ No Obligation

1

2

3

4

STEP 01

Your Family's Monthly Financial Needs

Enter the essential monthly expenses your family would still need to pay if your income suddenly stopped.

Monthly take-home income

Essential family expenses

Home/rent, food, electricity, school, medicines, travel, insurance and other must-pay expenses. Do not include EMI here.

Monthly EMI / debt payments

Emergency money available today

Include money that is quickly accessible: savings account, emergency FD or similar liquid reserves. Do not count property or long-term investments.

> **Simple idea:** Your emergency fund helps your family continue paying important bills during an unexpected financial problem.

Continue â??

`).join(''); if (Number(r.gap) <= 0) { q('.simpleExplanation').textContent = 'Your family needs about ' + money(r.outflow) + ' each month for essential commitments. Based on the information you entered, the calculator recommends ' + r.recommended_months + ' months of safety â?? ' + money(r.recommended_fund) + '. You already have ' + money(r.savings) + ' in liquid emergency savings, so no additional emergency-fund gap is currently identified. Keep the required reserve liquid and accessible, and review it when your circumstances materially change.'; } else { q('.simpleExplanation').textContent = 'Your family needs about ' + money(r.outflow) + ' each month for essential commitments. Based on your answers, a ' + r.recommended_months + '-month reserve is recommended, giving you a target of ' + money(r.recommended_fund) + '. You already have ' + money(r.savings) + ', so ' + money(r.gap) + ' remains to reach the recommended level.'; } q('.ladderGrid').innerHTML = r.ladder.map(item => ` ${escapeHtml(item.label)} **${money(item.amount)}** ${item.months} ${item.months === 1 ? 'month' : 'months'} protected
`).join(''); renderCheck('.healthCheck', 'Health Insurance', r.protection.health); renderCheck('.lifeCheck', 'Life Protection', r.protection.life); renderCheck('.debtCheck', 'High-interest Debt', r.protection.high_debt); setupTargetOptions(); } function renderCheck(selector, heading, check) { const el = q(selector); el.className = selector.replace('.', '') + ' checkCard ' + check.status; el.innerHTML = **'' + escapeHtml(heading) + ''** + escapeHtml(check.text); } safeBind('.editAnswers', 'click', function(){ showStep(1); }); safeBind('.openPlanner', 'click', function(){ setupTargetOptions(); showStep(4); }); safeBind('.backResult', 'click', function(){ showStep(3); }); function setupTargetOptions() { if (!calculation) return; const rec = Number(calculation.recommended_months); const standardMonths = [3, 6, 9, 12]; const months = [...new Set(standardMonths.concat(rec))].sort((a, b) => a - b); const container = q('.targetOptions'); container.innerHTML = ''; months.forEach(month => { const button = document.createElement('button'); button.type = 'button'; button.dataset.months = month; if (month === rec) button.classList.add('recommendedTarget'); button.innerHTML = (month === rec ? '*RECOMMENDED*' : '') + **'' + month + '**months' + '' + money(calculation.outflow * month) + ''; if (month === rec) button.classList.add('selected'); container.appendChild(button); }); selectedSafetyMonths = rec; } const targetOptions = q('.targetOptions'); if (targetOptions) { targetOptions.addEventListener('click', function(e){ const button = e.target.closest('button[data-months]'); if (!button || !this.contains(button)) return;

```
selectedSafetyMonths = Number(button.dataset.months); qa('.targetOptions button').forEach(b =>
b.classList.remove('selected')); button.classList.add('selected'); }); }
qa('input[name="plan_mode"]').forEach(radio => { radio.addEventListener('change', function(){ const mode
= value('plan_mode'); q('.timePlan').hidden = mode !== 'time'; q('.amountPlan').hidden = mode !== 'amount';
qa('.planModes label').forEach(label => label.classList.toggle('active', label.querySelector('input').checked));
}); }); async function requestPlan() { if (!calculation) return; const mode = value('plan_mode'); let planValue;
if (mode === 'time') { planValue = value('goal_months'); } else { planValue = value('monthly_amount'); if
(!planValue || Number(planValue) <= 0) { form.elements.monthly_amount.focus(); return; } } const button =
q('.updatePlan'); button.disabled = true; button.textContent = 'Calculatingâ?¦'; await refreshNonce(); const fd
= new FormData(); fd.append('action', 'aa_efc_master_plan'); fd.append('nonce', nonce); fd.append('data',
JSON.stringify(inputData)); fd.append('selected_months', selectedSafetyMonths); fd.append('plan_mode',
mode); fd.append('plan_value', planValue); try { const response = await fetch(ajaxUrl, { method:'POST',
credentials:'same-origin', body: fd }); const json = await response.json(); if (!json.success) { throw new
Error(json.data?.message || 'Unable to create your plan.'); } calculation = json.data.calculation; plan =
json.data.plan; renderPlan(); pushEvent('emergency_calculator_plan_created', { selected_months:
plan.selected_months, completion_months: plan.completion_months, selected_progress:
plan.selected_progress, capped: plan.completion_capped }); } catch (error) { alert(error.message || 'Unable
to create your plan.'); } finally { button.disabled = false; button.textContent = 'Calculate My Monthly Plan
â??'; } } safeBind('.updatePlan', 'click', requestPlan); function renderPlan() { q('.planResult').hidden = false;
q('.comparisonRecommended').textContent = calculation.recommended_months + ' months â?? ' +
money(calculation.recommended_fund); q('.comparisonSelected').textContent = plan.selected_months + '
months â?? ' + money(plan.selected_target); q('.roadmapSavings').textContent =
money(calculation.savings); q('.selectedProgress').textContent = Math.round(plan.selected_progress) + '%';
q('.selectedProgressTrack i').style.width = Math.min(100, plan.selected_progress) + '%'; if
(plan.selected_progress >= 100) { q('.progressMessage').textContent = 'Excellent â?? you have already
reached this chosen emergency-fund target.'; } else if (plan.selected_progress >= 75) {
q('.progressMessage').textContent = 'You are close. Most of this chosen safety target is already funded.'; }
else if (plan.selected_progress >= 50) { q('.progressMessage').textContent = 'Good progress â?? you are
already more than halfway toward your chosen target.'; } else if (plan.selected_progress > 0) {
q('.progressMessage').textContent = 'You have already started. Keep building one milestone at a time.'; }
else { q('.progressMessage').textContent = 'Start with your first 1-month safety milestone instead of worrying
about the full target today.'; } q('.planTarget').textContent = money(plan.selected_target);
q('.planGap').textContent = money(plan.gap); q('.planMonthly').textContent =
money(plan.monthly_contribution); q('.planDuration').textContent = Number(plan.gap) <= 0 ? 'Already
Achieved' : (plan.completion_capped ? '10+ years' : plan.completion_months + ' months');
q('.affordabilityBand').textContent = plan.affordability_band; q('.affordabilityText').textContent =
plan.affordability_text; q('.timelineNote').textContent = plan.timeline_note; q('.actionTwoTitle').textContent =
plan.monthly_contribution > 0 ? 'Automate ' + money(plan.monthly_contribution) + '/month' : 'Protect Your
Current Reserve'; q('.actionTwoText').textContent = plan.monthly_contribution > 0 ? 'Set an automatic
monthly transfer so your emergency fund grows without depending on memory or motivation.' : 'Keep your
completed emergency reserve separate and easily accessible for genuine emergencies.';
q('.actionThreeTitle').textContent = Number(plan.gap) <= 0 ? 'Review Your Reserve Periodically' : 'Review
at Your ' + plan.selected_months + '-Month Safety Milestone'; if (Number(plan.gap) <= 0) {
q('.actionThreeText').textContent = 'Review your emergency reserve whenever your income, EMI, essential
expenses or family responsibilities materially change.'; } else if (Number(plan.selected_months) <
Number(calculation.recommended_months)) { q('.actionThreeText').textContent = 'After reaching ' +
money(plan.selected_target) + ', review whether to continue toward your personalized ' +
calculation.recommended_months + '-month recommendation of ' + money(calculation.recommended_fund)
```

+ '.'; } else { q('.actionThreeText').textContent = 'Once you reach your chosen target, review the emergency reserve whenever income, EMI, essential expenses or family responsibilities materially change.'; } if (Number(plan.selected_months) < Number(calculation.recommended_months)) { q('.recommendedReminderText').textContent = 'Your ' + plan.selected_months + '-month target is an important first milestone â?? not a replacement for your personalized recommendation. Your recommended safety reserve remains ' + calculation.recommended_months + ' months â?? ' + money(calculation.recommended_fund) + '.'; } else { q('.recommendedReminderText').textContent = 'Your chosen target matches or exceeds the calculator recommendation. Once achieved, keep the reserve liquid and review it periodically as your financial situation changes.'; } q('.planResult').scrollIntoView({ behavior:'smooth', block:'start' }); } if (form) { form.addEventListener('submit', async function(event){ event.preventDefault(); if (!plan) { q('.status').textContent = 'Please calculate your monthly action plan first.'; return; } const status = q('.status'); const name = value('lead_name').trim(); let phone = value('lead_phone').replace(/\D/g, ''); const email = value('lead_email').trim(); const consent = form.elements.consent.checked; if (name.length < 2) { form.elements.lead_name.focus(); status.textContent = 'Please enter your name.'; return; } if (phone.length === 12 && phone.startsWith('91')) { phone = phone.substring(2); } if (!/^[6-9][0-9]{9}$/.test(phone)) { form.elements.lead_phone.focus(); status.textContent = 'Please enter a valid 10-digit Indian mobile number.'; return; } if (!consent) { status.textContent = 'Please accept the consent checkbox to receive your report.'; return; } const submit = q('.submitLead'); submit.disabled = true; status.textContent = 'Creating your personalized action reportâ?¦'; const params = new URLSearchParams(location.search); await refreshNonce(); const fd = new FormData(); fd.append('action', 'aa_efc_master_save_lead'); fd.append('nonce', nonce); fd.append('name', name); fd.append('phone', phone); fd.append('email', email); fd.append('website', value('website')); fd.append('data', JSON.stringify(inputData)); fd.append('selected_months', plan.selected_months); fd.append('plan_mode', plan.plan_mode); fd.append('plan_value', plan.plan_mode === 'time' ? plan.completion_months : plan.monthly_contribution); fd.append('wants_help', value('wants_help')); fd.append('consent', consent ? '1' : '0'); fd.append('page_url', location.href); ['utm_source','utm_medium','utm_campaign','utm_content','utm_term'].forEach(key => fd.append(key, params.get(key) || '')); try { const response = await fetch(ajaxUrl, { method:'POST', credentials:'same-origin', body: fd }); const json = await response.json(); if (!json.success) { throw new Error(json.data?.message || 'Unable to save your report.'); } calculation = json.data.calculation; plan = json.data.plan; status.textContent = 'Your personalized report is ready.'; leadCreated = true; submit.textContent = 'â?? Action Plan Created'; q('.afterLead').hidden = false; q('.healthCheckCTA').hidden = false; const actionMessage = 'Hi Ali Asgar, I completed the Emergency Fund Calculator. My Emergency Readiness Score is ' + calculation.score + '/100. My personalized recommendation is ' + calculation.recommended_months + ' months / ' + money(calculation.recommended_fund) + '. My chosen starting target is ' + plan.selected_months + ' months / ' + money(plan.selected_target) + '. My remaining chosen-target gap is ' + money(plan.gap) + '. Please help me review my action plan.'; q('.whatsappBtn').href = 'https://wa.me/' + whatsappNumber + '?text=' + encodeURIComponent(actionMessage); const healthMessage = 'Hi Ali Asgar, I completed the Emergency Fund Calculator. My Emergency Readiness Score is ' + calculation.score + '/100. I would now like to check the other important areas of my family\'s financial safety through the Family Financial Health Check.'; q('.healthCheckBtn').href = 'https://wa.me/' + whatsappNumber + '?text=' + encodeURIComponent(healthMessage); pushEvent('emergency_calculator_lead_submitted', { score: calculation.score, selected_months: plan.selected_months, wants_help: value('wants_help') }); } catch (error) { status.textContent = error.message || 'Something went wrong. Please try again.'; } finally { submit.disabled = leadCreated; } }); } async function loadPdfLibrary() { if (window.jspdf) return; const sources = ['https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js', 'https://cdn.jsdelivr.net/npm/jspdf@2.5.1/dist/jspdf.umd.min.js']; let lastError = null; for (const src of sources) { try { await new Promise((resolve, reject) => { const script = document.createElement('script');

```
script.src = src; script.async = true; script.onload = resolve; script.onerror = reject;
document.head.appendChild(script); }); if (window.jspdf) return; } catch (error) { lastError = error; } } throw
lastError || new Error('PDF library could not be loaded.'); } /* FINAL PDF: deliberately structured as a
balanced 3-page report. Page 1 = Diagnosis, Page 2 = Roadmap, Page 3 = Next financial step. This avoids
both content overlap and sparse accidental overflow pages. */ async function createPdf() { if (!calculation ||
!plan) return; try { await loadPdfLibrary(); const { jsPDF } = window.jspdf; const doc = new jsPDF({ unit:'mm',
format:'a4' }); const name = value('lead_name').trim() || 'Client'; const LEFT = 18; const RIGHT = 192; const
generatedDate = new Date().toLocaleDateString('en-IN', { day:'2-digit', month:'short', year:'numeric' });
function background() { doc.setFillColor(7, 20, 33); doc.rect(0, 0, 210, 297, 'F'); doc.setDrawColor(36, 54,
70); doc.line(LEFT, 23, RIGHT, 23); } function brand() { doc.setTextColor(220, 184, 95);
doc.setFont('helvetica', 'bold'); doc.setFontSize(8.5); doc.text('ALI ASGAR | FINANCIAL SAFETY ENGINE',
LEFT, 16); } function footer(pageNo) { doc.setDrawColor(34, 49, 64); doc.line(LEFT, 282, RIGHT, 282);
doc.setTextColor(82, 99, 115); doc.setFont('helvetica', 'normal'); doc.setFontSize(6.2); doc.text('Educational
planning report â?¢ Not individualized financial advice', LEFT, 289); doc.text('Page ' + pageNo + ' of 3',
RIGHT, 289, { align:'right' }); } function section(text, y) { doc.setTextColor(220, 184, 95);
doc.setFont('helvetica', 'bold'); doc.setFontSize(7.8); doc.text(String(text).toUpperCase(), LEFT, y); }
function body(text, y, width = 174, size = 7.7, x = LEFT) { doc.setTextColor(194, 204, 215);
doc.setFont('helvetica', 'normal'); doc.setFontSize(size); const lines = doc.splitTextToSize(String(text),
width); doc.text(lines, x, y); return lines.length * 4.05; } function row(label, amount, y, height = 10) {
doc.setFillColor(14, 31, 47); doc.roundedRect(LEFT, y, 174, height, 2, 2, 'F'); doc.setTextColor(145, 159,
175); doc.setFont('helvetica', 'normal'); doc.setFontSize(7.5); doc.text(String(label), 23, y + 6.5);
doc.setTextColor(255, 255, 255); doc.setFont('helvetica', 'bold'); doc.text(String(amount), 187, y + 6.5, {
align:'right' }); } function drawWhyReason(item, y) { const prefix = item.label + ': +' + item.months + ' month'
+ (Number(item.months) === 1 ? '' : 's') + '. '; doc.setFont('helvetica', 'bold'); doc.setFontSize(6.7);
doc.setTextColor(241, 212, 130); doc.text(prefix, LEFT, y); const prefixWidth = Math.min(64,
doc.getTextWidth(prefix) + 2); doc.setFont('helvetica', 'normal'); doc.setTextColor(187, 199, 211); const
available = 174 - prefixWidth; let reasonLines = doc.splitTextToSize(item.reason, Math.max(70, available));
if (reasonLines.length === 1) { doc.text(reasonLines, LEFT + prefixWidth, y); return 7.2; } // If the reason
wraps, put it on its own next line for readability. doc.text(doc.splitTextToSize(item.reason, 166), LEFT + 4, y
+ 4.2); return 10.3; } function startPage(pageNo) { if (pageNo > 1) doc.addPage(); background(); brand(); }
/* ============================================================ PAGE 1 â?? DIAGNOSIS
============================================================ */ startPage(1);
doc.setTextColor(255, 255, 255); doc.setFont('times', 'bold'); doc.setFontSize(22); doc.text('Emergency
Fund Action Report', LEFT, 37); doc.setTextColor(143, 158, 174); doc.setFont('helvetica', 'normal');
doc.setFontSize(8); doc.text('Personalized educational assessment', LEFT, 44); doc.setFillColor(14, 31,
47); doc.roundedRect(15, 51, 180, 32, 4, 4, 'F'); doc.setTextColor(220, 184, 95); doc.setFont('helvetica',
'bold'); doc.setFontSize(7); doc.text('PREPARED FOR', 22, 60); doc.setTextColor(255, 255, 255);
doc.setFont('times', 'bold'); doc.setFontSize(15.5); doc.text(name.substring(0, 42), 22, 70);
doc.setTextColor(241, 212, 130); doc.setFont('helvetica', 'bold'); doc.setFontSize(9.3); doc.text('Emergency
Readiness Score: ' + calculation.score + '/100 - ' + calculation.score_band, 22, 79); doc.setTextColor(131,
147, 163); doc.setFont('helvetica', 'normal'); doc.setFontSize(6.5); doc.text('Generated: ' + generatedDate,
188, 79, { align:'right' }); section('Information You Provided', 94); let y = 100; const inputs = [ ['Monthly
income', pdfMoney(calculation.income)], ['Essential expenses', pdfMoney(calculation.expenses)], ['Monthly
EMI', pdfMoney(calculation.emi)], ['Essential monthly outflow', pdfMoney(calculation.outflow)], ['Current
emergency savings', pdfMoney(calculation.savings)], ['Financial dependents',
String(calculation.dependents)], ]; inputs.forEach(item => { row(item[0], item[1], y, 9); y += 10; });
section('Your Emergency Fund Calculation', 168); doc.setTextColor(255, 255, 255); doc.setFont('times',
'bold'); doc.setFontSize(15); doc.text(pdfMoney(calculation.outflow) + ' x ' +
```

calculation.recommended_months + ' months = ' + pdfMoney(calculation.recommended_fund), LEFT, 179); const gapText = Number(calculation.gap) <= 0 ? 'Your family needs about ' + pdfMoney(calculation.outflow) + ' each month for essential commitments. Based on your answers, a ' + calculation.recommended_months + '-month reserve is recommended. You already have more than the recommended emergency reserve, so no additional emergency-fund gap is currently identified.' : 'Your family needs about ' + pdfMoney(calculation.outflow) + ' each month for essential commitments. Based on your answers, a ' + calculation.recommended_months + '-month reserve is recommended, giving you a target of ' + pdfMoney(calculation.recommended_fund) + '. You already have ' + pdfMoney(calculation.savings) + ', so ' + pdfMoney(calculation.gap) + ' remains to reach the recommended level.'; body(gapText, 190, 174, 7.2); section('Why ' + calculation.recommended_months + ' Months?', 214); let whyY = 223; calculation.why.forEach(item => { whyY += drawWhyReason(item, whyY); }); footer(1); /* == PAGE 2 â?? PERSONAL ROADMAP == */ startPage(2); doc.setTextColor(241, 212, 130); doc.setFont('times', 'italic'); doc.setFontSize(11.8); doc.text('"An ounce of prevention is worth a pound of cure."', 105, 34, { align:'center' }); doc.setTextColor(145, 159, 175); doc.setFont('helvetica', 'normal'); doc.setFontSize(7); doc.text('- Benjamin Franklin', 105, 41, { align:'center' }); doc.setTextColor(255, 255, 255); doc.setFont('times', 'bold'); doc.setFontSize(19); doc.text('Your Personal Emergency Fund Roadmap', LEFT, 55); doc.setTextColor(172, 185, 198); doc.setFont('helvetica', 'normal'); doc.setFontSize(7.8); doc.text('You do not need to solve everything today. Build one level of financial safety at a time.', LEFT, 63); section('Recommended vs Your Starting Target', 76); row('Calculator recommendation', calculation.recommended_months + ' months - ' + pdfMoney(calculation.recommended_fund), 82, 10); row('Your chosen starting target', plan.selected_months + ' months - ' + pdfMoney(plan.selected_target), 93, 10); doc.setFillColor(20, 38, 55); doc.setDrawColor(105, 88, 48); doc.roundedRect(15, 109, 180, 30, 4, 4, 'FD'); doc.setTextColor(220, 184, 95); doc.setFont('helvetica', 'bold'); doc.setFontSize(6.9); doc.text('YOUR PROGRESS (TOWARD CHOSEN TARGET)', 22, 117); doc.setTextColor(255, 255, 255); doc.setFont('times', 'bold'); doc.setFontSize(14); doc.text(pdfMoney(calculation.savings) + ' of ' + pdfMoney(plan.selected_target), 22, 127); doc.setTextColor(241, 212, 130); doc.setFontSize(15.5); doc.text(Math.round(plan.selected_progress) + '% COMPLETE', 187, 127, { align:'right' }); doc.setFillColor(44, 58, 72); doc.roundedRect(22, 132, 165, 3.5, 2, 2, 'F'); const progressWidth = 165 * (Math.min(100, plan.selected_progress) / 100); if (progressWidth > 0) { doc.setFillColor(220, 184, 95); doc.roundedRect(22, 132, progressWidth, 3.5, 2, 2, 'F'); } section('Your Monthly Action Plan', 151); const actionRows = [['Remaining gap', pdfMoney(plan.gap)], ['Monthly action', pdfMoney(plan.monthly_contribution) + '/month'], [Number(plan.gap) <= 0 ? 'Target status' : 'Estimated completion', Number(plan.gap) <= 0 ? 'Already Achieved' : (plan.completion_capped ? 'More than 10 years' : plan.completion_months + ' months')], ['Plan style', plan.affordability_band],]; let arY = 157; actionRows.forEach(item => { row(item[0], item[1], arY, 9.5); arY += 10.5; }); section('Your Next 3 Actions', 207); const actionOne = '01 Separate the Money'; const actionOneText = 'Keep your emergency reserve separate from normal spending and long-term investments.'; const actionTwo = plan.monthly_contribution > 0 ? '02 Automate ' + pdfMoney(plan.monthly_contribution) + '/month' : '02 Protect Your Current Reserve'; const actionTwoText = plan.monthly_contribution > 0 ? 'Set an automatic monthly transfer so your emergency fund grows without depending on memory.' : 'Keep your completed emergency reserve liquid, separate and available for genuine emergencies.'; const actionThree = Number(plan.gap) <= 0 ? '03 Review Your Reserve Periodically' : '03 Review at Your ' + plan.selected_months + '-Month Safety Milestone'; let actionThreeText; if (Number(plan.gap) <= 0) { actionThreeText = 'Review your emergency reserve whenever income, EMI, essential expenses or family responsibilities materially change.'; } else if (Number(plan.selected_months) < Number(calculation.recommended_months)) { actionThreeText = 'Once your reserve reaches ' + pdfMoney(plan.selected_target) + ', review whether to continue toward your personalized ' + calculation.recommended_months + '-month recommendation of ' +

pdfMoney(calculation.recommended_fund) + '.'; } else { actionThreeText = 'Once achieved, review your reserve whenever income, EMI, essential expenses or family responsibilities materially change.'; } const actions = [[actionOne, actionOneText], [actionTwo, actionTwoText], [actionThree, actionThreeText]]; let actY = 216; actions.forEach(action => { doc.setTextColor(241, 212, 130); doc.setFont('helvetica', 'bold'); doc.setFontSize(7.7); doc.text(action[0], LEFT, actY); actY += 5; actY += body(action[1], actY, 174, 6.9); actY += 4; }); footer(2); /* == PAGE 3 â?? NEXT FINANCIAL STEP == */ startPage(3); doc.setTextColor(255, 255, 255); doc.setFont('times', 'bold'); doc.setFontSize(21); doc.text('Your Next Financial Step', LEFT, 39); doc.setTextColor(172, 185, 198); doc.setFont('helvetica', 'normal'); doc.setFontSize(8); doc.text('Protect the emergency reserve first, then review the wider financial safety of your family.', LEFT, 48); doc.setFillColor(17, 34, 50); doc.setDrawColor(107, 89, 48); doc.roundedRect(15, 58, 180, 38, 4, 4, 'FD'); doc.setTextColor(220, 184, 95); doc.setFont('helvetica', 'bold'); doc.setFontSize(7); doc.text('REMEMBER', 22, 67); doc.setTextColor(255, 255, 255); doc.setFont('times', 'bold'); doc.setFontSize(12.5); let rememberTitle; let rememberText; if (Number(plan.selected_months) < Number(calculation.recommended_months)) { rememberTitle = 'Your starting milestone is not the final recommendation.'; rememberText = 'Your ' + plan.selected_months + '-month target is an important first milestone, not a replacement for your personalized recommendation. Your recommended safety reserve remains ' + calculation.recommended_months + ' months - ' + pdfMoney(calculation.recommended_fund) + '.'; } else if (Number(plan.gap) <= 0) { rememberTitle = 'Your chosen safety target is already achieved.'; rememberText = 'Keep the required reserve liquid and accessible. Review it periodically as income, EMI, essential expenses or family responsibilities change.'; } else { rememberTitle = 'Stay focused on the plan you selected.'; rememberText = 'Your chosen target matches or exceeds the calculator recommendation. Build it steadily and review the reserve whenever your financial situation materially changes.'; } doc.text(rememberTitle, 22, 77); body(rememberText, 86, 164, 6.8, 22); doc.setFillColor(20, 38, 55); doc.setDrawColor(109, 91, 49); doc.roundedRect(15, 108, 180, 102, 5, 5, 'FD'); doc.setTextColor(220, 184, 95); doc.setFont('helvetica', 'bold'); doc.setFontSize(7.2); doc.text('RECOMMENDED NEXT STEP', 105, 120, { align:'center' }); doc.setTextColor(255, 255, 255); doc.setFont('times', 'bold'); doc.setFontSize(18); doc.text('Family Financial Health Check', 105, 133, { align:'center' }); const healthCopy = "Your emergency fund is only one part of your family's financial safety. A structured review can help you identify what's already strong, where protection gaps may exist, and what deserves attention next."; doc.setTextColor(190, 201, 212); doc.setFont('helvetica', 'normal'); doc.setFontSize(7.4); doc.text(doc.splitTextToSize(healthCopy, 154), 105, 145, { align:'center' }); doc.setTextColor(241, 212, 130); doc.setFont('helvetica', 'bold'); doc.setFontSize(7); doc.text('WHAT THE CHECK REVIEWS', 105, 166, { align:'center' }); doc.setTextColor(196, 207, 218); doc.setFont('helvetica', 'normal'); doc.setFontSize(7); doc.text('Income Protection â?¢ Health Protection â?¢ Debt â?¢ Children\'s Goals', 105, 176, { align:'center' }); doc.text('Retirement â?¢ Nominations â?¢ Financial Documents', 105, 184, { align:'center' }); doc.setFillColor(220, 184, 95); doc.roundedRect(48, 192, 114, 11, 5, 5, 'F'); doc.setTextColor(7, 20, 33); doc.setFont('helvetica', 'bold'); doc.setFontSize(8); doc.text('CHECK MY FAMILY\'S FINANCIAL HEALTH', 105, 199, { align:'center' }); const healthMessagePdf = 'Hi Ali Asgar, I completed the Emergency Fund Calculator. My Emergency Readiness Score is ' + calculation.score + '/100. I would like to check the other important areas of my family\'s financial safety.'; const healthUrl = 'https://wa.me/' + whatsappNumber + '?text=' + encodeURIComponent(healthMessagePdf); if (typeof doc.link === 'function') { doc.link(48, 192, 114, 11, { url: healthUrl }); } doc.setTextColor(145, 159, 175); doc.setFont('helvetica', 'normal'); doc.setFontSize(6.7); doc.text('Optional â?¢ Educational â?¢ No obligation', 105, 218, { align:'center' }); doc.setTextColor(255, 255, 255); doc.setFont('times', 'bold'); doc.setFontSize(12.5); doc.text('Ali Asgar | Financial Safety Engine', LEFT, 235); doc.setTextColor(173, 186, 198); doc.setFont('helvetica', 'normal'); doc.setFontSize(7.2); doc.text('Website: aliasgar.in', LEFT, 243); doc.text('WhatsApp: +91 85888 84976', LEFT, 250); doc.setDrawColor(37, 53, 68); doc.line(LEFT, 258, RIGHT, 258); doc.setTextColor(103, 119, 136); doc.setFont('helvetica', 'normal'); doc.setFontSize(5.8); const disclaimer = 'Educational planning tool. Results are indicative and based on the information you provide and the calculator assumptions. Actual emergency-fund needs may vary according to income stability, dependents, liabilities, insurance protection, health circumstances and other personal factors. This report is not a guarantee, investment recommendation, insurance recommendation or substitute for individualized financial advice.'; doc.text(doc.splitTextToSize(disclaimer, 174), LEFT, 266); footer(3); const safeName = name.replace(/[^a-z0-9]+/gi, '-').replace(/^-|-$/g, '').substring(0, 40); doc.save('Ali-Asgar-Emergency-Fund-Action-Plan-' + safeName + '.pdf'); pushEvent('emergency_calculator_pdf_downloaded'); } catch (error) { console.error(error); alert('PDF generation could not start. Please try again.'); } } safeBind('.downloadPdf', 'click', createPdf); safeBind('.whatsappBtn', 'click', function(){ pushEvent('emergency_calculator_whatsapp_clicked'); }); safeBind('.healthCheckBtn', 'click', function(){ pushEvent('family_financial_health_check_clicked'); }); if (!form) { console.error('Emergency Fund Calculator: form could not be initialized.'); } pushEvent('emergency_calculator_loaded'); })();

Date Created
August 17, 2026
Author
admin