YouTube earnings

Description

Û?Ù?Ù¹Û?Ù?Ø¨ Û? Ú©Ù Ø§Ø¦Û? Ø§Ù?Ø± Ø§Û?Ú?Ù?Ø±Ù¹Ø§Ø¦Ø²Ù?Ú¯

Ø¨Ø¬Ù¹ Ø§Ù?Ø± ØÚ©Ù Øª Ø¹Ù Ù?Û? ØªÛ?Ø§Ø± Ú©Ø±Û?Úº

YouTube Dual Report â?? Ads & Creator Earnings

Choose package â?? answer 10 questions â?? get an instant business report (USD & PKR). Download PDF, email it, or save to Google Sheets.

Version: Enhanced

Package C â?? Dual (creator-focused) ▼

USD â?? PKR rate 285

Calculate Report | Export PDF | Email Report | Save to Google Sheets | Reset

Report

Currency rate: 285

Earnings vs Cost

`; html += `
ROI (advertiser): ${res.roiVal === null ? 'N/A' : ((res.roiVal*100).toFixed(1) + '%') }
Platform Est. Share (mid): $${res.platformUSD.mid.toFixed(2)}
`; html += `
`; // Earnings table html += `

RPM (USD/1000)	Creator USD	Creator PKR	Platform U
${rpmScenarios[s].toFixed(2)}	$${e.toFixed(2)}	PKR ${Math.round(e*res.usdPkr).toLocaleString()}	$${res.platformUS

`; // Breakdown html += `

Details

`; for(const k in res.breakdown){ html += `
${k}: ${res.breakdown[k]}
`; } html += `

`; document.getElementById('reportSummary').innerHTML = html; // Chart const ctx = document.getElementById('earnChart').getContext('2d'); const labels = ['Low RPM','Mid RPM','High RPM']; const earnData = [res.earningsUSD.low, res.earningsUSD.mid, res.earningsUSD.high]; const adCost = res.adCost || 0; const datasets = [{ label:'Creator Earnings (USD)', data: earnData, backgroundColor:'rgba(59,130,246,0.8)' }, { label:'Ad Cost (USD)', data: [adCost,adCost,adCost], backgroundColor:'rgba(107,114,128,0.6)' }]; if(chartInstance) chartInstance.destroy(); chartInstance = new Chart(ctx, { type:'bar', data:{ labels, datasets }, options:{ responsive:true, scales:{ y:{ beginAtZero:true } } } }); // Save last result for export/email window._lastReport = res; } /* Buttons: Calculate, PDF, Email, Save, Reset */ document.getElementById('calculateBtn').addEventListener('click', ()=>{ // validate required const required = formWrap.querySelectorAll('[required]'); for(const r of required){ if(!r.value){ r.focus(); alert('Please fill required fields.'); return; } } const vals = collectValues(); const pkg = pkgSelect.value; const res = calcReport(pkg, vals); renderReport(res); }); // PDF export using html2canvas + jsPDF document.getElementById('downloadPdfBtn').addEventListener('click', async ()=>{ if(!window._lastReport){ alert('Please calculate the report first.'); return; } const area = document.getElementById('reportArea'); // temporarily ensure visible for canvas area.style.display='block'; const canvas = await html2canvas(area, { scale: 1.4, useCORS:true }); const imgData = canvas.toDataURL('image/png'); const { jsPDF } = window.jspdf; const pdf = new jsPDF('p','mm','a4'); const imgProps = pdf.getImageProperties(imgData); const pdfWidth = pdf.internal.pageSize.getWidth(); const pdfHeight = (imgProps.height * pdfWidth) / imgProps.width; pdf.addImage(imgData,'PNG',0,0,pdfWidth,pdfHeight); pdf.save('youtube_report.pdf'); }); // Email via EmailJS (if configured) document.getElementById('emailBtn').addEventListener('click', async ()=>{ if(!window._lastReport){ alert('Calculate report first.'); return; } if(!EMAILJS_SERVICE_ID || !EMAILJS_TEMPLATE_ID || !EMAILJS_USER_ID){ alert('EmailJS not configured. Fill keys in the script section to enable.'); return; } // collect a recipient email quickly const to = prompt('Enter recipient email:'); if(!to) return; // Prepare payload (you must set corresponding template fields in EmailJS) const payload = { to_email: to,

subject: 'Your YouTube Report', message: JSON.stringify(window._lastReport, null, 2), // add other template fields as needed }; // init EmailJS emailjs.init(EMAILJS_USER_ID); try{ const res = await emailjs.send(EMAILJS_SERVICE_ID, EMAILJS_TEMPLATE_ID, payload); alert('Email sent (EmailJS response id: ' + res.status + ')'); }catch(err){ console.error(err); alert('Email send failed. See console for details.'); } }); // Save to Google Sheets via webhook (Apps Script) document.getElementById('saveSheetsBtn').addEventListener('click', async ()=>{ if(!window._lastReport){ alert('Calculate report first.'); return; } if(!SHEETS_WEBHOOK_URL){ alert('Sheets webhook not configured. Fill SHEETS_WEBHOOK_URL variable to enable.'); return; } const payload = { timestamp: new Date().toISOString(), package: pkgSelect.value, usdToPkr: document.getElementById('usdToPkr').value, report: window._lastReport }; try{ const r = await fetch(SHEETS_WEBHOOK_URL, { method:'POST', headers:{'Content-Type':'application/json'}, body:JSON.stringify(payload) }); if(r.ok){ alert('Saved to Google Sheets (webhook returned OK).'); } else { alert('Sheets save failed: '+r.statusText); } }catch(e){ console.error(e); alert('Save failed: see console'); } }); document.getElementById('resetBtn').addEventListener('click', ()=>{ renderTemplate(pkgSelect.value); document.getElementById('reportArea').classList.add('hidden'); }); /* Initialize optional libraries only if config provided */ if(EMAILJS_SERVICE_ID && EMAILJS_USER_ID){ try{ emailjs.init(EMAILJS_USER_ID); }catch(e){ console.warn('EmailJS init failed', e); } } /* Helpful: show placeholder usage instructions (console) */ console.log('YouTube Dual Report loaded. To enable Email send, fill EMAILJS_* variables. To enable Sheets saving, fill SHEETS_WEBHOOK_URL.');

Ask Questions