<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

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

<title>ColorCost — калькулятор</title>

<script src="https://telegram.org/js/telegram-web-app.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf-autotable/3.5.28/jspdf.plugin.autotable.min.js"></script>

<style>

\* { box-sizing: border-box; }

body {

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;

background: #f5f5f5;

padding: 0;

margin: 0;

max-width: 800px;

margin: 0 auto;

}

h2 {

margin: 0;

padding: 15px;

color: #fff;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

position: sticky;

top: 0;

z-index: 100;

box-shadow: 0 2px 10px rgba(0,0,0,0.15);

font-size: 20px;

}

.tabs {

display: flex;

background: #fff;

position: sticky;

top: 50px;

z-index: 99;

border-bottom: 2px solid #eee;

overflow-x: auto;

-webkit-overflow-scrolling: touch;

}

.tab {

flex: 1;

padding: 12px 10px;

text-align: center;

border: none;

background: transparent;

cursor: pointer;

font-size: 14px;

color: #666;

border-bottom: 3px solid transparent;

transition: all 0.3s;

white-space: nowrap;

min-width: 100px;

}

.tab.active {

color: #667eea;

border-bottom-color: #667eea;

font-weight: 600;

}

.tab-content {

display: none;

padding: 15px;

animation: fadeIn 0.3s;

}

.tab-content.active {

display: block;

}

@keyframes fadeIn {

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

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

}

.block {

background: #fff;

padding: 15px;

margin-bottom: 15px;

border-radius: 8px;

box-shadow: 0 2px 4px rgba(0,0,0,0.1);

}

.block h3 {

margin: 0 0 15px 0;

color: #333;

font-size: 16px;

}

label {

display: block;

margin: 0 0 4px 0;

font-size: 13px;

color: #666;

font-weight: 500;

}

input {

width: 100%;

padding: 12px;

margin-bottom: 12px;

border: 1px solid #ddd;

border-radius: 8px;

font-size: 16px;

-webkit-appearance: none;

}

input:focus {

outline: none;

border-color: #667eea;

box-shadow: 0 0 0 3px rgba(102, 126, 234, 0.1);

}

button {

padding: 14px 20px;

margin: 8px 8px 0 0;

border: none;

border-radius: 8px;

cursor: pointer;

font-size: 15px;

font-weight: 600;

transition: transform 0.2s, box-shadow 0.2s;

width: 100%;

touch-action: manipulation;

}

button:active {

transform: scale(0.98);

}

.btn-primary {

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

color: white;

box-shadow: 0 4px 15px rgba(102, 126, 234, 0.3);

}

.btn-secondary {

background: #2196F3;

color: white;

box-shadow: 0 4px 15px rgba(33, 150, 243, 0.3);

}

.btn-danger {

background: #f44336;

color: white;

padding: 10px 16px;

font-size: 14px;

margin: 8px 0 0 0;

}

.btn-warning {

background: #ff9800;

color: white;

box-shadow: 0 4px 15px rgba(255, 152, 0, 0.3);

}

.btn-clear {

background: #fff;

color: #f44336;

border: 2px solid #f44336;

box-shadow: none;

}

.row {

border: 2px solid #e8e8e8;

padding: 15px;

margin-bottom: 12px;

border-radius: 12px;

background: #fafafa;

position: relative;

}

.row-compact {

display: grid;

grid-template-columns: 1fr 1fr;

gap: 10px;

}

.row-compact label {

grid-column: 1 / -1;

}

.row-compact .field-group {

display: flex;

flex-direction: column;

}

.row-compact .field-group.full {

grid-column: 1 / -1;

}

pre {

background: #f9f9f9;

padding: 15px;

border-radius: 8px;

overflow-x: auto;

font-size: 13px;

line-height: 1.6;

border: 1px solid #e0e0e0;

max-height: 400px;

overflow-y: auto;

}

.error {

background: #ffebee;

color: #c62828;

padding: 12px 15px;

border-radius: 8px;

margin: 0 15px 15px 15px;

display: none;

font-size: 14px;

animation: slideDown 0.3s;

}

.success {

background: #e8f5e9;

color: #2e7d32;

padding: 12px 15px;

border-radius: 8px;

margin: 0 15px 15px 15px;

display: none;

font-size: 14px;

animation: slideDown 0.3s;

}

@keyframes slideDown {

from { transform: translateY(-20px); opacity: 0; }

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

}

.materials-summary {

display: flex;

justify-content: space-between;

padding: 10px;

background: #f0f0f0;

border-radius: 8px;

margin-bottom: 15px;

font-size: 13px;

}

.materials-summary div {

text-align: center;

}

.materials-summary .count {

font-size: 20px;

font-weight: bold;

color: #667eea;

}

.button-group {

display: grid;

grid-template-columns: 1fr 1fr;

gap: 8px;

}

.button-group button {

margin: 0;

}

.button-group .full {

grid-column: 1 / -1;

}

.fab {

position: fixed;

bottom: 20px;

right: 20px;

width: 60px;

height: 60px;

border-radius: 50%;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

color: white;

border: none;

font-size: 24px;

box-shadow: 0 4px 20px rgba(102, 126, 234, 0.5);

cursor: pointer;

z-index: 98;

display: flex;

align-items: center;

justify-content: center;

}

.fab:active {

transform: scale(0.95);

}

.empty-state {

text-align: center;

padding: 40px 20px;

color: #999;

}

.empty-state-icon {

font-size: 48px;

margin-bottom: 10px;

}

</style>

</head>

<body>

<h2>💇‍♀️ ColorCost</h2>

<div id="errorMsg" class="error"></div>

<div id="successMsg" class="success"></div>

<div class="tabs">

<button class="tab active" onclick="switchTab('paints')">Краски</button>

<button class="tab" onclick="switchTab('oxids')">Оксиды</button>

<button class="tab" onclick="switchTab('settings')">Настройки</button>

<button class="tab" onclick="switchTab('result')">Итог</button>

</div>

<div id="tab-paints" class="tab-content active">

<div class="materials-summary">

<div>

<div class="count" id="paints-count">0</div>

<div>материалов</div>

</div>

</div>

<div id="paints"></div>

<button class="btn-primary" onclick="addRow('paints')">+ Добавить краску</button>

</div>

<div id="tab-oxids" class="tab-content">

<div class="materials-summary">

<div>

<div class="count" id="oxids-count">0</div>

<div>материалов</div>

</div>

</div>

<div id="oxids"></div>

<button class="btn-primary" onclick="addRow('oxids')">+ Добавить оксид</button>

</div>

<div id="tab-settings" class="tab-content">

<div class="block">

<label>Наценка (%)</label>

<input id="margin" type="number" placeholder="30" min="0" value="30">

<label>Стоимость работы (₽)</label>

<input id="labor" type="number" placeholder="2000" min="0" value="2000">

</div>

</div>

<div id="tab-result" class="tab-content">

<div class="block">

<pre id="result">Добавьте материалы и нажмите "Рассчитать"</pre>

<div class="button-group">

<button class="btn-primary full" onclick="calculate()">🧮 Рассчитать</button>

<button class="btn-secondary" onclick="savePDF()">📄 PDF</button>

<button class="btn-warning" onclick="copyResult()">📋 Копировать</button>

<button class="btn-clear" onclick="clearAll()">🗑️ Очистить всё</button>

</div>

</div>

</div>

<button class="fab" onclick="calculate(); switchTab('result');" title="Рассчитать">🧮</button>

<script>

// Инициализация Telegram Web App

if (window.Telegram?.WebApp) {

const tg = window.Telegram.WebApp;

tg.ready();

tg.expand();

}

let lastCalculation = null;

// Загрузка сохранённых данных при старте

window.addEventListener('load', () => {

loadFromStorage();

updateCounts();

if (document.getElementById('paints').children.length === 0) {

addRow('paints');

}

if (document.getElementById('oxids').children.length === 0) {

addRow('oxids');

}

});

function showError(msg) {

const el = document.getElementById('errorMsg');

el.textContent = msg;

el.style.display = 'block';

setTimeout(() => { el.style.display = 'none'; }, 4000);

}

function showSuccess(msg) {

const el = document.getElementById('successMsg');

el.textContent = msg;

el.style.display = 'block';

setTimeout(() => { el.style.display = 'none'; }, 3000);

}

function addRow(containerId) {

const div = document.createElement('div');

div.className = 'row';

div.innerHTML = `

<div class="row-compact">

<div class="field-group full">

<label>Название</label>

<input type="text" class="name" placeholder="Напр. L'Oréal 8.1">

</div>

<div class="field-group">

<label>Объем (мл)</label>

<input type="number" class="ml" placeholder="60" min="0" step="0.1">

</div>

<div class="field-group">

<label>Цена (₽)</label>

<input type="number" class="price" placeholder="600" min="0" step="0.01">

</div>

<div class="field-group">

<label>Расход (мл)</label>

<input type="number" class="used" placeholder="30" min="0" step="0.1">

</div>

<div class="field-group">

<label>₽/мл</label>

<input type="text" class="perml" readonly style="background:#f0f0f0;" value="—">

</div>

</div>

<button class="btn-danger" onclick="this.parentNode.remove(); saveToStorage(); updateCounts();">Удалить</button>

`;

// Добавляем автосохранение при изменении

const inputs = div.querySelectorAll('input:not(.perml)');

inputs.forEach(input => {

input.addEventListener('change', saveToStorage);

input.addEventListener('input', () => updatePerMl(div));

});

document.getElementById(containerId).appendChild(div);

saveToStorage();

updateCounts();

}

function updatePerMl(row) {

const ml = parseFloat(row.querySelector('.ml').value) || 0;

const price = parseFloat(row.querySelector('.price').value) || 0;

const perMlInput = row.querySelector('.perml');

if (ml > 0) {

perMlInput.value = (price / ml).toFixed(2);

} else {

perMlInput.value = '—';

}

}

function updateCounts() {

const paintsCount = document.getElementById('paints').querySelectorAll('.row').length;

const oxidsCount = document.getElementById('oxids').querySelectorAll('.row').length;

document.getElementById('paints-count').textContent = paintsCount;

document.getElementById('oxids-count').textContent = oxidsCount;

}

function switchTab(tabName) {

// Скрываем все табы

document.querySelectorAll('.tab-content').forEach(tab => {

tab.classList.remove('active');

});

document.querySelectorAll('.tab').forEach(tab => {

tab.classList.remove('active');

});

// Показываем выбранный таб

document.getElementById(`tab-${tabName}`).classList.add('active');

event.target.classList.add('active');

}

function collectRows(containerId) {

const rows = document.getElementById(containerId).querySelectorAll('.row');

const items = [];

let hasError = false;

rows.forEach(r => {

const name = r.querySelector('.name').value.trim();

const ml = parseFloat(r.querySelector('.ml').value) || 0;

const price = parseFloat(r.querySelector('.price').value) || 0;

const used = parseFloat(r.querySelector('.used').value) || 0;

// Пропускаем полностью пустые строки

if (!name && ml === 0 && price === 0 && used === 0) return;

// Проверка на отрицательные значения

if (ml < 0 || price < 0 || used < 0) {

showError('Ошибка: все числовые значения должны быть неотрицательными');

hasError = true;

return;

}

// Проверка на заполненность

if (!name || ml === 0) {

showError('Ошибка: заполните название и объем для всех материалов');

hasError = true;

return;

}

const perMl = ml > 0 ? price / ml : 0;

const cost = perMl \* used;

items.push({ name, ml, price, used, perMl, cost });

});

if (hasError) return null;

return items;

}

function calculate() {

const paints = collectRows('paints');

const oxids = collectRows('oxids');

if (paints === null || oxids === null) return;

if (paints.length === 0 && oxids.length === 0) {

showError('Добавьте хотя бы один материал для расчёта');

return;

}

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

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

if (margin < 0 || labor < 0) {

showError('Наценка и стоимость работы не могут быть отрицательными');

return;

}

const costPaints = paints.reduce((s, i) => s + i.cost, 0);

const costOxids = oxids.reduce((s, i) => s + i.cost, 0);

const materials = costPaints + costOxids;

const marginVal = materials \* margin / 100;

const total = materials + marginVal + labor;

let text = '=== КРАСКИ ===\n';

if (paints.length === 0) {

text += 'Не добавлено\n';

} else {

paints.forEach(p => {

text += `${p.name}: ${p.used} мл × ${p.perMl.toFixed(2)} ₽/мл = ${p.cost.toFixed(2)} ₽\n`;

});

text += `Итого краски: ${costPaints.toFixed(2)} ₽\n`;

}

text += '\n=== ОКСИДЫ ===\n';

if (oxids.length === 0) {

text += 'Не добавлено\n';

} else {

oxids.forEach(o => {

text += `${o.name}: ${o.used} мл × ${o.perMl.toFixed(2)} ₽/мл = ${o.cost.toFixed(2)} ₽\n`;

});

text += `Итого оксиды: ${costOxids.toFixed(2)} ₽\n`;

}

text += '\n=== РАСЧЁТ ===\n';

text += `Себестоимость материалов: ${materials.toFixed(2)} ₽\n`;

text += `Наценка (${margin}%): ${marginVal.toFixed(2)} ₽\n`;

text += `Стоимость работы: ${labor.toFixed(2)} ₽\n`;

text += `\n💰 ИТОГО К ОПЛАТЕ: ${total.toFixed(2)} ₽`;

document.getElementById('result').textContent = text;

lastCalculation = { paints, oxids, margin, labor, materials, marginVal, total, costPaints, costOxids };

showSuccess('Расчёт выполнен успешно!');

return lastCalculation;

}

function savePDF() {

if (!lastCalculation) {

showError('Сначала выполните расчёт!');

return;

}

try {

const { jsPDF } = window.jspdf;

const doc = new jsPDF();

const data = lastCalculation;

// Заголовок

doc.setFillColor(76, 175, 80);

doc.rect(0, 0, 210, 35, "F");

doc.setTextColor(255, 255, 255);

doc.setFontSize(24);

doc.text("ColorCost", 15, 18);

doc.setFontSize(11);

doc.text("Raschet uslugi okrashivaniya", 15, 26);

doc.text(`Data: ${new Date().toLocaleString('ru-RU')}`, 130, 20);

// Таблица материалов

const rows = [];

[...data.paints, ...data.oxids].forEach(it => {

rows.push([

it.name || 'Material',

it.ml.toFixed(1),

it.price.toFixed(2),

it.used.toFixed(1),

it.perMl.toFixed(2),

it.cost.toFixed(2)

]);

});

doc.setTextColor(0, 0, 0);

doc.autoTable({

head: [['Nazvanie', 'Ob\'em (ml)', 'Cena (rub)', 'Rashod (ml)', 'rub/ml', 'Itog (rub)']],

body: rows,

startY: 45,

theme: 'grid',

headStyles: { fillColor: [76, 175, 80] }

});

let y = doc.lastAutoTable.finalY + 15;

doc.setFontSize(12);

doc.text(`Sebestoimost\': ${data.materials.toFixed(2)} rub`, 15, y);

y += 8;

doc.text(`Natsenka (${data.margin}%): ${data.marginVal.toFixed(2)} rub`, 15, y);

y += 8;

doc.text(`Rabota: ${data.labor.toFixed(2)} rub`, 15, y);

y += 12;

doc.setFontSize(16);

doc.setTextColor(76, 175, 80);

doc.text(`ITOGO: ${data.total.toFixed(2)} rub`, 15, y);

doc.save(`colorcost\_${new Date().toISOString().slice(0,10)}.pdf`);

showSuccess('PDF сохранён!');

} catch (error) {

showError('Ошибка при создании PDF: ' + error.message);

}

}

function copyResult() {

const text = document.getElementById('result').textContent;

if (text === 'Добавьте материалы и нажмите "Рассчитать"') {

showError('Сначала выполните расчёт!');

return;

}

navigator.clipboard.writeText(text).then(() => {

showSuccess('Результат скопирован в буфер обмена!');

}).catch(() => {

showError('Не удалось скопировать');

});

}

function clearAll() {

if (!confirm('Удалить все данные? Это действие нельзя отменить.')) {

return;

}

document.getElementById('paints').innerHTML = '';

document.getElementById('oxids').innerHTML = '';

document.getElementById('margin').value = '30';

document.getElementById('labor').value = '2000';

document.getElementById('result').textContent = 'Добавьте материалы и нажмите "Рассчитать"';

lastCalculation = null;

addRow('paints');

addRow('oxids');

saveToStorage();

updateCounts();

showSuccess('Все данные удалены');

}

// Автосохранение в localStorage

function saveToStorage() {

try {

const data = {

paints: Array.from(document.getElementById('paints').querySelectorAll('.row')).map(getRowData),

oxids: Array.from(document.getElementById('oxids').querySelectorAll('.row')).map(getRowData),

margin: document.getElementById('margin').value,

labor: document.getElementById('labor').value

};

localStorage.setItem('colorcost\_data', JSON.stringify(data));

} catch (e) {

console.error('Ошибка сохранения:', e);

}

}

function getRowData(row) {

return {

name: row.querySelector('.name').value,

ml: row.querySelector('.ml').value,

price: row.querySelector('.price').value,

used: row.querySelector('.used').value

};

}

function loadFromStorage() {

try {

const saved = localStorage.getItem('colorcost\_data');

if (!saved) return;

const data = JSON.parse(saved);

// Загрузка красок

document.getElementById('paints').innerHTML = '';

data.paints.forEach(item => {

addRow('paints');

const row = document.getElementById('paints').lastElementChild;

fillRow(row, item);

});

// Загрузка оксидов

document.getElementById('oxids').innerHTML = '';

data.oxids.forEach(item => {

addRow('oxids');

const row = document.getElementById('oxids').lastElementChild;

fillRow(row, item);

});

document.getElementById('margin').value = data.margin || '30';

document.getElementById('labor').value = data.labor || '2000';

updateCounts();

} catch (e) {

console.error('Ошибка загрузки:', e);

}

}

function fillRow(row, data) {

row.querySelector('.name').value = data.name || '';

row.querySelector('.ml').value = data.ml || '';

row.querySelector('.price').value = data.price || '';

row.querySelector('.used').value = data.used || '';

}

// Автосохранение для полей наценки и работы

document.getElementById('margin').addEventListener('change', saveToStorage);

document.getElementById('labor').addEventListener('change', saveToStorage);

</script>

</body>

</html>