!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<title>Honest Check‑in — Yes/No Questionnaire</title>

<meta name="description" content="Private yes/no questionnaire with a final comment box. Answers are emailed to the owner when submitted." />

<style>

:root { --bg:#0f172a; --card:#111827; --muted:#334155; --text:#e5e7eb; --accent:#22c55e; --accent2:#3b82f6; --danger:#ef4444; }

\*{box-sizing:border-box}

body{margin:0; font-family:system-ui,-apple-system,Segoe UI,Roboto,Ubuntu,Cantarell,'Helvetica Neue',Arial; background:linear-gradient(180deg,#0b1220,#0f172a 20%,#0f172a); color:var(--text);}

.wrap{max-width:720px;margin:40px auto;padding:20px}

.card{background:rgba(17,24,39,.9); border:1px solid #1f2937; border-radius:20px; box-shadow:0 20px 50px rgba(0,0,0,.4); overflow:hidden}

header{padding:28px 24px 10px}

header h1{margin:0;font-size:26px;letter-spacing:.2px}

header p{margin:6px 0 0;color:#cbd5e1}

.progress{height:10px;background:#0b1220;border-radius:999px;margin:20px 24px;overflow:hidden;border:1px solid #1f2937}

.bar{height:100%;width:0;background:linear-gradient(90deg,var(--accent2),var(--accent)); transition:width .25s ease}

.content{padding:8px 24px 24px}

.q{font-size:20px;line-height:1.4;margin:22px 0 14px}

.btnrow{display:flex;gap:12px;flex-wrap:wrap}

button{cursor:pointer;border:1px solid #334155;background:#0b1220;color:var(--text);padding:12px 16px;border-radius:14px;font-size:16px;transition:transform .05s ease, background .2s,border-color .2s}

button:hover{background:#111827}

button:active{transform:translateY(1px)}

.primary{background:#0b1a33;border-color:#1f3a6d}

.primary.sel{background:rgba(59,130,246,.15);border-color:var(--accent2)}

.danger.sel{background:rgba(239,68,68,.15);border-color:var(--danger)}

.nav{display:flex;justify-content:space-between;gap:10px;margin-top:16px}

.nav button{flex:1}

.muted{color:#94a3b8;font-size:14px}

.hide{display:none}

textarea{width:100%;min-height:120px;padding:12px 14px;border-radius:12px;border:1px solid #334155;background:#0b1220;color:var(--text);resize:vertical}

input[type="email"], input[type="text"]{width:100%;padding:12px 14px;border-radius:12px;border:1px solid #334155;background:#0b1220;color:var(--text)}

.pill{display:inline-block;padding:6px 10px;border-radius:999px;background:#0b1220;border:1px solid #1f2937;color:#cbd5e1;font-size:12px}

.footer{padding:16px 24px;border-top:1px solid #1f2937;background:#0b1220;color:#94a3b8;font-size:12px}

.success{border-color:#14532d !important;background:rgba(34,197,94,.1) !important}

.error{border-color:#7f1d1d !important;background:rgba(239,68,68,.08) !important}

.center{display:flex;align-items:center;justify-content:center}

a.link{color:#93c5fd}

</style>

</head>

<body>

<div class="wrap">

<div class="card" id="app">

<header>

<h1>Honest Check‑in</h1>

<p class="muted">This is a short private questionnaire. Please answer honestly. It should take about ~2 minutes.</p>

</header>

<div class="progress"><div class="bar" id="bar"></div></div>

<div class="content" id="screen-intro">

<p class="muted">Your answers are Yes/No, followed by a final comment box. When you submit, your responses will be sent to the owner of this form via email.</p>

<ul class="muted" style="margin:14px 0 18px; padding-left:18px">

<li>No skipping — choose Yes or No to proceed.</li>

<li>Be respectful and truthful.</li>

<li>Final slide has space for anything you want to say.</li>

</ul>

<label class="pill"><input id="consent" type="checkbox" style="vertical-align:-2px;margin-right:8px"> I understand and agree to proceed.</label>

<div class="nav" style="margin-top:18px">

<button id="start" class="primary">Start</button>

</div>

</div>

<div class="content hide" id="screen-question">

<div class="pill" id="stepPill">Question</div>

<div class="q" id="questionText"></div>

<div class="btnrow">

<button class="primary" id="btnYes">Yes</button>

<button class="danger" id="btnNo">No</button>

</div>

<div class="nav">

<button id="prev">Back</button>

<button id="next" class="primary">Next</button>

</div>

<p class="muted" id="timer" style="margin-top:10px"></p>

</div>

<div class="content hide" id="screen-final">

<div class="pill">Final comment (optional)</div>

<p class="q">Anything you want me to know? Clarify any Yes/No answers here.</p>

<textarea id="comment" placeholder="Write your message…"></textarea>

<div style="margin-top:14px">

<label class="muted" for="name">Your name (optional)</label>

<input id="name" type="text" placeholder="Name (optional)" />

</div>

<div style="margin-top:14px">

<label class="muted" for="email">Where should confirmation be sent?</label>

<input id="email" type="email" placeholder="Your email (optional — you’ll receive a copy)" />

</div>

<div class="nav" style="margin-top:18px">

<button id="backToQs">Back</button>

<button id="submit" class="primary">Submit</button>

</div>

<p class="muted" style="margin-top:10px">By submitting, your answers will be emailed to the owner. If email isn’t configured, this will open your mail app with a pre‑filled message.</p>

</div>

<div class="content hide center" id="screen-done">

<div>

<h2 style="margin:0 0 8px">Thanks for completing this</h2>

<p class="muted">Your responses were sent. You can close this tab.</p>

<div class="btnrow" style="margin-top:14px">

<button id="copyLink">Copy form link</button>

<button id="restart">Restart</button>

</div>

</div>

</div>

<div class="footer">

Built for a clear and respectful conversation. Your privacy matters.

</div>

</div>

</div>

<!-- EmailJS (client-side email). Replace placeholders below in CONFIG. -->

<script src="https://cdn.jsdelivr.net/npm/@emailjs/browser@4/dist/email.min.js"></script>

<script>

const CONFIG = {

EMAILJS\_PUBLIC\_KEY: "YOUR\_PUBLIC\_KEY",

EMAILJS\_SERVICE\_ID: "YOUR\_SERVICE\_ID",

EMAILJS\_TEMPLATE\_ID: "YOUR\_TEMPLATE\_ID",

OWNER\_EMAIL: "merilemarleen@gmail.com",

FORM\_TITLE: "Honest Check‑in: Yes/No Questionnaire",

QUESTIONS: [

"Do you acknowledge that you are our daughter’s father and have a duty toward her?",

"Since birth, have you supported her consistently in any way (financial, emotional, or physical presence)?",

"Are you willing to start providing consistent financial support from this month?",

"Will you agree to a clear written plan for support (amount + dates)?",

"Do you intend to be present for her important days (birthdays, school events, milestones)?",

"Will you respect me as her mother and avoid disrespectful words or actions in front of her?",

"Are your intentions with me romantic, or is your focus only co‑parenting?",

"Did you mean the promises you made about a future with me?",

"If yes, are you ready to act on those promises within the next few months?",

"Are you open to honest and respectful communication going forward?",

"Will you be transparent about your finances and personal situation when it affects our daughter?",

"Is there anyone else in your life who may interfere with our co‑parenting?",

"Do you truly love our daughter?",

"Do you intend to build a bond with her actively (spending time, calling, being present)?",

"Should I move on for good and focus only on peaceful co‑parenting with you?"

],

TARGET\_MINUTES: 2

};

const EMAILJS\_ENABLED = CONFIG.EMAILJS\_PUBLIC\_KEY && CONFIG.EMAILJS\_PUBLIC\_KEY !== "YOUR\_PUBLIC\_KEY"

&& CONFIG.EMAILJS\_SERVICE\_ID && CONFIG.EMAILJS\_SERVICE\_ID !== "YOUR\_SERVICE\_ID"

&& CONFIG.EMAILJS\_TEMPLATE\_ID && CONFIG.EMAILJS\_TEMPLATE\_ID !== "YOUR\_TEMPLATE\_ID";

try { if (EMAILJS\_ENABLED) emailjs.init({ publicKey: CONFIG.EMAILJS\_PUBLIC\_KEY }); } catch(e) {}

const qs = CONFIG.QUESTIONS.map((t, i) => ({ id: i+1, text: t, answer: null }));

let idx = -1;

const totalSteps = qs.length + 2;

const $ = sel => document.querySelector(sel);

const intro = $('#screen-intro');

const qscreen = $('#screen-question');

const final = $('#screen-final');

const done = $('#screen-done');

const bar = $('#bar');

const stepPill = $('#stepPill');

const qText = $('#questionText');

const btnYes = $('#btnYes');

const btnNo = $('#btnNo');

const prev = $('#prev');

const next = $('#next');

const timerEl = $('#timer');

const startBtn = $('#start');

const consent = $('#consent');

const backToQs = $('#backToQs');

const submitBtn = $('#submit');

const comment = $('#comment');

const nameEl = $('#name');

const emailEl = $('#email');

const copyLink = $('#copyLink');

const restart = $('#restart');

const TARGET\_SECONDS = CONFIG.TARGET\_MINUTES \* 60;

let startTime = null;

function setView(which){

intro.classList.toggle('hide', which !== 'intro');

qscreen.classList.toggle('hide', which !== 'qs');

final.classList.toggle('hide', which !== 'final');

done.classList.toggle('hide', which !== 'done');

}

function updateProgress(){

const step = Math.min(idx + 2, totalSteps);

const pct = Math.round((step / (totalSteps+1)) \* 100);

bar.style.width = pct + '%';

}

function showQuestion(){

setView('qs');

stepPill.textContent = `Question ${idx+1} of ${qs.length}`;

qText.textContent = qs[idx].text;

[btnYes, btnNo].forEach(b=>b.classList.remove('sel'));

if(qs[idx].answer === 'Yes') btnYes.classList.add('sel');

if(qs[idx].answer === 'No') btnNo.classList.add('sel');

next.disabled = !qs[idx].answer;

prev.disabled = idx===0;

updateProgress();

updateTimer();

}

function updateTimer(){

if (!startTime) return;

const elapsed = Math.round((Date.now()-startTime)/1000);

const remain = Math.max(0, TARGET\_SECONDS - elapsed);

const m = Math.floor(remain/60); const s = remain%60;

timerEl.textContent = `~${m}:${String(s).padStart(2,'0')} remaining`;

}

startBtn.addEventListener('click', () => {

if(!consent.checked){

intro.classList.add('error');

setTimeout(()=>intro.classList.remove('error'), 600);

return;

}

startTime = Date.now();

idx = 0;

showQuestion();

});

btnYes.addEventListener('click', () => { qs[idx].answer = 'Yes'; btnYes.classList.add('sel'); btnNo.classList.remove('sel'); next.disabled=false; });

btnNo.addEventListener('click', () => { qs[idx].answer = 'No'; btnNo.classList.add('sel'); btnYes.classList.remove('sel'); next.disabled=false; });

prev.addEventListener('click', () => { if(idx>0){ idx--; showQuestion(); }});

next.addEventListener('click', () => {

if(!qs[idx].answer){ qscreen.classList.add('error'); setTimeout(()=>qscreen.classList.remove('error'), 600); return; }

if(idx < qs.length-1){ idx++; showQuestion(); }

else { idx = qs.length; setView('final'); updateProgress(); }

});

backToQs.addEventListener('click', () => { idx = qs.length-1; showQuestion(); });

submitBtn.addEventListener('click', async () => {

const answers = qs.map(q => ({ id:q.id, question:q.text, answer:q.answer }));

const yesCount = answers.filter(a=>a.answer==='Yes').length;

const noCount = answers.filter(a=>a.answer==='No').length;

const startedAt = new Date(startTime || Date.now()).toISOString();

const finishedAt = new Date().toISOString();

const durationSec = Math.round(((Date.now())-(startTime||Date.now()))/1000);

const payload = {

form\_title: CONFIG.FORM\_TITLE,

name: nameEl.value.trim() || '(no name)',

commenter\_email: emailEl.value.trim() || '(no email)',

comment: comment.value.trim(),

yes\_count: yesCount,

no\_count: noCount,

total: answers.length,

started\_at: startedAt,

finished\_at: finishedAt,

duration\_seconds: durationSec,

user\_agent: navigator.userAgent,

answers\_json: JSON.stringify(answers, null, 2)

};

if(EMAILJS\_ENABLED){

try{

submitBtn.disabled = true; submitBtn.textContent = 'Sending…';

const res = await emailjs.send(CONFIG.EMAILJS\_SERVICE\_ID, CONFIG.EMAILJS\_TEMPLATE\_ID, payload);

submitBtn.textContent = 'Sent';

setView('done');

return;

}catch(err){

console.warn('EmailJS failed, falling back to mailto', err);

} finally {

submitBtn.disabled = false; submitBtn.textContent = 'Submit';

}

}

const subject = encodeURIComponent(CONFIG.FORM\_TITLE);

const body = encodeURIComponent(

`Name: ${payload.name}\nEmail: ${payload.commenter\_email}\n\nComment:\n${payload.comment}\n\nSummary: Yes=${yesCount} / No=${noCount} / Total=${answers.length}\nDuration: ${durationSec}s\nStarted: ${startedAt}\nFinished: ${finishedAt}\nUA: ${payload.user\_agent}\n\nAnswers:\n${answers.map(a=>`Q${a.id}: ${a.question}\n→ ${a.answer}`).join('\n\n')}`);

const mailto = `mailto:${encodeURIComponent(CONFIG.OWNER\_EMAIL)}?subject=${subject}&body=${body}`;

window.location.href = mailto;

setView('done');