

Quirk AI Sales Kiosk code
(includes objection handling)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Quirk AI Sales Kiosk</title>
  <meta name="color-scheme" content="light" />
  <base href="/Quirk-AI-Project-Overview/">

  <!-- Quirk global styles (for footer consistency across site) -->
  <link rel="stylesheet" href="assets/style.css" />

  <!-- Tailwind (CDN) -->
  <script src="https://cdn.tailwindcss.com"></script>

  <!-- React 18 + Babel (CDN) -->
  <script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>
  <script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

  <style>
    :focus-visible { outline: 2px solid #34d399; outline-offset: 2px; }
  </style>
</head>
<body class="bg-gradient-to-b from-emerald-50 to-slate-50 min-h-screen">
  <div id="root"></div>
```

<!-- Site-wide footer (matches other pages) -->

<footer class="footer">

<small>© 2025 Quirk AI — Sales Kiosk</small>

</footer>

<script type="text/babel">

const { useMemo, useState } = React;

/* =====

PHOTOS (local, with safe fallback)

Place images under /docs/assets/kiosk-photos/ (optional).

===== */

const PHOTO_ROOT = 'assets/kiosk-photos/';

const MODEL_PHOTOS = {

Chevrolet: {

'Silverado 1500': 'chevrolet-silverado-1500.jpg',

'Equinox': 'chevrolet-equinox.jpg',

'Traverse': 'chevrolet-traverse.jpg',

'Trax': 'chevrolet-trax.jpg',

'Trailblazer': 'chevrolet-trailblazer.jpg',

'Malibu': 'chevrolet-malibu.jpg',

'Colorado': 'chevrolet-colorado.jpg',

'Blazer': 'chevrolet-blazer.jpg',

'Tahoe': 'chevrolet-tahoe.jpg'

},

GMC: {

'Sierra 1500': 'gmc-sierra-1500.jpg',

'Terrain': 'gmc-terrain.jpg',

```

    'Acadia':'gmc-acadia.jpg',
    'Canyon':'gmc-canyon.jpg',
    'Yukon':'gmc-yukon.jpg'
  },
  Buick: {
    'Encore GX':'buick-encore-gx.jpg',
    'Envision':'buick-envision.jpg',
    'Envista':'buick-envista.jpg'
  }
};

```

```

const MAKE_DEFAULT = {
  Chevrolet:'chevrolet-default.jpg',
  GMC:'gmc-default.jpg',
  Buick:'buick-default.jpg',
  Used:'used-default.jpg'
};

```

```

function photoFor(make, model) {
  const file =
    (MODEL_PHOTOS[make] && MODEL_PHOTOS[make][model]) ||
    MAKE_DEFAULT[make] || 'placeholder.jpg';
  return `${PHOTO_ROOT}${file}`;
}

```

```

/* =====
OBJECTION PLAYBOOK + HELPER
===== */

```

```

// Inline playbook (you can move to assets/objections.json later)

const OBJECTION_LIB = {
  price_high: {
    triggers: ["price", "too high", "best price", "lower price"],
    probe: "Totally fair. Are you aiming at a monthly target or out-the-door?",
    evidence: "This {year} {make} {model} pencils at ${price} in today's market.",
    resolve: "Two ways to hit your target: {term_alt} months gets ~${payment_alt}/mo, or {alt_vehicle}
saves ~${price_save}.",
  },
  payment_high: {
    triggers: ["payment", "$", "per month"],
    probe: "What monthly target are you trying to hit?",
    evidence: "At {term} months and {apr}% with ${down} down, it's ~${payment}/mo.",
    resolve: "At {term_alt} months you're ~${payment_alt}/mo. Or add ${down_alt} down →
~${payment_with_down}/mo.",
  },
  trade_low: {
    triggers: ["trade", "value", "lowball", "trade value"],
    probe: "Can you share mileage and any packages/damage I should note?",
    evidence: "Similar vehicles appraise between ${range_low} and ${range_high} before recon.",
    resolve: "I can lock it with a quick photo set or a 10-minute drive-through appraisal.",
  },
};

// Fill {tokens} with live numbers/text from context
function renderPlay(obj, ctx) {
  const fill = s => s?.replace(/\{(\w+)\}/g, (_, k) => (ctx[k] ?? ""));
  return {
    probe: fill(obj.probe),
  };
}

```

```

    evidence: fill(obj.evidence),
    resolve: fill(obj.resolve),
  };
}

```

```

function useObjection(triggerText, context) {
  const [card, setCard] = React.useState(null);

  React.useEffect(() => {
    if (!triggerText) return;
    const lower = triggerText.toLowerCase();
    const key = Object.keys(OBJECTION_LIB).find(k =>
      OBJECTION_LIB[k].triggers.some(t => lower.includes(t))
    );
    if (!key) return;
    const msg = renderPlay(OBJECTION_LIB[key], context);
    setCard({ key, ...msg });
  }, [triggerText, context]);

  return { card, dismiss: () => setCard(null) };
}

```

```

function ObjectionCard({ data, onClose, onProceed }) {
  if (!data) return null;
  return (
    <div className="fixed bottom-4 right-4 z-50 max-w-md bg-white border border-slate-200 rounded-
2xl shadow-xl p-4">
      <div className="text-sm text-slate-700 space-y-2">
        {data.probe && <p><strong>Quick question:</strong> {data.probe}</p>}

```

```

    {data.evidence && <p>{data.evidence}</p>}

    {data.resolve && <p><strong>Option:</strong> {data.resolve}</p>}

    <div className="flex gap-2 pt-1">

        <button className="px-3 py-2 rounded-xl border" onClick={onClose}>Dismiss</button>

        {onProceed && <button className="px-3 py-2 rounded-xl bg-emerald-600 text-white"
onClick={onProceed}>Proceed</button>}

    </div>

</div>

</div>

);
}

```

```

/* =====

INVENTORY GENERATION

- 170 NEW (Chevrolet 90 / GMC 55 / Buick 25)

- 30 USED (non-Buick/Chevrolet/GMC brands)

===== */

const MODEL_MAP = {
  Chevrolet:['Silverado
1500','Equinox','Traverse','Trax','Trailblazer','Malibu','Colorado','Blazer','Tahoe'],
  GMC:['Sierra 1500','Terrain','Acadia','Canyon','Yukon'],
  Buick:['Encore GX','Envision','Envista']
};

const COLORS = ['Black','White','Silver','Gray','Blue','Red','Green','Brown'];

function basePrice(make, model) {
  const map = {
    Chevrolet: {
      'Trax':21990,'Trailblazer':24990,'Malibu':24990,'Equinox':28990,

```

```

    'Blazer':34990,'Traverse':36990,'Colorado':33990,'Silverado 1500':38990,'Tahoe':58990
  },
  GMC: { 'Terrain':30990,'Acadia':36990,'Canyon':35990,'Sierra 1500':41990,'Yukon':64990 },
  Buick:{ 'Envista':23990,'Encore GX':27990,'Envision':33990 }
};
return (map[make] && map[make][model]) || 29990;
}

```

```

function vinFor(prefix, idx) {
  return prefix + String(1000000000000000 + idx).slice(0,14);
}

```

```

function generateNewVehicles() {
  const inv = [];
  const add = (make, count, pfx) => {
    const models = MODEL_MAP[make];
    for (let i=0;i<count;i++){
      const model = models[i % models.length];
      const color = COLORS[i % COLORS.length];
      const price = basePrice(make, model) + (i % 7) * 750;
      inv.push({
        vin: vinFor(pfx, i+1000),
        stockType:'New',
        year:2025,
        make, model,
        trim:(['LS','LT','LTZ','RS','Denali','Preferred','Avenir'])[i%7],
        price,
        miles: 12 + (i % 9),
        color,

```

```

        photo: photoFor(make, model),
        features:['CarPlay/Android Auto','Backup Camera','Alloy Wheels','Keyless Entry']
    });
}
};
add('Chevrolet', 90, '1GC');
add('GMC', 55, '1GT');
add('Buick', 25, '1G4');
return inv; // 170
}

```

```

function generateUsedVehicles(count=30){
    const usedMakes = {
        Toyota:['Camry','RAV4','Tacoma'],
        Honda:['Civic','CR-V','Accord'],
        Ford:['Escape','Edge','F-150'],
        Nissan:['Altima','Rogue','Sentra'],
        Tesla:['Model 3','Model Y'],
        Subaru:['Outback','Forester'],
        Hyundai:['Tucson','Santa Fe'],
        Kia:['Sportage','Sorento'],
        Volkswagen:['Tiguan','Jetta'],
        Mazda:['CX-5','Mazda3']
    };
    const makes = Object.keys(usedMakes);
    const inv = [];
    for (let i=0;i<count;i++){
        const make = makes[i % makes.length];
        const models = usedMakes[make];
    }
}

```



```

const model = models[i % models.length];
const year = 2019 + (i % 6); // 2019–2024
const price = 17000 + (i % 14) * 1200;
const miles = 18000 + (i % 25) * 3100;
inv.push({
  vin: vinFor('2UX', i+5000),
  stockType:'Used',
  year, make, model,
  trim:(['Base','S','SE','SEL','Premium'])[i%5],
  price, miles,
  color: COLORS[(i+2) % COLORS.length],
  photo: photoFor('Used','default'),
  features:['CarPlay','Bluetooth','Alloys']
});
}
return inv; // 30
}

```

```

const INVENTORY = [...generateNewVehicles(), ...generateUsedVehicles(30)]; // 200 total

```

```

/* =====
PAYMENT UTILS
===== */
function calcPayment(amount, aprPct, termMonths) {
  const r = aprPct / 100 / 12;
  if (r === 0) return amount / termMonths;
  return (r * amount) / (1 - Math.pow(1 + r, -termMonths));
}

```

```
const STEPS = ['Welcome','Discovery','Vehicle Match','Payments','Trade-In','Credit Start','F&I Options','Review & Handoff'];
```

```
/* =====
```

```
APP
```

```
===== */
```

```
function App() {
```

```
  const [step, setStep] = useState(0);
```

```
  const [consent, setConsent] = useState({ sms:false, email:false, privacy:false });
```

```
  const [profile, setProfile] = useState({
```

```
    name:'', contact:'', useCase:'Commute', bodyStyle:'Sedan', budget:400, timeline:'This week'
```

```
  });
```

```
  const [selectedVin, setSelectedVin] = useState(null);
```

```
  const [desk, setDesk] = useState({ price:0, down:2000, apr:6.9, term:72, taxesFees:995,  
products:{vsc:false,gap:false,maint:false} });
```

```
  const [trade, setTrade] = useState({ vin:'', miles:'', condition:'Average', rangeLow:0, rangeHigh:0 });
```

```
  // === Objections state + context
```

```
  const [objectionTrigger, setObjectionTrigger] = useState("");
```

```
  const vehicle = useMemo(() => INVENTORY.find(v => v.vin === selectedVin) || null, [selectedVin]);
```

```
  const productTotal = useMemo(() => (desk.products.vsc?1895:0) + (desk.products.gap?795:0) +  
(desk.products.maint?495:0), [desk.products]);
```

```
  const amountFinanced = useMemo(() => {
```

```
    const price = vehicle ? vehicle.price : desk.price;
```

```
    const tradeMid = (trade.rangeLow + trade.rangeHigh) / 2 || 0;
```

```
    const gross = price + desk.taxesFees + productTotal - Number(desk.down || 0) - tradeMid;
```

```
    return Math.max(gross, 0);
```

```

    }, [vehicle, desk.down, desk.taxesFees, productTotal, trade.rangeLow, trade.rangeHigh]);

    const monthly = useMemo(() => calcPayment(amountFinanced, desk.apr, desk.term),
    [amountFinanced, desk.apr, desk.term]);

    // Live context passed to objection messages
    const objectionContext = useMemo(() => ({
      year: (vehicle?.year ?? 2025),
      make: (vehicle?.make ?? ""),
      model: (vehicle?.model ?? ""),
      price: (vehicle ? vehicle.price : desk.price).toLocaleString(),
      term: desk.term,
      apr: desk.apr,
      down: Number(desk.down).toLocaleString(),
      payment: Math.round(monthly).toLocaleString(),
      term_alt: desk.term === 72 ? 84 : 72,
      payment_alt: Math.round(calcPayment(amountFinanced, desk.apr, desk.term === 72 ? 84 :
72)).toLocaleString(),
      down_alt: 1000,
      payment_with_down: Math.round(calcPayment(Math.max(amountFinanced-1000,0), desk.apr,
desk.term)).toLocaleString(),
      alt_vehicle: "similar unit with fewer options",
      price_save: "1,500",
      range_low: trade.rangeLow.toLocaleString(),
      range_high: trade.rangeHigh.toLocaleString(),
    }), [vehicle, desk, monthly, amountFinanced, trade]);

    const { card, dismiss } = useObjection(objectionTrigger, objectionContext);

    function next(){ setStep(s => Math.min(s+1, STEPS.length-1)); }
    function back(){ setStep(s => Math.max(s-1, 0)); }

```

```

function estimateTradeRange() {
  const base=12000;
  const miles=parseInt(trade.miles||'0',10);
  const adjMiles=Math.max(0,(60000-miles)*0.05);
  const cond=trade.condition==='Great'?1500:(trade.condition==='Rough'?-1500:0);
  const low=Math.max(500, base+adjMiles+cond-1500);
  const high=Math.max(low+500, base+adjMiles+cond+1500);
  setTrade(t => ({...t, rangeLow:Math.round(low), rangeHigh:Math.round(high)}));

  // Trigger objection helper if range looks low
  if (low < 9000) setObjectTrigger("trade value");
}

function selectVehicle(vin) {
  const v = INVENTORY.find(x => x.vin === vin);
  if (v) {
    setSelectedVin(v.vin);
    setDesk(d => ({...d, price: v.price}));
    setStep(3);
  }
}

return (
  <div className="relative w-full min-h-screen">
    {/* decorative glows */}
    <div aria-hidden className="pointer-events-none absolute inset-0 overflow-hidden">
      <div className="absolute -top-24 -left-24 h-72 w-72 rounded-full bg-emerald-300/30 blur-3xl"></div>

```

```
<div className="absolute -bottom-24 -right-24 h-72 w-72 rounded-full bg-sky-300/20 blur-3xl"></div>
```

```
</div>
```

```
<div className="max-w-6xl mx-auto px-4 sm:px-8 py-6 sm:py-10 relative">
```

```
  { /* Header — Centered */ }
```

```
  <header className="mb-4 sm:mb-6 flex items-center justify-center text-center">
```

```
    <h1 className="text-2xl sm:text-3xl font-extrabold tracking-tight text-emerald-900">
```

```
      Quirk AI Sales Kiosk
```

```
    </h1>
```

```
  </header>
```

```
  { /* Stepper */ }
```

```
  <nav aria-label="Progress" className="mb-5 sm:mb-7">
```

```
    <ol className="grid grid-cols-4 sm:grid-cols-8 gap-2">
```

```
      { STEPS.map((label, i) => (
```

```
        <li key={label} className={`flex items-center gap-2 p-1 rounded-full ${i <= step ? 'bg-emerald-600 text-white': 'bg-white/70 text-slate-600 border'} border-slate-200`} >
```

```
          <span className={`inline-flex h-5 w-5 items-center justify-center rounded-full text-xs font-bold ${i <= step ? 'bg-white/20': 'bg-slate-200 text-slate-700'}`} >{i+1}</span>
```

```
          <span className="hidden sm:block text-xs font-semibold truncate pr-1">{label}</span>
```

```
        </li>
```

```
      )))
```

```
    </ol>
```

```
  </nav>
```

```
  { /* Card */ }
```

```
  <div className="bg-white/90 backdrop-blur rounded-2xl shadow-2xl border border-slate-200 p-4 sm:p-6">
```

```
    { step === 0 && (
```

```

<Welcome
  consent={consent} setConsent={setConsent}
  profile={profile} setProfile={setProfile}
  onNext={next}
  onSelect={selectVehicle}
/>
)}

{step===1 && (
  <Discovery profile={profile} setProfile={setProfile} onNext={next} onBack={back} />
)}

{step===2 && (
  <VehicleMatch selectedVin={selectedVin} onSelect={(vin)=>{ setSelectedVin(vin); next(); }}
onBack={back} />
)}

{step===3 && (
  <Payments
    vehicle={vehicle}
    desk={desk}
    setDesk={setDesk}
    monthly={monthly}
    amountFinanced={amountFinanced}
    onBack={back}
    onNext={next}
    onObjection={setObjectionTrigger}
    budgetTarget={profile.budget}
  />

```

```

    })

    {step===4 && (
        <Trade trade={trade} setTrade={setTrade} estimate={estimateTradeRange} onBack={back}
onNext={next} />
    )}

    {step===5 && <CreditStart onBack={back} onNext={next} />}

    {step===6 && (
        <FandI desk={desk} setDesk={setDesk} monthly={monthly} onBack={back} onNext={next} />
    )}

    {step===7 && (
        <Review profile={profile} vehicle={vehicle} desk={desk} trade={trade} monthly={monthly}
onBack={back} />
    )}
</div>

    {/* Objection helper overlay */}
    <ObjectionCard data={card} onClose={dismiss} />
</div>
</div>
);
}

/* ----- UI helpers ----- */
function SectionTitle({ title, subtitle }) {
    return (

```

```

<div className="mb-4">

  <h2 className="text-xl sm:text-2xl font-bold text-emerald-800 text-center">{title}</h2>

  {subtitle} && <p className="text-center text-slate-600 mt-1">{subtitle}</p>

</div>

);
}

function Field({ label, children }) {
  return (
    <label className="block mb-3">

      <span className="block text-sm font-semibold text-slate-700 mb-1">{label}</span>

      {children}

    </label>

  );
}

function PrimaryButton(props){
  const cls = (props.className || "") + " px-5 py-3 rounded-2xl bg-gradient-to-r from-emerald-600 to-emerald-700 text-white font-semibold shadow hover:brightness-110 active:brightness-95";

  return <button {...props} className={cls} />;
}

/* ----- Welcome + Browse ----- */

function Welcome({ consent, setConsent, profile, setProfile, onNext, onSelect }) {
  const [filters, setFilters] = useState({
    stockType: 'New',
    make: "",
    model: "",
    color: "",
    minPrice: "",
    maxPrice: ""
  });

```



```

});

const [page, setPage] = useState(1);

const perPage = 12;

const modelOptions = useMemo(() => filters.make ? MODEL_MAP[filters.make] : [], [filters.make]);

// Robust filter (case-insensitive) — fixes "Used" still showing "New"
const results = useMemo(() => {
  const min = Number(filters.minPrice || 0);
  const max = Number(filters.maxPrice || 999999);
  return INVENTORY.filter(v => {
    const stockOK = !filters.stockType || v.stockType.toLowerCase() ===
filters.stockType.toLowerCase();

    const makeOK = !filters.make || v.make === filters.make;
    const modelOK = !filters.model || v.model === filters.model;
    const colorOK = !filters.color || v.color === filters.color;
    const priceOK = v.price >= min && v.price <= max;

    return stockOK && makeOK && modelOK && colorOK && priceOK;
  });
}, [filters]);

const totalPages = Math.max(1, Math.ceil(results.length / perPage));
const pageItems = results.slice((page-1)*perPage, page*perPage);

return (
  <div>

    <SectionTitle title="Welcome to Quirk Auto Dealers" subtitle="Start the guided flow or browse our
inventory below." />

```

```

<div className="grid sm:grid-cols-2 gap-4 mb-6">
  <Field label="Your name (optional)">
    <input aria-label="Your name" className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300"
      value={profile.name} onChange={e=>setProfile({...profile, name:e.target.value})}/>
  </Field>
  <Field label="Text or email to send your quote (optional)">
    <input aria-label="Contact" className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300" placeholder="(555) 555-5555 or you@example.com"
      value={profile.contact} onChange={e=>setProfile({...profile, contact:e.target.value})}/>
  </Field>
</div>

```

```

{/* Browse Inventory */}

```

```

<div className="rounded-2xl border border-slate-200 p-4 bg-white/80">
  <h3 className="text-lg font-bold text-emerald-800 mb-2 text-center">Browse Inventory</h3>
  <div className="grid md:grid-cols-6 gap-3">
    <Field label="Stock Type">
      <select className="w-full rounded-xl border p-3" value={filters.stockType}
        onChange={e=>{setFilters({...filters, stockType:e.target.value}); setPage(1);}}>
        <option value="">Any</option>
        <option value="New">New</option>
        <option value="Used">Used</option>
      </select>
    </Field>
    <Field label="Make">
      <select className="w-full rounded-xl border p-3" value={filters.make}
        onChange={e=>{ setFilters(f=>({...f, make:e.target.value, model:" "}); setPage(1); }}>
        <option value="">Any</option>

```

```

    <option>Chevrolet</option>

    <option>GMC</option>

    <option>Buick</option>

  </select>

</Field>

<Field label="Model">

  <select className="w-full rounded-xl border p-3" value={filters.model}

    onChange={e=>{ setFilters({...filters, model:e.target.value}); setPage(1); }}
disabled={!filters.make}>

    <option value="">{filters.make ? 'Any' : 'Choose make first'}</option>

    {modelOptions.map(m => <option key={m}>{m}</option>)}

  </select>

</Field>

<Field label="Color">

  <select className="w-full rounded-xl border p-3" value={filters.color}

    onChange={e=>{ setFilters({...filters, color:e.target.value}); setPage(1); }}>

    <option value="">Any</option>

    {COLORS.map(c => <option key={c}>{c}</option>)}

  </select>

</Field>

<Field label="Min Price ($)">

  <input type="number" className="w-full rounded-xl border p-3" value={filters.minPrice}

    onChange={e=>{ setFilters({...filters, minPrice:e.target.value}); setPage(1); }}/>

</Field>

<Field label="Max Price ($)">

  <input type="number" className="w-full rounded-xl border p-3" value={filters.maxPrice}

    onChange={e=>{ setFilters({...filters, maxPrice:e.target.value}); setPage(1); }}/>

</Field>

</div>

```

```
<div className="mt-3 text-sm text-slate-600">{results.length.toLocaleString()} vehicles  
found</div>
```

```
<div className="mt-3 grid md:grid-cols-3 gap-4">  
  {pageItems.map(v => (  
    <button key={v.vin} onClick={()=>onSelect(v.vin)}  
      className="group text-left rounded-2xl border border-slate-200 shadow-sm  
      hover:shadow-lg hover:-translate-y-0.5 transition p-3 bg-white">  
        <div className="relative">  
          <img  
            src={v.photo}  
            alt={` ${v.make} ${v.model}`}  
            className="w-full h-40 object-cover rounded-xl"  
            loading="lazy"  
            onError={(e)=>{ e.currentTarget.onerror=null; e.currentTarget.src = PHOTO_ROOT +  
              'placeholder.jpg'; }}  
          />  
          <span className="absolute bottom-2 left-2 text-xs px-2 py-1 rounded-full bg-black/60 text-  
            white">{v.stockType}</span>  
        </div>  
        <div className="mt-2 font-semibold">{v.year} {v.make} {v.model}</div>  
        <div className="text-emerald-700 font-extrabold text-lg">${v.price.toLocaleString()}</div>  
        <div className="text-xs text-slate-600">{v.color} • {v.trim}</div>  
      </button>  
    )})  
</div>  
  
{/* Pagination */}  
  
<div className="mt-3 flex items-center justify-between">
```

```

        <button className="px-3 py-2 rounded-xl border bg-white disabled:opacity-50"
            onClick={()=>setPage(p=>Math.max(1,p-1))} disabled={page<=1}>< Prev</button>

        <div className="text-sm text-slate-600">Page {page} / {totalPages}</div>

        <button className="px-3 py-2 rounded-xl border bg-white disabled:opacity-50"
            onClick={()=>setPage(p=>Math.min(totalPages,p+1))} disabled={page>=totalPages}>Next
    →</button>

    </div>

</div>

<div className="mt-4 flex flex-wrap gap-3 justify-end">
    <PrimaryButton onClick={onNext} aria-label="Start guided flow">Start Guided Flow
→</PrimaryButton>

</div>

<details className="mt-3 text-sm">
    <summary className="cursor-pointer text-slate-600">Consent & privacy</summary>

    <div className="grid sm:grid-cols-3 gap-3 mt-2 text-slate-700">

        <label className="flex items-center gap-2"><input type="checkbox" checked={consent.sms}
onChange={e=>setConsent({...consent, sms:e.target.checked})}/> SMS updates</label>

        <label className="flex items-center gap-2"><input type="checkbox" checked={consent.email}
onChange={e=>setConsent({...consent, email:e.target.checked})}/> Email updates</label>

        <label className="flex items-center gap-2"><input type="checkbox" checked={consent.privacy}
onChange={e=>setConsent({...consent, privacy:e.target.checked})}/> I agree to privacy policy</label>

    </div>

</details>

</div>

);
}

/* ----- Discovery / Matches / Deal ----- */

```

```

function Discovery({ profile, setProfile, onNext, onBack }) {

  return (

    <div>

      <SectionTitle title="Tell us what you need" subtitle="A few quick choices so we can tailor the
match." />

      <div className="grid sm:grid-cols-2 gap-4">

        <Field label="Primary use">

          <select className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300"
value={profile.useCase} onChange={e=>setProfile({...profile, useCase:e.target.value})}>

            {[ 'Commute', 'Family', 'Adventure', 'Business' ].map(x => <option key={x}>{x}</option>)}

          </select>

        </Field>

        <Field label="Preferred body style">

          <select className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300"
value={profile.bodyStyle} onChange={e=>setProfile({...profile, bodyStyle:e.target.value})}>

            {[ 'Sedan', 'SUV', 'Truck', 'EV' ].map(x => <option key={x}>{x}</option>)}

          </select>

        </Field>

        <Field label="Monthly budget target">

          <input type="number" className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-
emerald-300" value={profile.budget} onChange={e=>setProfile({...profile,
budget:Number(e.target.value)})}/>

        </Field>

        <Field label="Timeline">

          <select className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300"
value={profile.timeline} onChange={e=>setProfile({...profile, timeline:e.target.value})}>

            {[ 'Today', 'This week', 'This month', 'Just researching' ].map(x => <option key={x}>{x}</option>)}

          </select>

        </Field>

      </div>

      <div className="mt-4 flex justify-between">

```

```
      <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={onBack}>←  
Back</button>
```

```
      <PrimaryButton onClick={onNext}>See matches →</PrimaryButton>
```

```
    </div>
```

```
  </div>
```

```
);
```

```
}
```

```
function VehicleMatch({ selectedVin, onSelect, onBack }) {
```

```
  const picks = INVENTORY.slice(0, 9);
```

```
  return (
```

```
    <div>
```

```
      <SectionTitle title="Best matches" subtitle="Tap a vehicle to continue." />
```

```
      <div className="grid md:grid-cols-3 gap-4">
```

```
        {picks.map(v => (
```

```
          <button key={v.vin} className={`group text-left rounded-2xl border border-slate-200 shadow-  
sm hover:shadow-lg hover:-translate-y-0.5 transition p-3 bg-white/90 ${selectedVin===v.vin?'ring-4  
ring-emerald-300':''}`} onClick={()=>onSelect(v.vin)}>
```

```
            <div className="relative">
```

```
              <img
```

```
                src={v.photo}
```

```
                alt={` ${v.make} ${v.model}`}
```

```
                className="w-full h-44 object-cover rounded-xl"
```

```
                loading="lazy"
```

```
                onError={(e)=>{ e.currentTarget.onerror=null; e.currentTarget.src = PHOTO_ROOT +  
'placeholder.jpg'; }}
```

```
              />
```

```
              <span className="absolute bottom-2 left-2 text-xs px-2 py-1 rounded-full bg-black/60 text-  
white">{v.miles.toLocaleString()} mi</span>
```

```
            </div>
```

```

    <div className="mt-2 font-semibold">{v.year} {v.make} {v.model}</div>

    <div className="text-emerald-700 font-extrabold text-lg">${v.price.toLocaleString()}</div>

    <div className="text-xs text-slate-600">{v.color} • {v.trim}</div>

  </button>

  )}}

</div>

<div className="mt-4 flex justify-between">

  <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={onBack}><←
Back</button>

  <span className="text-sm text-slate-500">Choose a vehicle to continue</span>

</div>

</div>

);

}

```

```

function Payments({ vehicle, desk, setDesk, monthly, amountFinanced, onBack, onNext, onObjection,
budgetTarget }) {

```

```

  // Auto-fire objection if payment is well above customer's budget

```

```

  React.useEffect(() => {

```

```

    if (!budgetTarget) return;

```

```

    const overBy = monthly - Number(budgetTarget);

```

```

    if (overBy > 50) onObjection && onObjection("payment too high"); // matches payment_high

```

```

  }, [monthly, budgetTarget, onObjection]);

```

```

const price = vehicle ? vehicle.price : desk.price;

```

```

return (

```

```

  <div>

```

```

    <SectionTitle title="Build your payment" subtitle="Deterministic calculator; real taxes/fees plug in
here." />

```

```

    <div className="grid md:grid-cols-2 gap-6">

```



```

<div>

  <div className="rounded-xl border p-4 bg-white/90">

    <div className="text-sm text-slate-600 mb-2">Vehicle price</div>

    <div className="text-2xl font-bold">${price.toLocaleString()}</div>

    <div className="grid grid-cols-2 gap-3 mt-4">

      <Field label="Down payment">

        <input type="number" className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300" value={desk.down} onChange={e=>setDesk({...desk, down:Number(e.target.value)}} />

      </Field>

      <Field label="Taxes & fees (est.)">

        <input type="number" className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300" value={desk.taxesFees} onChange={e=>setDesk({...desk, taxesFees:Number(e.target.value)}} />

      </Field>

      <Field label="APR %">

        <input type="number" className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300" value={desk.apr} step="0.1" onChange={e=>setDesk({...desk, apr:Number(e.target.value)}} />

      </Field>

      <Field label="Term (months)">

        <select className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300" value={desk.term} onChange={e=>setDesk({...desk, term:Number(e.target.value)}}>

          {[36,48,60,72,84].map(n => <option key={n} value={n}>{n}</option>)}

        </select>

      </Field>

    </div>

  <details className="mt-2">

    <summary className="text-slate-600 cursor-pointer">Products</summary>

    <div className="grid sm:grid-cols-3 gap-3 mt-2 text-slate-700">

      <label className="flex items-center gap-2"><input type="checkbox" checked={desk.products.vsc} onChange={e=>setDesk({...desk, products:{...desk.products, vsc:e.target.checked}})} /> VSC ($1,895)</label>

```

```
      <label className="flex items-center gap-2"><input type="checkbox"
checked={desk.products.gap} onChange={e=>setDesk({...desk, products:{...desk.products,
gap:e.target.checked}})} /> GAP ($795)</label>
```

```
      <label className="flex items-center gap-2"><input type="checkbox"
checked={desk.products.maint} onChange={e=>setDesk({...desk, products:{...desk.products,
maint:e.target.checked}})} /> Maintenance ($495)</label>
```

```
    </div>
```

```
  </details>
```

```
</div>
```

```
</div>
```

```
<div className="rounded-2xl border p-4 bg-emerald-50/80">
```

```
  <div className="text-sm text-emerald-700">Estimated monthly</div>
```

```
  <div className="text-4xl font-extrabold text-emerald-800">${monthly.toFixed(0)}</div>
```

```
  <div className="text-sm text-slate-600 mt-2">Amount financed:
  ${amountFinanced.toLocaleString()}</div>
```

```
  <div className="mt-3 flex flex-wrap gap-2 text-xs text-slate-700">
```

```
    <span className="px-2 py-1 rounded-full bg-white border">APR {desk.apr}%</span>
```

```
    <span className="px-2 py-1 rounded-full bg-white border">{desk.term} mos</span>
```

```
    <span className="px-2 py-1 rounded-full bg-white border">Down
    ${Number(desk.down).toLocaleString()}</span>
```

```
  </div>
```

```
  <p className="text-xs text-slate-500 mt-2">Estimates only; subject to lender approval, credit
tier, LTV, and program rules. Your final payment may vary.</p>
```

```
</div>
```

```
</div>
```

```
<div className="mt-4 flex justify-between">
```

```
  <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={onBack}>←
Back</button>
```

```
  <PrimaryButton onClick={onNext}>Add trade →</PrimaryButton>
```

```
</div>
```

```
</div>
```

```
);
```

```
}
```

```
function Trade({ trade, setTrade, estimate, onBack, onNext }) {  
  return (  
    <div>  
      <SectionTitle title="Trade-in pre-valuation" subtitle="Get a market-based range. Final number  
after in-person inspection." />  
      <div className="grid md:grid-cols-3 gap-4">  
        <Field label="VIN">  
          <input className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300"  
value={trade.vin} onChange={e=>setTrade({...trade, vin:e.target.value})} placeholder="Scan or type VIN"  
/>  
        </Field>  
        <Field label="Mileage">  
          <input className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300"  
value={trade.miles} onChange={e=>setTrade({...trade, miles:e.target.value})} placeholder="e.g., 45,500"  
/>  
        </Field>  
        <Field label="Condition">  
          <select className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-emerald-300"  
value={trade.condition} onChange={e=>setTrade({...trade, condition:e.target.value})}>  
            {[ 'Great', 'Average', 'Rough' ].map(x => <option key={x}>{x}</option>)}  
          </select>  
        </Field>  
      </div>  
      <div className="mt-2 flex gap-2">  
        <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={estimate}>Estimate  
range</button>  
        {trade.rangeLow>0 && (  
          <div className="px-4 py-2 rounded-xl bg-emerald-50 border border-emerald-200 text-emerald-  
800">  
            Estimated range: ${trade.rangeLow.toLocaleString()} – ${trade.rangeHigh.toLocaleString()}
```

```

        </div>

    })

</div>

<div className="mt-4 flex justify-between">

    <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={onBack}>←
Back</button>

    <PrimaryButton onClick={onNext}>Start credit →</PrimaryButton>

</div>

</div>

);
}

function CreditStart({ onBack, onNext }) {

    return (

        <div>

            <SectionTitle title="Start your credit application" subtitle="Soft pull first. No impact to score until
you consent to submit." />

            <div className="grid md:grid-cols-3 gap-4">

                <Field label="Full name"><input className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-
emerald-300" /></Field>

                <Field label="Email"><input className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-
emerald-300" /></Field>

                <Field label="Mobile"><input className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-
emerald-300" /></Field>

                <Field label="ZIP"><input className="w-full rounded-xl border p-3 focus:ring-2 focus:ring-
emerald-300" /></Field>

                <Field label="Income (monthly)"><input className="w-full rounded-xl border p-3 focus:ring-2
focus:ring-emerald-300" /></Field>

                <Field label="Consent to soft pull">

                    <label className="flex items-center gap-2"><input type="checkbox" /> I agree to a soft
inquiry</label>

                </Field>

```

```

    </div>

    <div className="mt-4 flex justify-between">

      <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={onBack}><←
Back</button>

      <PrimaryButton onClick={onNext}>Preview protection →</PrimaryButton>

    </div>

  </div>

);
}

function FandI({ desk, setDesk, monthly, onBack, onNext }) {
  return (
    <div>

      <SectionTitle title="Protect your purchase" subtitle="Add options and see the impact instantly." />

      <div className="grid sm:grid-cols-3 gap-4">

        <label className="rounded-2xl border p-4 flex items-start gap-3 bg-white/90">

          <input type="checkbox" checked={desk.products.vsc} onChange={e=>setDesk({...desk,
products:{...desk.products, vsc:e.target.checked}})} />

          <div>

            <div className="font-semibold">Vehicle Service Contract</div>

            <div className="text-sm text-slate-600">Covers major repairs beyond factory warranty.
+$1,895</div>

          </div>

        </label>

        <label className="rounded-2xl border p-4 flex items-start gap-3 bg-white/90">

          <input type="checkbox" checked={desk.products.gap} onChange={e=>setDesk({...desk,
products:{...desk.products, gap:e.target.checked}})} />

          <div>

            <div className="font-semibold">GAP</div>

            <div className="text-sm text-slate-600">Protects against total loss payoff gap. +$795</div>

```

```

    </div>
  </label>

  <label className="rounded-2xl border p-4 flex items-start gap-3 bg-white/90">
    <input type="checkbox" checked={desk.products.maint} onChange={e=>setDesk({...desk,
products:{...desk.products, maint:e.target.checked}})} />

    <div>
      <div className="font-semibold">Maintenance</div>
      <div className="text-sm text-slate-600">Prepaid maintenance services. +$495</div>
    </div>
  </label>
</div>

<div className="mt-3 p-3 rounded-xl bg-emerald-50 border border-emerald-200 text-emerald-
800">
  Current estimated payment: <strong>${monthly.toFixed(0)}</strong>
</div>

<div className="mt-4 flex justify-between">
  <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={onBack}>←
Back</button>

  <PrimaryButton onClick={onNext}>Review & handoff →</PrimaryButton>
</div>
</div>
);
}

function Review({ profile, vehicle, desk, trade, monthly, onBack }) {
  return (
    <div>
      <SectionTitle title="Review & handoff" subtitle="We'll prep your buyer's order and connect you to
a manager or F&I." />

      <div className="grid md:grid-cols-2 gap-4">

```

```

<div className="rounded-2xl border p-4 bg-white/90">
  <div className="font-semibold mb-2">Your picks</div>
  <ul className="text-sm text-slate-700 space-y-1">
    <li><strong>Name:</strong> {profile.name || '—'}</li>
    <li><strong>Contact:</strong> {profile.contact || '—'}</li>
    <li><strong>Use:</strong> {profile.useCase} • <strong>Body style:</strong>
{profile.bodyStyle}</li>
    <li><strong>Budget target:</strong> ${profile.budget}/mo</li>
    <li><strong>Vehicle:</strong> {vehicle ? `${vehicle.year} ${vehicle.make} ${vehicle.model}` : '—
'}</li>
    <li><strong>Trade range:</strong> {trade.rangeLow ? `${trade.rangeLow.toLocaleString()}–
${trade.rangeHigh.toLocaleString()}` : '—'}</li>
  </ul>
</div>

<div className="rounded-2xl border p-4 bg-white/90">
  <div className="font-semibold mb-2">Deal preview</div>
  <ul className="text-sm text-slate-700 space-y-1">
    <li><strong>Est. monthly:</strong> ${monthly.toFixed(0)}</li>
    <li><strong>Down:</strong> ${Number(desk.down).toLocaleString()}</li>
    <li><strong>APR/Term:</strong> {desk.apr}% • {desk.term} mos</li>
    <li><strong>Taxes & fees:</strong> ${desk.taxesFees.toLocaleString()}</li>
    <li><strong>Products:</strong> {(desk.products.vsc?'VSC ':'') + (desk.products.gap?'GAP ':'') +
(desk.products.maint?'Maint ':'') || '—'}</li>
  </ul>
  <p className="text-xs text-slate-500 mt-2">This is a draft for review, not a binding offer. Final
numbers subject to inspection and lender approval.</p>
</div>
</div>

<div className="mt-4 flex flex-wrap gap-3 justify-between">
  <button className="px-4 py-2 rounded-xl border bg-white/80" onClick={onBack}>←
Back</button>

```

```
    <div className="flex gap-2">
      <button className="px-4 py-2 rounded-xl border bg-white/90">Raise hand for
manager</button>
      <PrimaryButton>Text me this summary</PrimaryButton>
      <button className="px-4 py-2 rounded-2xl bg-emerald-700 text-white font-semibold">I'm
ready now</button>
    </div>
  </div>
</div>
);
}
```

```
ReactDOM.createRoot(document.getElementById('root')).render(<App />);
</script>
</body>
</html>
```