

```

import React, { useState, useEffect } from 'react';
import { Camera, Mic, Save, CheckCircle, Circle, ChevronRight, X } from 'lucide-react';

export default function INAFormInterface() {
  const [activeSection, setActiveSection] = useState(1);
  const [lastSaved, setLastSaved] = useState(new Date());
  const [bodyMapMarks, setBodyMapMarks] = useState([]);
  const [bodyMapGender, setBodyMapGender] = useState('female');
  const [painScale, setPainScale] = useState(null);
  const [gcsScores, setGcsScores] = useState({ eye: null, verbal: null, motor: null });
  const [photos, setPhotos] = useState([]);
  const [customADLs, setCustomADLs] = useState([]);
  const [showBodyMap, setShowBodyMap] = useState(false);

  useEffect(() => {
    const interval = setInterval(() => {
      setLastSaved(new Date());
    }, 30000);
    return () => clearInterval(interval);
  }, []);

  const sections = [
    { id: 1, title: 'Referrer Details', complete: true },
    { id: 2, title: 'Client Details', complete: true },
    { id: 3, title: 'Background Information', complete: false },
    { id: 4, title: 'Mental Capacity Assessment', complete: false },
    { id: 5, title: 'Consent Verification', complete: false },
    { id: 6, title: 'Employment & Education', complete: false },
    { id: 7, title: 'Clinical Assessments', complete: false },
    { id: 8, title: 'Physical Measurements', complete: false },
    { id: 9, title: 'Cognitive & Sensory', complete: false },
    { id: 10, title: 'Mobility & ADLs', complete: false },
    { id: 11, title: 'Goals & Outcomes', complete: false },
    { id: 12, title: 'Environment & Social', complete: false },
    { id: 13, title: 'Client Values', complete: false },
    { id: 14, title: 'Emotional Wellbeing', complete: false },
    { id: 15, title: 'Sign-Off', complete: false },
  ];
}

const adlCategories = [
  'Maintaining a Safe Environment',
  'Breathing',
  'Controlling Body Temperature',
  'Communication',
  'Cognition',
  'Eating and Drinking',
  'Cooking/Meal Preparation',
  'Housekeeping',
  'Managing Medications',
  'Managing Finances',
  'Eliminating (Toileting & Continence)',
  'Personal Hygiene',
  'Dressing',
  'Mobilising (Indoors)',
  'Mobilising (Outdoors)',
  'Working and Playing',
  'Expressing Sexuality',
  'Sleeping',
  'Dying (where applicable)',
];

```

```

const handleBodyMapClick = (e) => {
  const rect = e.currentTarget.getBoundingClientRect();
  const x = ((e.clientX - rect.left) / rect.width) * 100;
  const y = ((e.clientY - rect.top) / rect.height) * 100;
  setBodyMapMarks([...bodyMapMarks, { x, y, severity: 'moderate', id: Date.now() }]);
};

const calculateGCS = () => {
  if (!gcsScores.eye || !gcsScores.verbal || !gcsScores.motor) return '-';
  return gcsScores.eye + gcsScores.verbal + gcsScores.motor;
};

const addPhoto = () => {
  setPhotos([...photos, { id: Date.now(), status: 'pending', description: '' }]);
};

const addCustomADL = () => {
  setCustomADLs([...customADLs, { category: '', independent: null, support: null, details: '' }]);
};

const BodyMapModal = () => {
  return (
    <div className="fixed inset-0 bg-black bg-opacity-75 flex items-center justify-center z-50 p-4">
      <div className="bg-white rounded-lg max-w-4xl w-full max-h-[90vh] overflow-y-auto">
        <div className="p-4 border-b border-gray-200 flex justify-between items-center sticky top-0 bg-white z-10">
          <h3 className="text-lg font-semibold">Interactive Body Map</h3>
          <button onClick={() => setShowBodyMap(false)} className="p-2 hover:bg-gray-100 rounded">
            <X size={20} />
          </button>
        </div>

        <div className="p-6">
          <div className="flex items-center justify-between mb-4">
            <p className="text-sm text-gray-600">
              Click on the body diagram to mark areas of pain, injury, or concern.
            </p>
            <div className="flex gap-2">
              <button
                onClick={() => setBodyMapGender('female')}
                className={`${px-4 py-2 rounded text-sm font-medium transition-all ${bodyMapGender === 'female' ? 'bg-pink-100 text-pink-700 border-2 border-pink-300' : 'bg-gray-100 text-gray-600 border border-gray-300 hover:bg-gray-200'}`}
              >
                Female
              </button>
              <button
                onClick={() => setBodyMapGender('male')}
                className={`${px-4 py-2 rounded text-sm font-medium transition-all ${bodyMapGender === 'male' ? 'bg-blue-100 text-blue-700 border-2 border-blue-300' : 'bg-gray-100 text-gray-600 border border-gray-300 hover:bg-gray-200'}`}
              >
                Male
              </button>
            </div>
          </div>
        </div>
      </div>
    </div>
  );
};

```

```

        </button>
    </div>
</div>

<div className="mb-4 p-3 bg-yellow-50 border border-yellow-200 rounded-lg">
    <div className="text-xs font-semibold text-yellow-800 mb-1">⚠️ Temporary Placeholder</div>
    <div className="text-xs text-yellow-700">
        Replace with your body map image URL: <code className="bg-yellow-100 px-1 rounded">/images/{bodyMapGender}-body-map.jpg</code>
    </div>
</div>

<div className="border-2 border-gray-300 rounded-lg overflow-hidden bg-white">
    <div
        className="relative cursor-crosshair bg-gray-50"
        onClick={handleBodyMapClick}
        style={{ minHeight: '500px', position: 'relative' }}
    >
        <div className="absolute inset-0 flex items-center justify-center">
            <div className="text-center p-8">
                <div className="text-6xl mb-4">∅</div>
                <div className="text-lg font-semibold text-gray-700 mb-2">
                    Professional Body Map - {bodyMapGender === 'female' ? 'Female' : 'Male'}
                </div>
                <div className="text-sm text-gray-600 mb-4">
                    Your anatomical diagram will display here
                </div>
                <div className="text-xs text-gray-500 bg-gray-100 rounded p-3 max-w-md mx-auto">
                    Click anywhere to test marking functionality
                </div>
            </div>
        </div>
    </div>

    <svg className="absolute inset-0 w-full h-full pointer-events-none">
        {bodyMapMarks.map((mark) => (
            <g key={mark.id}>
                <circle
                    cx={`${mark.x}%`}
                    cy={`${mark.y}%`}
                    r="12"
                    fill={mark.severity === 'mild' ? '#fbbf24' : mark.severity === 'moderate' ? '#f97316' : '#dc2626'}
                    stroke="white"
                    strokeWidth="3"
                />
                <text
                    x={`${mark.x}%`}
                    y={`${mark.y}%`}
                    textAnchor="middle"
                    dominantBaseline="middle"
                    fill="white"
                    fontSize="10"
                    fontWeight="bold"
                >
                    !
                </text>
            </g>
        ))
    </svg>
</div>

```

```

        ))}
      </svg>
    </div>
  </div>

<div className="mt-4 p-3 bg-blue-50 border border-blue-200 rounded-lg">
  <div className="text-sm font-semibold text-blue-800 mb-2">📍 How to Use:</div>
  <ul className="text-xs text-blue-700 space-y-1 ml-4 list-disc">
    <li>Click anywhere on the body diagram above to mark pain/injury locations</li>
    <li>Each mark appears as a colored dot with an exclamation mark</li>
    <li>Change severity in the list below</li>
  </ul>
</div>

<div className="mt-4 p-3 bg-gray-50 rounded-lg border border-gray-200">
  <div className="text-sm font-semibold mb-2 text-gray-700">Severity Legend:</div>
  <div className="flex gap-6">
    <div className="flex items-center gap-2">
      <div className="w-5 h-5 rounded-full bg-yellow-400 border-2 border-white shadow"></div>
      <span className="text-sm">Mild</span>
    </div>
    <div className="flex items-center gap-2">
      <div className="w-5 h-5 rounded-full bg-orange-500 border-2 border-white shadow"></div>
      <span className="text-sm">Moderate</span>
    </div>
    <div className="flex items-center gap-2">
      <div className="w-5 h-5 rounded-full bg-red-600 border-2 border-white shadow"></div>
      <span className="text-sm">Severe</span>
    </div>
  </div>
</div>

{bodyMapMarks.length > 0 && (
  <div className="mt-4">
    <div className="text-sm font-semibold mb-2">Marked Areas</div>
    <div className="space-y-2 max-h-40 overflow-y-auto">
      {bodyMapMarks.map((mark, idx) => (
        <div key={mark.id} className="flex items-center gap-2 text-sm bg-gray-50 p-3 rounded border border-gray-200">
          <span className="font-medium text-gray-700">Mark {idx + 1}:</span>
          <select
            className="border border-gray-300 rounded px-3 py-1 text-sm flex-1"
            value={mark.severity}
            onChange={(e) => {
              const updated = bodyMapMarks.map(m =>
                m.id === mark.id ? { ...m, severity: e.target.value } : m
              );
              setBodyMapMarks(updated);
            }}
          >
            <option value="mild">Mild</option>
            <option value="moderate">Moderate</option>
            <option value="severe">Severe</option>
          </select>
          <button
            onClick={() => setBodyMapMarks(bodyMapMarks.filter(m => m.id !== mark.id))}>
        
```

```

        className="ml-auto text-red-600 hover:text-red-700 font-medium text-sm"
      >
      Remove
    </button>
  </div>
)}
</div>
</div>
)}

<div className="mt-6 flex justify-end gap-2">
  <button
    onClick={() => setShowBodyMap(false)}
    className="px-6 py-2.5 bg-blue-600 text-white rounded-lg hover:bg-blue-700 font-medium transition-colors"
  >
    Save & Close
  </button>
</div>
</div>
</div>
</div>
);
};

return (
  <div className="min-h-screen bg-gray-100 flex flex-col">
    <div className="bg-white border-b border-gray-200 shadow-sm">
      <div className="max-w-7xl mx-auto px-4 py-3">
        <div className="flex items-center justify-between">
          <div>
            <h1 className="text-xl font-bold text-gray-900">Initial Needs Assessment (INA) Form</h1>
            <div>
              <div>INA-2024-001 • Sarah Johnson • DOB: 15/03/1978</div>
            </div>
            <div>
              <div><Save size={14} className="inline mr-1" /> Last saved: {lastSaved.toLocaleTimeString()}</div>
            </div>
            <div>
              <div>Auto-save active</div>
            </div>
          </div>
        </div>
      </div>
    </div>
    <div className="flex-1 flex max-w-7xl mx-auto w-full">
      <div className="w-64 bg-white border-r border-gray-200 overflow-y-auto">
        <div>
          <div>SECTIONS</div>
          {sections.map((section) => (
            <button
              key={section.id}
              onClick={() => setActiveSection(section.id)}>

```

```

    className={'w-full flex items-center gap-2 px-3 py-2 rounded mb-1 text-left transition-
colors ${(
      activeSection === section.id
      ? 'bg-blue-100 text-blue-900'
      : 'hover:bg-gray-50 text-gray-700'
    )}}
  >
  {section.complete ? (
    <CheckCircle size={16} className="text-green-600 flex-shrink-0" />
  ) : (
    <Circle size={16} className="text-gray-400 flex-shrink-0" />
  )}
  <span className="text-sm flex-1">{section.title}</span>
  {activeSection === section.id && <ChevronRight size={16} />}
</button>
)})}
</div>

<div className="p-4 border-t border-gray-200">
  <div className="text-xs font-semibold text-gray-500 mb-2">COMPLETION</div>
  <div className="flex items-center gap-2">
    <div className="flex-1 bg-gray-200 rounded-full h-2">
      <div className="bg-green-600 h-2 rounded-full" style={{ width: '13%' }}></div>
    </div>
    <span className="text-xs text-gray-600">2/15</span>
  </div>
</div>
</div>

<div className="flex-1 overflow-y-auto p-6">
  {activeSection === 7 && (
    <div className="space-y-6">
      <div className="bg-white rounded-lg border border-gray-200 p-6">
        <h2 className="text-lg font-bold text-gray-900 mb-4">Section 7: Clinical
Assessments</h2>
        <div className="mb-6 pb-6 border-b border-gray-200">
          <h3 className="text-md font-semibold text-gray-800 mb-3">Pain Assessment</h3>
          <div className="grid grid-cols-4 gap-3">
            [
              { value: 0, label: 'No Pain', emoji: '😊' },
              { value: 1, label: 'Mild', emoji: '😐' },
              { value: 2, label: 'Moderate', emoji: '😖' },
              { value: 3, label: 'Severe', emoji: '😣' }
            ].map((scale) => (
              <button
                key={scale.value}
                onClick={() => setPainScale(scale.value)}
                className={`p-4 rounded-lg border-2 transition-all ${(
                  painScale === scale.value
                  ? 'border-blue-500 bg-blue-50'
                  : 'border-gray-200 hover:border-gray-300'
                )}`}
              >
                <div className="text-3xl mb-2">{scale.emoji}</div>
                <div className="text-sm font-medium">{scale.value}</div>
                <div className="text-xs text-gray-600">{scale.label}</div>
              </button>
            ))
          </div>
        </div>
      </div>
    </div>
  )
  </div>
</div>

```

```
        )}
    </div>
</div>

<div className="mb-6 pb-6 border-b border-gray-200">
    <h3 className="text-md font-semibold text-gray-800 mb-3">Body Map</h3>
    <button
        onClick={() => setShowBodyMap(true)}
        className="w-full bg-blue-50 border-2 border-blue-200 rounded-lg p-6 hover:bg-blue-100 transition-colors">
        >
            <div className="text-blue-700 font-medium mb-2">Open Interactive Body Map</div>
            <div className="text-sm text-blue-600">
                {bodyMapMarks.length} area(s) marked
            </div>
        </button>
    </div>

<div className="mb-6 pb-6 border-b border-gray-200">
    <h3 className="text-md font-semibold text-gray-800 mb-3">Vital Signs</h3>
    <div className="grid grid-cols-2 md:grid-cols-3 gap-4">
        <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">Heart Rate (bpm)</label>
            <input type="number" className="w-full border border-gray-300 rounded px-3 py-2" placeholder="72" />
        </div>
        <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">Resp Rate (breaths/min)</label>
            <input type="number" className="w-full border border-gray-300 rounded px-3 py-2" placeholder="16" />
        </div>
        <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">O2 Saturation (%)</label>
            <input type="number" className="w-full border border-gray-300 rounded px-3 py-2" placeholder="98" />
        </div>
        <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">Temperature (°C)</label>
            <input type="number" step="0.1" className="w-full border border-gray-300 rounded px-3 py-2" placeholder="36.5" />
        </div>
        <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">Blood Pressure</label>
            <input type="text" className="w-full border border-gray-300 rounded px-3 py-2" placeholder="120/80" />
        </div>
        <div>
            <label className="block text-sm font-medium text-gray-700 mb-1">NEWS Score</label>
            <input type="number" className="w-full border border-gray-300 rounded px-3 py-2" placeholder="0" />
        </div>
    </div>
</div>
```

```

<div className="mb-6">
  <h3 className="text-md font-semibold text-gray-800 mb-3">Glasgow Coma Scale
  (GCS)</h3>
  <div className="grid grid-cols-1 md:grid-cols-3 gap-4">
    <div>
      <label className="block text-sm font-medium text-gray-700 mb-2">Eye Response
      (E)</label>
      <select
        className="w-full border border-gray-300 rounded px-3 py-2"
        value={gcsScores.eye || ''}
        onChange={(e) => setGcsScores({ ...gcsScores, eye: parseInt(e.target.value) })}>
        >
          <option value="">Select...</option>
          <option value="4">4 - Opens spontaneously</option>
          <option value="3">3 - Opens to voice</option>
          <option value="2">2 - Opens to pain</option>
          <option value="1">1 - No response</option>
        </select>
    </div>
    <div>
      <label className="block text-sm font-medium text-gray-700 mb-2">Verbal
      Response (V)</label>
      <select
        className="w-full border border-gray-300 rounded px-3 py-2"
        value={gcsScores.verbal || ''}
        onChange={(e) => setGcsScores({ ...gcsScores, verbal: parseInt(e.target.value) })}>
        >
          <option value="">Select...</option>
          <option value="5">5 - Oriented</option>
          <option value="4">4 - Confused</option>
          <option value="3">3 - Inappropriate words</option>
          <option value="2">2 - Incomprehensible sounds</option>
          <option value="1">1 - No response</option>
        </select>
    </div>
    <div>
      <label className="block text-sm font-medium text-gray-700 mb-2">Motor
      Response (M)</label>
      <select
        className="w-full border border-gray-300 rounded px-3 py-2"
        value={gcsScores.motor || ''}
        onChange={(e) => setGcsScores({ ...gcsScores, motor: parseInt(e.target.value) })}>
        >
          <option value="">Select...</option>
          <option value="6">6 - Obeys commands</option>
          <option value="5">5 - Localises pain</option>
          <option value="4">4 - Withdraws from pain</option>
          <option value="3">3 - Abnormal flexion</option>
          <option value="2">2 - Abnormal extension</option>
          <option value="1">1 - No response</option>
        </select>
    </div>
  </div>
  <div className="mt-4 p-4 bg-gray-50 rounded">
    <div className="text-sm font-medium text-gray-700">
      Total GCS Score: <span className="text-2xl font-bold text-blue-600
      ml-2">{calculateGCS()}</span> / 15
    </div>
  </div>

```

```

        </div>
        </div>

        <div className="bg-white rounded-lg border border-gray-200 p-6">
          <div className="flex items-center justify-between mb-4">
            <h3 className="text-md font-semibold text-gray-800">Photos & Media</h3>
            <button
              onClick={addPhoto}
              className="flex items-center gap-2 px-3 py-1.5 bg-blue-600 text-white rounded
text-sm hover:bg-blue-700"
            >
              <Camera size={16} />
              Add Photo
            </button>
          </div>
          <div className="grid grid-cols-3 gap-3">
            {photos.map((photo) => (
              <div key={photo.id} className="border-2 border-dashed border-gray-300 rounded-
lg p-4 text-center">
                <Camera size={32} className="mx-auto text-gray-400 mb-2" />
                <div className="text-xs text-gray-500">Pending Upload</div>
              </div>
            )))
            {photos.length === 0 && (
              <div className="col-span-3 text-center py-8 text-gray-500 text-sm">
                No photos added yet
              </div>
            )}
          </div>
        </div>
      </div>
    </div>
  </div>
}

{activeSection === 10 && (
  <div className="space-y-6">
    <div className="bg-white rounded-lg border border-gray-200 p-6">
      <h2 className="text-lg font-bold text-gray-900 mb-4">Section 10: Activities of Daily
Living (ADLs)</h2>
      <div className="overflow-x-auto">
        <table className="w-full text-sm">
          <thead>
            <tr className="border-b-2 border-gray-300">
              <th className="text-left py-3 px-2 font-semibold">ADL Category</th>
              <th className="text-center py-3 px-2 font-semibold w-32">Independent?</th>
              <th className="text-center py-3 px-2 font-semibold w-32">Support Required?</
th>
              <th className="text-left py-3 px-2 font-semibold">Details of Support Needed</
th>
            </tr>
          </thead>
          <tbody>
            {adlCategories.map((category, idx) => (
              <tr key={idx} className="border-b border-gray-200 hover:bg-gray-50">
                <td className="py-3 px-2 font-medium text-gray-700">{category}</td>
                <td className="py-3 px-2 text-center">
                  <select className="border border-gray-300 rounded px-2 py-1 w-20">
                    <option value="">--</option>
                    <option value="Y">Yes</option>
                    <option value="N">No</option>
                  </select>
                </td>
              </tr>
            )))
          </tbody>
        </table>
      </div>
    </div>
  </div>
)

```

```

        </select>
    </td>
    <td className="py-3 px-2 text-center">
        <select className="border border-gray-300 rounded px-2 py-1 w-20">
            <option value="">—</option>
            <option value="Y">Yes</option>
            <option value="N">No</option>
        </select>
    </td>
    <td className="py-3 px-2">
        <input
            type="text"
            className="w-full border border-gray-300 rounded px-2 py-1"
            placeholder="Describe support needs...">
    </td>
</tr>
)}
{customADLs.map((custom, idx) => (
    <tr key={custom-$idx} className="border-b border-gray-200 bg-blue-50">
        <td className="py-3 px-2">
            <input
                type="text"
                className="w-full border border-blue-300 rounded px-2 py-1 font-medium"
                placeholder="Custom ADL category..."
                value={custom.category}
                onChange={(e) => {
                    const updated = [...customADLs];
                    updated[idx].category = e.target.value;
                    setCustomADLs(updated);
                }}
            />
        </td>
        <td className="py-3 px-2 text-center">
            <select className="border border-blue-300 rounded px-2 py-1 w-20">
                <option value="">—</option>
                <option value="Y">Yes</option>
                <option value="N">No</option>
            </select>
        </td>
        <td className="py-3 px-2 text-center">
            <select className="border border-blue-300 rounded px-2 py-1 w-20">
                <option value="">—</option>
                <option value="Y">Yes</option>
                <option value="N">No</option>
            </select>
        </td>
        <td className="py-3 px-2">
            <input
                type="text"
                className="w-full border border-blue-300 rounded px-2 py-1"
                placeholder="Describe support needs...">
        </td>
    </tr>
)}
</tbody>
</table>
</div>

```

```

        <button
          onClick={addCustomADL}
          className="mt-4 px-4 py-2 border-2 border-dashed border-blue-300 rounded text-blue-600 hover:bg-blue-50 text-sm font-medium w-full"
        >
          + Add Custom ADL Category
        </button>
      </div>
    </div>
  </>
}

{activeSection !== 7 && activeSection !== 10 && (
  <div className="bg-white rounded-lg border border-gray-200 p-6">
    <h2 className="text-lg font-bold text-gray-900 mb-4">
      Section {activeSection}: {sections.find(s => s.id === activeSection)?.title}
    </h2>
    <div className="text-gray-600">
      <p className="mb-4">Section content will appear here...</p>
      <p className="text-sm text-gray-500">
        This is a placeholder. Each section will have its specific form fields based on the INA
        Form PDF template.
      </p>
    </div>
    </div>
  </>
)
</div>
</div>

{showBodyMap && <BodyMapModal />}

<div className="bg-white border-t border-gray-200 shadow-lg">
  <div className="max-w-7xl mx-auto px-4 py-3 flex items-center justify-between">
    <div className="flex items-center gap-4">
      <button className="flex items-center gap-2 px-3 py-1.5 border border-gray-300 rounded
text-sm hover:bg-gray-50">
        <Mic size={16} />
        Voice Note
      </button>
      <div className="text-xs text-gray-500">
        {photos.length} photo(s) • {bodyMapMarks.length} body map mark(s)
      </div>
    </div>
    <div className="flex items-center gap-2">
      <button className="px-4 py-2 border border-gray-300 rounded text-sm hover:bg-
gray-50">
        Save Draft
      </button>
      <button
        onClick={() => {
          if (activeSection < 15) setActiveSection(activeSection + 1);
        }}
        className="px-4 py-2 bg-blue-600 text-white rounded text-sm hover:bg-blue-700 flex
items-center gap-2"
      >
        Next Section
        <ChevronRight size={16} />
      </button>
    </div>
  </div>
</div>
</div>

```

```
</div>
);
}
```