

```

import React, { useState } from 'react';
import { Phone, Calendar, Clock, FileText, Check, X, AlertCircle, User, MapPin } from 'lucide-react';

export default function PreINAContactDashboard() {
  const [expandedCase, setExpandedCase] = useState(null);
  const [callOutcome, setCallOutcome] = useState('');
  const [clientStory, setClientStory] = useState('');
  const [visitDate, setVisitDate] = useState('');
  const [visitTime, setVisitTime] = useState('');
  const [specialNeeds, setSpecialNeeds] = useState('');
  const [callDuration, setCallDuration] = useState('');

  // Sample cases - focusing on Stage 2 (Pre-INA Contact)
  const cases = [
    {
      id: 'INA-2024-001',
      client: 'Sarah Johnson',
      stage: 'Pre-INA Contact',
      stageNum: 2,
      consent: 'Received',
      days: 1,
      action: 'Call client to book INA',
      address: '42 High Street, Glasgow G1 1LX'
    },
    {
      id: 'INA-2024-002',
      client: 'Michael Chen',
      stage: 'R&D Phase',
      stageNum: 7,
      consent: 'Received',
      days: 5,
      action: 'Contact GP for records'
    },
    {
      id: 'INA-2024-003',
      client: 'Emma Davis',
      stage: 'Pre-INA Contact',
      stageNum: 2,
      consent: 'Received',
      days: 2,
      action: 'Call client to book INA',
      address: '15 Park Avenue, Glasgow G3 6LP'
    }
  ];

  const handleSaveContact = (caseId) => {
    alert(✅ Pre-INA Contact saved for ${caseId}!\n\nNext: INA visit will appear in your calendar.');
```

setExpandedCase(null);

// Reset form

```

    setCallOutcome('');
    setClientStory('');
    setVisitDate('');
    setVisitTime('');
    setSpecialNeeds('');
    setCallDuration('');
  };

  return (
    <div className="min-h-screen bg-gray-100 p-6">

```

```

<div className="max-w-6xl mx-auto">
  {/ * Header */}
  <div className="bg-white rounded-lg shadow-sm p-6 mb-6">
    <h1 className="text-2xl font-bold text-gray-900 mb-2">Case Manager Dashboard</h1>
    <p className="text-sm text-gray-600">Welcome back, Liz • Friday, October 10, 2025</p>
  </div>

  {/ * Action Center */}
  <div className="mb-6">
    <h2 className="text-lg font-semibold mb-3 text-gray-700">Action Center</h2>
    <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-3">
      <div className="bg-red-50 border border-red-200 rounded-lg p-4">
        <div className="flex justify-between items-start mb-2">
          <AlertCircle className="text-red-500" size={20} />
          <span className="text-2xl font-bold text-red-600">3</span>
        </div>
        <div className="text-sm font-semibold text-red-700">Urgent</div>
        <div className="text-xs text-red-600">clients waiting for INA booking</div>
      </div>
      <div className="bg-yellow-50 border border-yellow-200 rounded-lg p-4">
        <div className="flex justify-between items-start mb-2">
          <Clock className="text-yellow-600" size={20} />
          <span className="text-2xl font-bold text-yellow-700">2</span>
        </div>
        <div className="text-sm font-semibold text-yellow-700">Today</div>
        <div className="text-xs text-yellow-600">INA visits scheduled</div>
      </div>
      <div className="bg-green-50 border border-green-200 rounded-lg p-4">
        <div className="flex justify-between items-start mb-2">
          <Check className="text-green-600" size={20} />
          <span className="text-2xl font-bold text-green-700">5</span>
        </div>
        <div className="text-sm font-semibold text-green-700">Ready</div>
        <div className="text-xs text-green-600">consents received - contact clients</div>
      </div>
      <div className="bg-blue-50 border border-blue-200 rounded-lg p-4">
        <div className="flex justify-between items-start mb-2">
          <FileText className="text-blue-600" size={20} />
          <span className="text-2xl font-bold text-blue-700">2</span>
        </div>
        <div className="text-sm font-semibold text-blue-700">Pending</div>
        <div className="text-xs text-blue-600">cases awaiting reader assignment</div>
      </div>
    </div>
  </div>

  {/ * Cases List */}
  <div className="bg-white rounded-lg shadow-sm">
    <div className="p-4 border-b border-gray-200">
      <h3 className="font-semibold text-gray-800">My Cases</h3>
      <p className="text-sm text-gray-500">6 cases shown</p>
    </div>

    <div className="divide-y divide-gray-200">
      {cases.map((c) => (
        <div key={c.id} className="p-4 hover:bg-gray-50 transition-colors">
          {/ * Case Header Row */}
          <div className="flex items-start justify-between mb-2">
            <div className="flex-1">
              <div className="flex items-center gap-3 mb-1">


```

```

<div className="font-medium text-gray-900">{c.id}</div>
<span className={text-xs px-2 py-1 rounded ${
  c.stageNum === 2
    ? 'bg-purple-100 text-purple-700'
    : 'bg-blue-100 text-blue-700'
}}>
  Stage {c.stageNum}: {c.stage}
</span>
</div>
<div className="text-sm text-gray-600 flex items-center gap-2">
  <User size={14} />
  {c.client}
</div>
{c.address && (
  <div className="text-xs text-gray-500 flex items-center gap-1 mt-1">
    <MapPin size={12} />
    {c.address}
  </div>
)}
</div>
<div className="text-right">
  <div className="text-xs text-gray-500 mb-1">Consent: {c.consent}</div>
  <div className="text-xs text-gray-400">{c.days} days ago</div>
</div>
</div>

{/* Action Row */}
<div className="flex items-center justify-between mt-3">
  <div className="text-sm text-gray-600">
    <span className="font-medium">Next:</span> {c.action}
  </div>
  <button
    onClick={() => setExpandedCase(expandedCase === c.id ? null : c.id)}
    className={px-4 py-1.5 rounded text-sm font-medium transition-colors ${
      expandedCase === c.id
        ? 'bg-gray-200 text-gray-700'
        : c.stageNum === 2
          ? 'bg-purple-600 text-white hover:bg-purple-700'
          : 'bg-blue-600 text-white hover:bg-blue-700'
        }}
  >
    {expandedCase === c.id ? 'Collapse' : c.stageNum === 2 ? 'Start Call' : 'Expand'}
  </button>
</div>

{/* Expanded Pre-INA Contact Interface */}
{expandedCase === c.id && c.stageNum === 2 && (
  <div className="mt-4 pt-4 border-t border-gray-200 bg-purple-50 rounded-lg p-4">
    <div className="flex items-center gap-2 mb-4 pb-3 border-b border-purple-200">
      <Phone className="text-purple-600" size={20} />
      <h4 className="font-semibold text-purple-900">Pre-INA Contact - {c.id}</h4>
    </div>

    <div className="space-y-4">
      {/* Call Details */}
      <div className="bg-white rounded-lg p-4 border border-purple-200">
        <label className="block text-sm font-semibold text-gray-700 mb-3">
           Call Details
        </label>

```

```

<div className="grid grid-cols-1 md:grid-cols-3 gap-3">
  <div>
    <label className="block text-xs text-gray-600 mb-1">Date</label>
    <input
      type="date"
      className="w-full border border-gray-300 rounded px-3 py-2 text-sm"
      defaultValue={new Date().toISOString().split('T')[0]}
    />
  </div>
  <div>
    <label className="block text-xs text-gray-600 mb-1">Time</label>
    <input
      type="time"
      className="w-full border border-gray-300 rounded px-3 py-2 text-sm"
      defaultValue={new Date().getTimeString().slice(0,5)}
    />
  </div>
  <div>
    <label className="block text-xs text-gray-600 mb-1">Duration (mins)</label>
    <input
      type="number"
      className="w-full border border-gray-300 rounded px-3 py-2 text-sm"
      placeholder="15"
      value={callDuration}
      onChange={(e) => setCallDuration(e.target.value)}
    />
  </div>
</div>
<div className="mt-3">
  <label className="block text-xs text-gray-600 mb-2">Call Outcome</label>
  <div className="flex gap-2 flex-wrap">
    {[ 'Reached - Booking Made', 'Reached - Follow-up Needed', 'Voicemail Left',
'No Answer', 'Wrong Number' ].map((outcome) => (
      <button
        key={outcome}
        onClick={() => setCallOutcome(outcome)}
        className={ `px-3 py-1.5 rounded text-xs font-medium transition-colors ${
          callOutcome === outcome
            ? 'bg-purple-600 text-white'
            : 'bg-gray-100 text-gray-700 hover:bg-gray-200'
        }`}
      >
        {outcome}
      </button>
    ))}
  </div>
</div>
</div>

{/* Client Story */}
<div className="bg-white rounded-lg p-4 border border-purple-200">
  <label className="block text-sm font-semibold text-gray-700 mb-2">
     Client's Story & Concerns
  </label>
  <textarea
    className="w-full border border-gray-300 rounded px-3 py-2 text-sm"
    rows="3"
    placeholder="Brief notes about client's situation, key concerns, priority needs..."
    value={clientStory}
    onChange={(e) => setClientStory(e.target.value)}
  />
</div>

```

```

    />
    <p className="text-xs text-gray-500 mt-1">Keep it brief - detailed assessment
happens during INA visit</p>
</div>

{/* INA Visit Booking */}
{callOutcome === 'Reached - Booking Made' && (
  <div className="bg-white rounded-lg p-4 border border-green-300 border-2">
    <label className="block text-sm font-semibold text-gray-700 mb-3">
       17 INA Visit Booking
    </label>
    <div className="grid grid-cols-1 md:grid-cols-2 gap-3">
      <div>
        <label className="block text-xs text-gray-600 mb-1">Visit Date</label>
        <input
          type="date"
          className="w-full border border-gray-300 rounded px-3 py-2 text-sm"
          value={visitDate}
          onChange={(e) => setVisitDate(e.target.value)}
        />
      </div>
      <div>
        <label className="block text-xs text-gray-600 mb-1">Visit Time</label>
        <input
          type="time"
          className="w-full border border-gray-300 rounded px-3 py-2 text-sm"
          value={visitTime}
          onChange={(e) => setVisitTime(e.target.value)}
        />
      </div>
    </div>
    <div className="mt-3">
      <label className="block text-xs text-gray-600 mb-1">Special Requirements
(optional)</label>
      <input
        type="text"
        className="w-full border border-gray-300 rounded px-3 py-2 text-sm"
        placeholder="e.g., wheelchair access, interpreter needed, quiet environment..."
        value={specialNeeds}
        onChange={(e) => setSpecialNeeds(e.target.value)}
      />
    </div>
    <div className="mt-3 p-2 bg-blue-50 rounded text-xs text-blue-700">
      <strong>Address:</strong> {c.address}
    </div>
  </div>
)}

{/* Action Buttons */}
<div className="flex gap-2 pt-2">
  <button
    onClick={() => handleSaveContact(c.id)}
    className="flex-1 bg-purple-600 text-white px-4 py-2.5 rounded font-medium
hover:bg-purple-700 transition-colors flex items-center justify-center gap-2"
  >
    <Check size={16} />
    Save Contact Notes
  </button>
  <button
    onClick={() => setExpandedCase(null)}

```

```

        className="px-4 py-2.5 border border-gray-300 rounded text-gray-700
        hover:bg-gray-50 transition-colors"
    >
        Cancel
    </button>
</div>

    {/ * Helper Info */}
    <div className="bg-blue-50 border border-blue-200 rounded-lg p-3">
        <div className="text-xs font-semibold text-blue-800 mb-1">💡 What Happens
Next:</div>
        <div className="text-xs text-blue-700 space-y-1">
            <div>✓ Contact notes saved to case timeline</div>
            {callOutcome === 'Reached - Booking Made' && visitDate && (
                <>
                    <div>✓ INA visit added to your calendar for {visitDate}</div>
                    <div>✓ Client receives confirmation email</div>
                    <div>✓ INA Checklist will be available day before visit</div>
                </>
            )}
            {callOutcome === 'Voicemail Left' && (
                <div>🕒 Set reminder to follow up in 24-48 hours</div>
            )}
        </div>
    </div>
</div>
</div>
    )}

    {/ * For other stages - simpler expansion */}
    {expandedCase === c.id && c.stageNum !== 2 && (
        <div className="mt-4 pt-4 border-t border-gray-200">
            <div className="grid grid-cols-1 md:grid-cols-2 gap-3">
                <div className="border border-gray-200 rounded-lg p-3 bg-white">
                    <div className="text-sm font-semibold text-gray-700 mb-2">Timeline</div>
                    <div className="space-y-2 text-xs text-gray-600">
                        <div className="flex items-center gap-2">
                            <div className="w-2 h-2 bg-green-500 rounded-full"></div>
                            <span>Oct 7: INA Visit Completed</span>
                        </div>
                        <div className="flex items-center gap-2">
                            <div className="w-2 h-2 bg-green-500 rounded-full"></div>
                            <span>Oct 5: Visit Scheduled</span>
                        </div>
                        <div className="flex items-center gap-2">
                            <div className="w-2 h-2 bg-gray-300 rounded-full"></div>
                            <span>Oct 1: Case Opened</span>
                        </div>
                    </div>
                </div>
                <div className="border border-gray-200 rounded-lg p-3 bg-white">
                    <div className="text-sm font-semibold text-gray-700 mb-2">Quick Actions</div>
                    <div className="space-y-2">
                        <button className="w-full bg-blue-600 text-white text-sm py-2 px-3 rounded
                        hover:bg-blue-700">
                            View R&D Notes
                        </button>
                    </div>
                </div>
            </div>
        </div>
    )}

```

```

    <button className="w-full border border-gray-300 text-gray-700 text-sm py-2
px-3 rounded hover:bg-gray-50">
      Contact Client
    </button>
  </div>
</div>
</div>
</div>
)}
</div>
)}}
</div>
</div>
</div>
</div>
);
}

```