Phase 1

```tsx
"use client"

import axios from "axios"
import type React from "react"

import { useEffect, useState, useRef } from "react"
import { Dialog, DialogContent, DialogHeader,
DialogTitle } from "@/components/ui/dialog"
import { Button } from "@/components/ui/button"
import { Input } from "@/components/ui/input"
import { Label } from "@/components/ui/label"
import { Textarea } from "@/components/ui/textarea"
import { Select, SelectContent, SelectItem,
SelectTrigger, SelectValue } from
"@/components/ui/select"
import { Checkbox } from "@/components/ui/checkbox"
import { RadioGroup, RadioGroupItem } from
"@/components/ui/radio-group"
import { Card, CardContent, CardHeader, CardTitle } from
"@/components/ui/card"
import { Separator } from "@/components/ui/separator"
import { ScrollArea } from "@/components/ui/scroll-area"
import { decryptObject } from "@/utils/decrypt"

interface Phase1FormModalProps {
  open: boolean
  onOpenChange: (open: boolean) => void
}

const initialFormState = {
  phase1_date: "",
  phase1_city: "",
  patient_id: "",

  hearing_loss: "",
  sign_language_use: "",
  speech_use: "",
  hearing_loss_cause: [""],
  ringing_sensation: "",
```

```
  ear_pain: "",
  hearing_satisfaction: "",
  repeat_request: "",

  ears_clear_for_impressions: "",
  left_wax: "",
  left_infection: false,
  left_perforation: false,
  left_tinnitus: false,
  left_atresia: false,
  left_implant: false,
  left_other: false,
  right_wax: false,
  right_infection: false,
  right_perforation: false,
  right_tinnitus: false,
  right_atresia: false,
  right_implant: false,
  right_other: false,
  medical_recommendation: "",
  medication_given: [""],
  otoscopy_comments: "",
  screening_method: "",
  left_ear_result: "",
  right_ear_result: "",
  satisfaction_with_hearing: "",
  left_ear_impression: false,
  right_ear_impression: false,
  impression_comments: "",
  completed_counseling: false,
  received_aftercare_info: false,
  trained_as_student_ambassador: false,
  ear_impressions_inspected: false,
  shf_id_card_given: false,
}

const api = axios.create({
  baseURL: process.env.NEXT_PUBLIC_API_URL ||
"http://localhost:9000",
  withCredentials: true,
})
```

```
export function Phase1FormModal({ open, onOpenChange }:
Phase1FormModalProps) {
  const [formData, setFormData] = useState({
...initialFormState })
  const initialRef = useRef({ ...initialFormState })
  const [confirmCloseOpen, setConfirmCloseOpen] =
useState(false)
  const [submitConfirmOpen, setSubmitConfirmOpen] =
useState(false)

  const hearingLossCauses = [
    "Medication",
    "Ear Infection",
    "Meningitis",
    "Aging",
    "Malaria",
    "Birth Trauma",
    "Tuberculosis",
    "HIV",
    "Other",
    "Unknown",
  ]

  const medications = ["Antiseptic", "Analgesic",
"Antifungal", "Antibiotic"]

  const isFormDirty = () => {
    const current = formData
    const initial = initialRef.current

    // simple shallow-detect: any string/non-empty
array/true boolean difference
    for (const key of Object.keys(initial) as (keyof
typeof initial)[]) {
      const a = (initial as any)[key]
      const b = (current as any)[key]

      if (Array.isArray(a) && Array.isArray(b)) {
        const aFiltered = a.filter(Boolean)
        const bFiltered = b.filter(Boolean)
```

```
        if (aFiltered.length !== bFiltered.length)
return true
        if (aFiltered.some((v, i) => v !==
bFiltered[i])) return true
      } else if (typeof a === "string") {
        if ((a || "").trim() !== (b || "").trim())
return true
      } else if (typeof a === "boolean") {
        if (a !== b) return true
      } else {
        if (a !== b) return true
      }
    }
    return false
  }

  const requestClose = () => {
    if (isFormDirty()) {
      setConfirmCloseOpen(true)
    } else {
      // safe to close
      onOpenChange(false)
      // reset form to initial state when closed
(optional)
      setFormData({ ...initialRef.current })
    }
  }

  const handleConfirmClose = () => {
    setConfirmCloseOpen(false)
    onOpenChange(false)
    setFormData({ ...initialRef.current })
  }

  const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault()
    // show confirmation modal so user can recheck data
    setSubmitConfirmOpen(true)
  }

  const handleConfirmSubmit = async () => {
```

```
    try {
      const token = sessionStorage.getItem("token") ||
localStorage.getItem("token")
      await api.post(
        "/api/phase1/registration",
        {
          // Map frontend fields to backend fields as
needed
          patient_id: formData.patient_id,
          registration_date: formData.phase1_date,
          city: formData.phase1_city,
          has_hearing_loss: formData.hearing_loss,
          uses_sign_language:
formData.sign_language_use,
          uses_speech: formData.speech_use,
          hearing_loss_causes:
formData.hearing_loss_cause,
          ringing_sensation: formData.ringing_sensation,
          ear_pain: formData.ear_pain,
          hearing_satisfaction_18_plus:
formData.hearing_satisfaction,
          conversation_difficulty:
formData.repeat_request,
          ears_clear_for_impressions:
formData.ears_clear_for_impressions,
          otoscopy: {
            left: {
              wax: formData.left_wax,
              infection: formData.left_infection,
              perforation: formData.left_perforation,
              tinnitus: formData.left_tinnitus,
              atresia: formData.left_atresia,
              implant: formData.left_implant,
              other: formData.left_other,
            },
            right: {
              wax: formData.right_wax,
              infection: formData.right_infection,
              perforation: formData.right_perforation,
              tinnitus: formData.right_tinnitus,
              atresia: formData.right_atresia,
```

```
              implant: formData.right_implant,
              other: formData.right_other,
            },
            medical_recommendation:
formData.medical_recommendation,
            medication_given:
formData.medication_given.filter(Boolean),
            comments: formData.otoscopy_comments,
          },
          hearing_screening: {
            method: formData.screening_method,
            left_ear_result: formData.left_ear_result,
            right_ear_result: formData.right_ear_result,
            satisfaction_with_hearing:
formData.satisfaction_with_hearing,
          },
          ear_impressions: {
            left: formData.left_ear_impression,
            right: formData.right_ear_impression,
            comments: formData.impression_comments,
          },
          final_quality_control: {
            ear_impressions_inspected:
formData.ear_impressions_inspected,
            shf_id_card_given:
formData.shf_id_card_given,
          },
        },
        {
          headers: token ? { Authorization: `Bearer
${token}` } : {},
        }
      )
      // Optionally show success message
      onOpenChange(false)
      setFormData({ ...initialRef.current })
      setSubmitConfirmOpen(false)
    } catch (error) {
      // Handle error (show error message)
      console.error("Failed to submit Phase 1 form:",
error)
```

```
    }
  }

  // Intercept dialog open change (covers overlay/X)
  const handleDialogOpenChange = (val: boolean) => {
    if (!val) {
      requestClose()
    } else {
      onOpenChange(true)
    }
  }

  return (
    <>
      <Dialog open={open}
onOpenChange={handleDialogOpenChange}>
        <DialogContent className="sm:max-w-[60vw] sm:h-
[94vh]">
          <DialogHeader>
            <DialogTitle>Phase 1 Form</DialogTitle>
          </DialogHeader>

          <ScrollArea className="h-[80vh] pr-4">
            <form onSubmit={handleSubmit}
className="space-y-6">
              {/* 1. REGISTRATION - Phase 1 Specific */}
              <Card>
                <CardHeader>
                  <CardTitle className="text-lg">1.1
Registration</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div className="grid grid-cols-3 gap-
4">
                    <div>
                      <Label htmlFor="patient_id">SHF -
ID</Label>
                      <Input
                        id="patient_id"
                        value={formData.patient_id}
                        placeholder="SHF - ID"
```

```
                        onChange={(e) => setFormData({
...formData, patient_id: e.target.value })}
                      />
                    </div>

                    <div>
                      <Label htmlFor="phase1_date">Phase
1 Date</Label>
                      <Input
                        id="phase1_date"
                        type="date"
                        value={formData.phase1_date}
                        onChange={(e) => setFormData({
...formData, phase1_date: e.target.value })}
                        required
                      />
                    </div>

                    <div>
                      <Label htmlFor="phase1_city">Phase
1 City</Label>
                      <Input
                        id="phase1_city"
                        placeholder="Phase 1 City"
                        value={formData.phase1_city}
                        onChange={(e) => setFormData({
...formData, phase1_city: e.target.value })}
                      />
                    </div>
                  </div>


                  {/* General Questions */}
                  <Separator className="my-4" />
                  <h4 className="font-semibold text-
2xl">General Questions</h4>

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

```
                    <Label className="font-medium mb-
4"><strong>1. Do you have a hearing
loss?</strong></Label>
                    <RadioGroup
                      className="grid grid-cols-3 gap-
4"
                      value={formData.hearing_loss}
                      onValueChange={(value) =>
setFormData({ ...formData, hearing_loss: value })}
                    >
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem value="Yes"
id="hearing_loss_yes" />
                        <Label
htmlFor="hearing_loss_yes">Yes</Label>
                      </div>
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem value="No"
id="hearing_loss_no" />
                        <Label
htmlFor="hearing_loss_no">No</Label>
                      </div>
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem
value="Undecided" id="hearing_loss_undecided" />
                        <Label
htmlFor="hearing_loss_undecided">Undecided</Label>
                      </div>
                    </RadioGroup>
                  </div>
                  <div>
                    <Label className="font-medium mb-
4" ><strong>2. Do you use sign
language?</strong></Label>
                    <RadioGroup
                      className="grid grid-cols-3 gap-
4"
```

```
                        value={formData.sign_language_use}
                        onValueChange={(value) => setFormData({ ...formData, sign_language_use: value })}
                      >
                        <div className="flex items-center space-x-2">
                          <RadioGroupItem value="Yes" id="sign_yes" />
                          <Label htmlFor="sign_yes">Yes</Label>
                        </div>
                        <div className="flex items-center space-x-2">
                          <RadioGroupItem value="No" id="sign_no" />
                          <Label htmlFor="sign_no">No</Label>
                        </div>
                        <div className="flex items-center space-x-2">
                          <RadioGroupItem value="A little" id="sign_little" />
                          <Label htmlFor="sign_little">A little</Label>
                        </div>
                      </RadioGroup>
                    </div>
                    <div>
                      <Label className="font-medium mb-4"><strong>3. Do you use speech?</strong></Label>
                      <RadioGroup
                        className="grid grid-cols-3 gap-4"
                        value={formData.speech_use}
                        onValueChange={(value) => setFormData({ ...formData, speech_use: value })}
                      >
                        <div className="flex items-center space-x-2">
```

```
                          <RadioGroupItem value="Yes"
id="speech_yes" />
                          <Label
htmlFor="speech_yes">Yes</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="No"
id="speech_no" />
                          <Label
htmlFor="speech_no">No</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="A
little" id="speech_little" />
                          <Label
htmlFor="speech_little">A little</Label>
                        </div>
                      </RadioGroup>
                    </div>
                  </div>

                  <div className="mt-4 mb-4">
                    <Label className="mb-4
block"><strong>4. Hearing Loss Cause:</strong></Label>
                    <div className="grid grid-cols-2
gap-4">
                      {hearingLossCauses.map((cause) =>
(
                        <div key={cause} className="flex
items-center space-x-2">
                          <Checkbox
                            id={cause}
                            checked={formData.hearing_lo
ss_cause?.includes(cause)}
                            onCheckedChange={(checked)
=> {
                              if (checked) {
                                setFormData({
                                  ...formData,
```

```
                                    hearing_loss_cause:
[...(formData.hearing_loss_cause || []), cause],
                                  })
                                } else {
                                  setFormData({
                                    ...formData,
                                    hearing_loss_cause:
formData.hearing_loss_cause.filter((c) => c !== cause),
                                  })
                                }
                              }}
                            />
                            <Label
htmlFor={cause}>{cause}</Label>
                          </div>
                        ))}
                      </div>
                    </div>

                    <div className="grid grid-cols gap-4">
                      <div>
                        <Label className="mb-4"><strong>5.
Do you experience a ringing
                          sensation in your
ear?</strong></Label>
                        <RadioGroup
                          className="grid grid-cols-3 gap-
4"
                          value={formData.ringing_sensatio
n}
                          onValueChange={(value) =>
setFormData({ ...formData, ringing_sensation: value })}
                        >
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem value="Yes"
id="ringing_yes" />
                            <Label
htmlFor="ringing_yes">Yes</Label>
                          </div>
```

```
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="No"
id="ringing_no" />
                          <Label
htmlFor="ringing_no">No</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="A
little" id="ringing_little" />
                          <Label
htmlFor="ringing_little">A little</Label>
                        </div>
                      </RadioGroup>
                    </div>
                    <div>
                      <Label className="mb-4"><strong>6.
Do you have pain in your ear?</strong></Label>
                      <RadioGroup
                        className="grid grid-cols-3 gap-
4"
                        value={formData.ear_pain}
                        onValueChange={(value) =>
setFormData({ ...formData, ear_pain: value })}
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Yes"
id="pain_yes" />
                          <Label
htmlFor="pain_yes">Yes</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="No"
id="pain_no" />
                          <Label
htmlFor="pain_no">No</Label>
                        </div>
```

```
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="A
little" id="pain_little" />
                          <Label htmlFor="pain_little">A
little</Label>
                        </div>
                      </RadioGroup>
                    </div>
                    <Separator className="my-4" />
                    <div>
                      <Label className="font-medium mb-
4"><strong>(FOR PATIENTS 18 & OLDER)</strong></Label>
                      {/* Question 7 */}
                      <div>
                        <Label className="font-medium
mb-4 block">
                          <strong>7. How satisfied are
you with your hearing?</strong>
                        </Label>
                        <RadioGroup
                          className="grid grid-cols-3
gap-4 mb-4"
                          value={formData.hearing_satisf
action}
                          onValueChange={(value) =>
setFormData({ ...formData, hearing_satisfaction: value
})}
                        >
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem
value="Unsatisfied" id="hearing_unsatisfied" />
                            <Label
htmlFor="hearing_unsatisfied">Unsatisfied</Label>
                          </div>
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem
value="Undecided" id="hearing_undecided" />
```

```
                            <Label
htmlFor="hearing_undecided">Undecided</Label>
                          </div>
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem
value="Satisfied" id="hearing_satisfied" />
                            <Label
htmlFor="hearing_satisfied">Satisfied</Label>
                          </div>
                        </RadioGroup>
                      </div>

                      {/* Question 8 */}
                      <div>
                        <Label className="font-medium
mb-4 block">
                          <strong>8. Do you ask people
to repeat themselves or speak louder in
conversation?</strong>
                        </Label>
                        <RadioGroup
                          className="grid grid-cols-3
gap-4"
                          value={formData.repeat_request
}
                          onValueChange={(value) =>
setFormData({ ...formData, repeat_request: value })}
                        >
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem value="No"
id="repeat_no" />
                            <Label
htmlFor="repeat_no">No</Label>
                          </div>
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem
value="Sometimes" id="repeat_sometimes" />
```

```
                            <Label
htmlFor="repeat_sometimes">Sometimes</Label>
                          </div>
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem value="Yes"
id="repeat_yes" />
                            <Label
htmlFor="repeat_yes">Yes</Label>
                          </div>
                        </RadioGroup>
                      </div>

                    </div>
                  </div>
                </CardContent>
              </Card>

              <Separator />

              {/* 2A. EAR SCREENING */}
              <Card>
                <CardHeader>
                  <CardTitle className="text-lg">2A. Ear
Screening</CardTitle>
                </CardHeader>
                <CardContent>
                  <div>
                    <Label className="mb-4">Ears Clear
for Impressions <strong>(IF YES, SKIP TO SECTION
3)</strong>:</Label>
                    <RadioGroup
                      className="grid grid-cols-2 gap-3"
                      value={formData.ears_clear_for_imp
ressions}
                      onValueChange={(value) =>
setFormData({ ...formData, ears_clear_for_impressions:
value })}
                    >
                      <div className="flex items-center
space-x-2">
```

```
                        <RadioGroupItem value="Yes"
id="clear_yes" />
                        <Label
htmlFor="clear_yes">Yes</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem value="No"
id="clear_no" />
                        <Label
htmlFor="clear_no">No</Label>
                      </div>
                    </RadioGroup>
                  </div>
                </CardContent>
              </Card>

              {/* 2B. OTOSCOPY - HIDDEN WHEN EARS CLEAR
FOR IMPRESSIONS = Yes */}
              {formData.ears_clear_for_impressions !==
"Yes" && (
                <Card>
                  <CardHeader>
                    <CardTitle className="text-lg">2B.
Otoscopy</CardTitle>
                  </CardHeader>
                  <CardContent className="space-y-4">
                    {/* Otoscopy content (unchanged) */}
                    <div className="grid grid-cols-2
gap-6">
                      <div>
                        <h4 className="font-medium mb-
3">Left Ear</h4>
                        <div className="space-y-2">
                          {[
                            { key: "left_wax", label:
"Wax" },
                            { key: "left_infection",
label: "Infection" },
                            { key: "left_perforation",
label: "Perforation" },
```

```
                            { key: "left_tinnitus",
label: "Tinnitus" },
                            { key: "left_atresia",
label: "Atresia" },
                            { key: "left_implant",
label: "Implant" },
                            { key: "left_other", label:
"Other" },
                          ].map(({ key, label }) => (
                            <div key={key}
className="flex items-center space-x-2">
                              <Checkbox
                                id={key}
                                checked={formData[key as
keyof typeof formData] as boolean}
                                onCheckedChange={(checke
d) => setFormData({ ...formData, [key]: checked })}
                              />
                              <Label
htmlFor={key}>{label}</Label>
                            </div>
                          ))}
                        </div>
                      </div>
                      <div>
                        <h4 className="font-medium mb-
3">Right Ear</h4>
                        <div className="space-y-2">
                          {[
                            { key: "right_wax", label:
"Wax" },
                            { key: "right_infection",
label: "Infection" },
                            { key: "right_perforation",
label: "Perforation" },
                            { key: "right_tinnitus",
label: "Tinnitus" },
                            { key: "right_atresia",
label: "Atresia" },
                            { key: "right_implant",
label: "Implant" },
```

```
                            { key: "right_other", label:
"Other" },
                          ].map(({ key, label }) => (
                            <div key={key}
className="flex items-center space-x-2">
                              <Checkbox
                                id={key}
                                checked={formData[key as
keyof typeof formData] as boolean}
                                onCheckedChange={(checke
d) => setFormData({ ...formData, [key]: checked })}
                              />
                              <Label
htmlFor={key}>{label}</Label>
                            </div>
                          ))}
                        </div>
                      </div>
                    </div>

                    <div>
                      <Label className="mb-4"
htmlFor="medical_recommendation">Medical
Recommendation</Label>
                      <RadioGroup
                        className="grid grid-cols-2 gap-
3"
                        value={formData.medical_recommen
dation}
                        onValueChange={(value) =>
setFormData({ ...formData, medical_recommendation: value
})}
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Yes"
id="medical_reco_left" />
                          <Label
htmlFor="medical_reco_left">Left</Label>
                        </div>
```

```
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="No"
id="medical_reco_right" />
                          <Label
htmlFor="medical_reco_right">Right</Label>
                        </div>
                      </RadioGroup>
                    </div>

                    <div>
                      <Label>Medication Given</Label>
                      <div className="grid grid-cols-2
gap-2 mt-2">
                        {medications.map((med) => (
                          <div key={med} className="flex
items-center space-x-2">
                            <Checkbox
                              id={`med_${med}`}
                              checked={formData.medicati
on_given.includes(med)}
                              onCheckedChange={(checked)
=> {
                                if (checked) {
                                  setFormData({
...formData, medication_given:
[...formData.medication_given, med] })
                                } else {
                                  setFormData({
                                    ...formData,
                                    medication_given:
formData.medication_given.filter((m) => m !== med),
                                  })
                                }
                              }}
                            />
                            <Label
htmlFor={`med_${med}`}>{med}</Label>
                          </div>
                        ))}
                      </div>
```

```
                  </div>

                  <div>
                    <Label className="mb-4"
htmlFor="otoscopy_comments">Comments</Label>
                    <Textarea
                      id="otoscopy_comments"
                      value={formData.otoscopy_comment
s}
                      onChange={(e) => setFormData({
...formData, otoscopy_comments: e.target.value })}
                    />
                  </div>
                </CardContent>
              </Card>
            )}

            {/* 3. HEARING SCREENING */}
            <Card>
              <CardHeader>
                <CardTitle className="text-lg">3.
Hearing Screening</CardTitle>
              </CardHeader>
              <CardContent className="space-y-4">
                <div>
                  <Label className="mb-4">Screening
Method</Label>
                  <RadioGroup
                    className="grid grid-cols-2 gap-4"
                    value={formData.screening_method}
                    onValueChange={(value) =>
                      setFormData({ ...formData,
screening_method: value })
                    }
                  >
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem
value="Audiogram" id="audiogram" />
                      <Label
htmlFor="audiogram">Audiogram</Label>
```

```
                    </div>
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="Voice
Test" id="voice_test" />
                      <Label htmlFor="voice_test">WFA®
Voice Test</Label>
                    </div>
                  </RadioGroup>
                </div>

                <div className="grid grid-cols-2 gap-
4">
                  <div>
                    <Label className="mb-4">Left Ear
Result</Label>
                    <RadioGroup
                      className="grid grid-cols-3 gap-
4"
                      value={formData.left_ear_result}
                      onValueChange={(value) =>
setFormData({ ...formData, left_ear_result: value })}
                    >
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem value="Pass"
id="left_pass" />
                        <Label
htmlFor="left_pass">Pass</Label>
                      </div>
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem value="Fail"
id="left_fail" />
                        <Label
htmlFor="left_fail">Fail</Label>
                      </div>
                    </RadioGroup>
                  </div>
                  <div>
```

```
                    <Label className="mb-4">Right Ear
Result</Label>
                    <RadioGroup
                      className="grid grid-cols-3 gap-
4"
                      value={formData.right_ear_result
}
                      onValueChange={(value) =>
setFormData({ ...formData, right_ear_result: value })}
                    >
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem value="Pass"
id="right_pass" />
                        <Label
htmlFor="right_pass">Pass</Label>
                      </div>
                      <div className="flex items-
center space-x-2 ">
                        <RadioGroupItem value="Fail"
id="right_fail" />
                        <Label
htmlFor="right_fail">Fail</Label>
                      </div>
                    </RadioGroup>
                  </div>
                </div>
                <hr />
                <div>
                  <Label className="mb-4">Satisfaction
with Hearing (18+ if passes)</Label>
                  <RadioGroup
                    className="grid grid-cols-3 gap-4"
                    value={formData.satisfaction_with_
hearing}
                    onValueChange={(value) =>
setFormData({ ...formData, satisfaction_with_hearing:
value })}
                  >
                    <div className="flex items-center
space-x-2">
```

```
                        <RadioGroupItem
value="Unsatisfied" id="sat_unsatisfied" />
                        <Label
htmlFor="sat_unsatisfied">Unsatisfied</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Undecided" id="sat_undecided" />
                        <Label
htmlFor="sat_undecided">Undecided</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Satisfied" id="sat_satisfied" />
                        <Label
htmlFor="sat_satisfied">Satisfied</Label>
                      </div>
                    </RadioGroup>
                  </div>
                </CardContent>
              </Card>

              {/* 4. EAR IMPRESSIONS */}
              <Card>
                <CardHeader>
                  <CardTitle className="text-lg">4. Ear
Impressions</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div className="grid grid-cols-2 gap-
4">
                    <div className="flex items-center
space-x-2">
                      <Checkbox
                        id="left_ear_impression"
                        checked={formData.left_ear_impre
ssion}
                        onCheckedChange={(checked) =>
```

```
                        setFormData({ ...formData,
left_ear_impression: checked as boolean })
                      }
                    />
                    <Label
htmlFor="left_ear_impression">Left Ear
Impression</Label>
                  </div>
                  <div className="flex items-center
space-x-2">
                    <Checkbox
                      id="right_ear_impression"
                      checked={formData.right_ear_impr
ession}
                      onCheckedChange={(checked) =>
                        setFormData({ ...formData,
right_ear_impression: checked as boolean })
                      }
                    />
                    <Label
htmlFor="right_ear_impression">Right Ear
Impression</Label>
                  </div>
                </div>
                <div>
                  <Label
                    className="mb-4"
htmlFor="impression_comments">Comments</Label>
                  <Textarea
                    id="impression_comments"
                    value={formData.impression_comment
s}
                    onChange={(e) => setFormData({
...formData, impression_comments: e.target.value })}
                  />
                </div>
              </CardContent>
            </Card>

            {/* 5. FINAL QUALITY CONTROL */}
            <Card>
```

```
                <CardHeader>
                  <CardTitle className="text-lg">5.
Final Quality Control</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div className="grid grid-cols-2 gap-
4">
                    <div className="flex items-center
space-x-2">
                      <Checkbox
                        id="ear_impressions_inspected"
                        checked={formData.ear_impression
s_inspected}
                        onCheckedChange={(checked) =>
                          setFormData({ ...formData,
ear_impressions_inspected: checked as boolean })
                        }
                      />
                      <Label
htmlFor="ear_impressions_inspected">Ear impressions
inspected & collected</Label>
                    </div>
                    <div className="flex items-center
space-x-2">
                      <Checkbox
                        id="shf_id_card_given"
                        checked={formData.shf_id_card_gi
ven}
                        onCheckedChange={(checked) =>
setFormData({ ...formData, shf_id_card_given: checked as
boolean })}
                      />
                      <Label
htmlFor="shf_id_card_given">SHF ID number and ID card
given</Label>
                    </div>
                  </div>
                </CardContent>
              </Card>

            </form>
```

```
          </ScrollArea>

          <div className="flex justify-end space-x-1 pt-
1">
            <Button type="button" variant="outline"
onClick={requestClose}>
              Cancel
            </Button>
            <Button type="submit"
onClick={handleSubmit}>Submit Phase 1 Form</Button>
          </div>
        </DialogContent>
      </Dialog>

      {/* Confirmation dialog to avoid accidental close
*/}
      <Dialog open={confirmCloseOpen} onOpenChange={(v)
=> setConfirmCloseOpen(v)}>
        <DialogContent className="max-w-md">
          <DialogHeader>
            <DialogTitle>Discard changes?</DialogTitle>
          </DialogHeader>
          <div className="py-4">
            <p className="mb-4">You have unsaved
changes. Are you sure you want to close and discard
them?</p>
            <div className="flex justify-end gap-2">
              <Button variant="outline" onClick={() =>
setConfirmCloseOpen(false)}>Keep editing</Button>
              <Button
onClick={handleConfirmClose}>Discard & Close</Button>
            </div>
          </div>
        </DialogContent>
      </Dialog>

      {/* Submit confirmation: recheck patient data
before proceeding */}
      <Dialog open={submitConfirmOpen} onOpenChange={(v)
=> setSubmitConfirmOpen(v)}>
        <DialogContent className="max-w-md">
```

```
          <div className="py-4">
            <h3 className="text-lg font-semibold mb-
2">Please recheck patient data</h3>
            <p className="mb-4">Before submitting,
please confirm you have reviewed all patient data.
Proceed?</p>
            <div className="flex justify-end gap-2">
              <Button variant="outline" onClick={() =>
setSubmitConfirmOpen(false)}>Cancel</Button>
              <Button
onClick={handleConfirmSubmit}>Confirm & Submit</Button>
            </div>
          </div>
        </DialogContent>
      </Dialog>
    </>
  )
}
```

Phase 2

```
"use client"

import type React from "react"

import { useState, useRef, useEffect } from "react"
import { Dialog, DialogContent, DialogHeader,
DialogTitle } from "@/components/ui/dialog"
import { Button } from "@/components/ui/button"
import { Input } from "@/components/ui/input"
import { Label } from "@/components/ui/label"
import { Textarea } from "@/components/ui/textarea"
import { Select, SelectContent, SelectItem,
SelectTrigger, SelectValue } from
"@/components/ui/select"
import { Checkbox } from "@/components/ui/checkbox"
import { RadioGroup, RadioGroupItem } from
"@/components/ui/radio-group"
import { Card, CardContent, CardHeader, CardTitle } from
"@/components/ui/card"
```

```tsx
import { Separator } from "@/components/ui/separator"
import { ScrollArea } from "@/components/ui/scroll-area"

interface Phase2FormModalProps {
  open: boolean
  onOpenChange: (open: boolean) => void
}

export function Phase2FormModal({ open, onOpenChange }:
Phase2FormModalProps) {
  const initialState = {
    // Registration
    patient_id: "",
    phase2_date: "",
    phase2_city: "",
    patient_type: "",

    // Ear Screening
    ears_clear_for_fitting: "",

    // Otoscopy - Left Ear
    left_wax: false,
    left_infection: false,
    left_perforation: false,
    left_tinnitus: false,
    left_atresia: false,
    left_implant: false,
    left_other: false,

    // Otoscopy - Right Ear
    right_wax: false,
    right_infection: false,
    right_perforation: false,
    right_tinnitus: false,
    right_atresia: false,
    right_implant: false,
    right_other: false,

    medical_recommendation: "",
    medication_given: [""],
    otoscopy_comments: "",
```

```
  // Hearing Screening
  screening_method: "",
  left_ear_result: "",
  right_ear_result: "",
  satisfaction_with_hearing: "",

  // Hearing Aid Fitting
  fitter_name: "",
  num_hearing_aids_fit: 0,
  special_device: "",
  reason_for_fewer_aids: "",
  fitting_comments: "",
  patient_clear_for_counseling: false,
  reasons: {
    left: [] as string[],
    right: [] as string[],
  },

  // Fitting Results - Left Ear
  left_power_level: "",
  left_volume: "",
  left_model: "",
  left_battery_type: "",
  left_earmold_type: "",

  // Fitting Results - Right Ear
  right_power_level: "",
  right_volume: "",
  right_model: "",
  right_battery_type: "",
  right_earmold_type: "",

  // Counseling
  completed_counseling: false,
  received_aftercare_info: false,
  trained_as_student_ambassador: false,

  // Final Quality Control
  batteries_13: 0,
  batteries_675: 0,
```

```
    final_satisfaction: "",
    patient_signature: false,
    parent_guardian_signature: false,
    final_comments: "",
  }

  const [formData, setFormData] = useState({
...initialState })
  const initialRef = useRef({ ...initialState })
  const [confirmCloseOpen, setConfirmCloseOpen] =
useState(false)
  const [submitConfirmOpen, setSubmitConfirmOpen] =
useState(false)

  const fittingRef = useRef<HTMLDivElement | null>(null)
  const scrollAreaRef = useRef<HTMLDivElement |
null>(null)
  const fitterInputRef = useRef<HTMLInputElement |
null>(null)

  const medications = ["Antiseptic", "Analgesic",
"Antifungal", "Antibiotic"]
  const reasonsForFewerAids = ["No Response", "Other",
"Normal Hearing", "Distortion", "Implant",
"Recruitment"]

  // Simple dirty check
  const isFormDirty = () => {
    const current = formData
    const initial = initialRef.current

    for (const key of Object.keys(initial) as (keyof
typeof initial)[]) {
      const a = (initial as any)[key]
      const b = (current as any)[key]

      if (Array.isArray(a) && Array.isArray(b)) {
        const aFiltered = a.filter(Boolean)
        const bFiltered = b.filter(Boolean)
        if (aFiltered.length !== bFiltered.length)
return true
```

```
        if (aFiltered.some((v, i) => v !==
bFiltered[i])) return true
      } else if (typeof a === "string") {
        if ((a || "").trim() !== (b || "").trim())
return true
      } else if (typeof a === "boolean") {
        if (a !== b) return true
      } else if (typeof a === "number") {
        if (a !== b) return true
      } else {
        if (a !== b) return true
      }
    }
    return false
  }

  // Intercept dialog open change to protect unsaved
changes
  const handleDialogOpenChange = (val: boolean) => {
    if (!val) {
      if (isFormDirty()) {
        setConfirmCloseOpen(true)
      } else {
        onOpenChange(false)
        setFormData({ ...initialRef.current })
      }
    } else {
      onOpenChange(true)
    }
  }

  const requestClose = () => {
    if (isFormDirty()) {
      setConfirmCloseOpen(true)
    } else {
      onOpenChange(false)
      setFormData({ ...initialRef.current })
    }
  }

  const handleConfirmClose = () => {
```

```
    setConfirmCloseOpen(false)
    onOpenChange(false)
    setFormData({ ...initialRef.current })
  }

  const OTOSCOPY_KEYS = [
    "left_wax",
    "left_infection",
    "left_perforation",
    "left_tinnitus",
    "left_atresia",
    "left_implant",
    "left_other",
    "right_wax",
    "right_infection",
    "right_perforation",
    "right_tinnitus",
    "right_atresia",
    "right_implant",
    "right_other",
    "otoscopy_comments",
    "medical_recommendation",
    "medication_given",
  ] as (keyof typeof initialState)[]

  // When ears_clear_for_fitting becomes "Yes", hide
otoscopy and scroll to fitting section
  useEffect(() => {
    if (formData.ears_clear_for_fitting === "Yes") {
      // Clear otoscopy related fields to avoid stale
data
      setFormData((prev) => {
        const next = { ...prev } as any
        for (const k of OTOSCOPY_KEYS) {
          if (Array.isArray(next[k])) next[k] = []
          else if (typeof next[k] === "boolean") next[k]
= false
          else next[k] = "" // strings and comments
        }
        return next
      })
```

```
      // small delay to ensure DOM updated, then scroll
fitting into view and focus fitter name
      setTimeout(() => {
        if (fittingRef.current) {
          fittingRef.current.scrollIntoView({ behavior:
"smooth", block: "start" })
        } else if (scrollAreaRef.current) {
          scrollAreaRef.current.scrollTop = 9999
        }
        // focus fitter input if present
        setTimeout(() => {
          if (fitterInputRef.current)
fitterInputRef.current.focus()
        }, 120)
      }, 150)
    }
  }, [formData.ears_clear_for_fitting])

  const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault()
    setSubmitConfirmOpen(true)
  }

  const handleConfirmSubmit = () => {
    console.log("Phase 2 Form Data:", formData)
    // TODO: submit to backend
    onOpenChange(false)
    setFormData({ ...initialRef.current })
    setSubmitConfirmOpen(false)
  }

  return (
    <>
      {/* use intercepted handler so we can show confirm
dialog when closing with dirty state */}
      <Dialog open={open}
onOpenChange={handleDialogOpenChange}>
        <DialogContent className="sm:max-w-[60vw] sm:h-
[94vh]">
          <DialogHeader>
```

```
            <DialogTitle>Phase 2 Hearing Aid Fitting &
Assessment</DialogTitle>
          </DialogHeader>

          {/* attach scroll area ref if component
supports forwarded ref */}
          <ScrollArea className="h-[80vh] pr-4"
ref={scrollAreaRef as any}>
            <form onSubmit={handleSubmit}
className="space-y-6">
              {/* 1. REGISTRATION */}
              <Card>
                <CardHeader>
                  <CardTitle className="text-lg">1.
Registration</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div className="grid grid-cols-4 gap-
4">
                    <div>
                      <Label htmlFor="patient_id">SHF -
ID</Label>
                      <Input
                        id="patient_id"
                        value={formData.patient_id}
                        placeholder="SHF - ID"
                        onChange={(e) => setFormData({
...formData, patient_id: e.target.value })}
                      />
                    </div>
                    <div>
                      <Label htmlFor="phase2_date">Phase
2 Date</Label>
                      <Input
                        id="phase2_date"
                        type="date"
                        value={formData.phase2_date}
                        onChange={(e) => setFormData({
...formData, phase2_date: e.target.value })}
                        required
                      />
```

```
                  </div>
                  <div>
                    <Label htmlFor="phase2_city">Phase
2 City</Label>
                    <Input
                      id="phase2_city"
                      value={formData.phase2_city}
                      placeholder="Phase 2 City"
                      onChange={(e) => setFormData({
...formData, phase2_city: e.target.value })}
                    />
                  </div>
                  <div>
                    <Label>Patient Type</Label>
                    <Select
                      value={formData.patient_type}
                      onValueChange={(value) =>
setFormData({ ...formData, patient_type: value })}
                    >
                      <SelectTrigger className="w-
full">
                        <SelectValue
placeholder="Select type" />
                      </SelectTrigger>
                      <SelectContent>
                        <SelectItem value="Registered
Phase 1">Registered Phase 1</SelectItem>
                        <SelectItem value="Walk-in No
Earmolds">Walk-in No Earmolds</SelectItem>
                      </SelectContent>
                    </Select>
                  </div>
                </div>
              </CardContent>
            </Card>

            <Separator />

            {/* 2A. EAR SCREENING */}
            <Card>
              <CardHeader>
```

```
                  <CardTitle className="text-lg">2A. Ear
Screening</CardTitle>
                </CardHeader>
                <CardContent>
                  <div>
                    <Label>Ears Clear for
Fitting</Label>
                    <RadioGroup
                      className="grid grid-cols-3 gap-4
mt-4"
                      value={formData.ears_clear_for_fit
ting}
                      onValueChange={(value) =>
                        // handle change and let effect
do the rest (clear + scroll)
                        setFormData((prev) => ({
...prev, ears_clear_for_fitting: value }))
                      }
                    >
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem value="Yes"
id="clear_fitting_yes" />
                        <Label
htmlFor="clear_fitting_yes">Yes</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem value="No"
id="clear_fitting_no" />
                        <Label
htmlFor="clear_fitting_no">No</Label>
                      </div>
                    </RadioGroup>
                  </div>
                </CardContent>
              </Card>

              {/* 2B. OTOSCOPY - hidden when
ears_clear_for_fitting === "Yes" */}
```

```
            {formData.ears_clear_for_fitting !== "Yes"
&& (
              <Card>
                <CardHeader>
                  <CardTitle className="text-lg">2B.
Otoscopy</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div className="grid grid-cols-2
gap-6">
                    <div>
                      <h4 className="font-medium mb-
3">Left Ear</h4>
                      <div className="space-y-2">
                        {[
                          { key: "left_wax", label:
"Wax" },
                          { key: "left_infection",
label: "Infection" },
                          { key: "left_perforation",
label: "Perforation" },
                          { key: "left_tinnitus",
label: "Tinnitus" },
                          { key: "left_atresia",
label: "Atresia" },
                          { key: "left_implant",
label: "Implant" },
                          { key: "left_other", label:
"Other" },
                        ].map(({ key, label }) => (
                          <div key={key}
className="flex items-center space-x-2">
                            <Checkbox
                              id={key}
                              checked={formData[key as
keyof typeof formData] as boolean}
                              onCheckedChange={(checke
d) => setFormData({ ...formData, [key]: checked })}
                            />
                            <Label
htmlFor={key}>{label}</Label>
```

```
                            </div>
                          ))}
                        </div>
                      </div>
                      <div>
                        <h4 className="font-medium mb-
3">Right Ear</h4>
                        <div className="space-y-2">
                          {[
                            { key: "right_wax", label:
"Wax" },
                            { key: "right_infection",
label: "Infection" },
                            { key: "right_perforation",
label: "Perforation" },
                            { key: "right_tinnitus",
label: "Tinnitus" },
                            { key: "right_atresia",
label: "Atresia" },
                            { key: "right_implant",
label: "Implant" },
                            { key: "right_other", label:
"Other" },
                          ].map(({ key, label }) => (
                            <div key={key}
className="flex items-center space-x-2">
                              <Checkbox
                                id={key}
                                checked={formData[key as
keyof typeof formData] as boolean}
                                onCheckedChange={(checke
d) => setFormData({ ...formData, [key]: checked })}
                              />
                              <Label
htmlFor={key}>{label}</Label>
                            </div>
                          ))}
                        </div>
                      </div>
                    </div>
```

```
                    <div>
                      <Label className="mb-4"
htmlFor="medical_recommendation">Medical
Recommendation</Label>
                      <RadioGroup
                        className="grid grid-cols-2 gap-
3"
                        value={formData.medical_recommen
dation}
                        onValueChange={(value) =>
setFormData({ ...formData, medical_recommendation: value
})}
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Yes"
id="medical_reco_left" />
                          <Label
htmlFor="medical_reco_left">Left</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="No"
id="medical_reco_right" />
                          <Label
htmlFor="medical_reco_right">Right</Label>
                        </div>
                      </RadioGroup>
                    </div>

                    <div>
                      <Label>Medication Given</Label>
                      <div className="grid grid-cols-2
gap-2 mt-2">
                        {medications.map((med) => (
                          <div key={med} className="flex
items-center space-x-2">
                            <Checkbox
                              id={`med_${med}`}
                              checked={formData.medicati
on_given.includes(med)}
```

```
                                onCheckedChange={(checked)
=> {
                                  if (checked) {
                                    setFormData({
...formData, medication_given:
[...formData.medication_given, med] })
                                  } else {
                                    setFormData({
                                      ...formData,
                                      medication_given:
formData.medication_given.filter((m) => m !== med),
                                    })
                                  }
                                }}
                              />
                              <Label
htmlFor={`med_${med}`}>{med}</Label>
                            </div>
                          ))}
                        </div>
                      </div>

                      <div>
                        <Label htmlFor="otoscopy_comments"
className="mb-4">Comments</Label>
                        <Textarea
                          id="otoscopy_comments"
                          value={formData.otoscopy_comment
s}
                          onChange={(e) => setFormData({
...formData, otoscopy_comments: e.target.value })}
                        />
                      </div>
                    </CardContent>
                  </Card>
                )}

                {/* 3. HEARING SCREENING */}
                <Card>
                  <CardHeader>
```

```
                  <CardTitle className="text-lg">3.
Hearing Screening</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div>
                    <Label className="mb-4">Screening
Method (ONLY for Walk-in patients):</Label>
                    <RadioGroup
                      className="grid grid-cols-2 gap-4
mb-6"
                      value={formData.screening_method}
                      onValueChange={(value) =>
                        setFormData({ ...formData,
screening_method: value })
                      }
                    >
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Audiogram" id="audiogram" />
                        <Label
htmlFor="audiogram">Audiogram</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem value="Voice
Test" id="voice_test" />
                        <Label htmlFor="voice_test">WFA®
Voice Test</Label>
                      </div>
                    </RadioGroup>
                  </div>

                  <div className="grid grid-cols-2 gap-
4">
                    <div>
                      <Label>Left Ear Result</Label>
                      <RadioGroup
                        className="grid grid-cols-2 gap-
3 mt-4"
                        value={formData.left_ear_result}
```

```
                        onValueChange={(value) =>
setFormData({ ...formData, left_ear_result: value })}
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Pass"
id="left_pass" />
                          <Label
htmlFor="left_pass">Pass</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Fail"
id="left_fail" />
                          <Label
htmlFor="left_fail">Fail</Label>
                        </div>
                      </RadioGroup>
                    </div>
                    <div>
                      <Label>Right Ear Result</Label>
                      <RadioGroup
                        className="grid grid-cols-2 gap-
4 mt-4"
                        value={formData.right_ear_result
}
                        onValueChange={(value) =>
setFormData({ ...formData, right_ear_result: value })}
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Pass"
id="right_pass" />
                          <Label
htmlFor="right_pass">Pass</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Fail"
id="right_fail" />
```

```
                          <Label
htmlFor="right_fail">Fail</Label>
                        </div>
                      </RadioGroup>
                    </div>
                  </div>

                  <div>
                    <Label>Satisfaction with Hearing
(18+ if passes)</Label>
                    <RadioGroup
                      className="grid grid-cols-3 gap-4
mt-4"
                      value={formData.satisfaction_with_
hearing}
                      onValueChange={(value) =>
setFormData({ ...formData, satisfaction_with_hearing:
value })}
                    >
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Unsatisfied" id="sat_unsatisfied" />
                        <Label
htmlFor="sat_unsatisfied">Unsatisfied</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Undecided" id="sat_undecided" />
                        <Label
htmlFor="sat_undecided">Undecided</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Satisfied" id="sat_satisfied" />
                        <Label
htmlFor="sat_satisfied">Satisfied</Label>
                      </div>
                    </RadioGroup>
```

```
                </div>
              </CardContent>
            </Card>

            {/* 4A. HEARING AID FITTING */}
            {/* wrap with ref so we can scroll/focus
when ears_clear_for_fitting === "Yes" */}
            <div ref={fittingRef}>
              <Card>
                <CardHeader className="p-4 border-b
border-gray-200">
                  <div className="flex justify-between
items-center">
                    <CardTitle className="text-lg">4A.
Hearing Aid Fitting</CardTitle>
                    <div className="flex items-center
space-x-2 text-sm">
                      <Label htmlFor="fitter_name"
className="font-bold whitespace-nowrap">Fitter
Name:</Label>
                      <Input
                        id="fitter_name"
                        ref={fitterInputRef as any}
                        value={formData.fitter_name}
                        placeholder="Fitter Name"
                        onChange={(e) => setFormData({
...formData, fitter_name: e.target.value })}
                        className="h-7 w-40 p-1
border-b border-t-0 border-l-0 border-r-0 rounded-none
focus-visible:ring-0"
                      />
                    </div>
                  </div>
                </CardHeader>

                <CardContent className="space-y-4 pt-
4">
                  {/* FIX 2: Implement the Main
Results Table (Combines 4A's old data with 4B's old
data) */}
```

```
                    <table className="w-full border-
collapse border border-gray-400 text-sm">
                      <thead className="bg-gray-100">
                        <tr className="border-b border-
gray-400">
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto text-left
w-[10%]">RESULTS</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[20%]">POWER LEVEL</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[20%]">VOLUME</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[20%]">MODEL</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[10%]">BATTERY</th>
                          <th className="font-bold p-2
text-black h-auto w-[25%]">EARMOLD</th>
                        </tr>
                      </thead>
                      <tbody>
                        {/* LEFT EAR ROW */}
                        <tr className="border-b border-
gray-400">
                          <td className="font-bold
border-r border-gray-400 p-2">LEFT EAR</td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* LEFT POWER LEVEL Input
*/}
                            <Input
                              id="left_power_level"
                              value={formData.left_power
_level}
                              onChange={(e) =>
setFormData({ ...formData, left_power_level:
e.target.value })}
```

```
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* LEFT VOLUME Input */}
                            <Input
                              id="left_volume"
                              value={formData.left_volum
e}
                              onChange={(e) =>
setFormData({ ...formData, left_volume: e.target.value
})}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* LEFT MODEL Input */}
                            <Input
                              id="left_model"
                              value={formData.left_model
}
                              onChange={(e) =>
setFormData({ ...formData, left_model: e.target.value
})}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                          <td className="border-r
border-gray-400 p-2">
                            {/* LEFT Battery Radio Group
*/}
                            <RadioGroup
                              value={formData.left_batte
ry_type}
                              onValueChange={(value) =>
setFormData({ ...formData, left_battery_type: value })}
```

```
                              className="flex space-x-4
justify-center"
                            >
                              <div className="flex
items-center space-x-1">
                                {/* NOTE: Using
RadioGroupItem for standard component look */}
                                <RadioGroupItem
value="13" id="left_batt_13" />
                                <Label
htmlFor="left_batt_13" className="font-normal text-
sm">13</Label>
                              </div>
                              <div className="flex
items-center space-x-1">
                                <RadioGroupItem
value="675" id="left_batt_675" />
                                <Label
htmlFor="left_batt_675" className="font-normal text-
sm">675</Label>
                              </div>
                            </RadioGroup>
                          </td>
                          <td className="p-0">
                            {/* LEFT EARMOLD Input */}
                            <Input
                              id="left_earmold_type"
                              value={formData.left_earmo
ld_type}
                              onChange={(e) =>
setFormData({ ...formData, left_earmold_type:
e.target.value })}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                        </tr>
                        {/* RIGHT EAR ROW */}
                        <tr>
                          <td className="font-bold
border-r border-gray-400 p-2">RIGHT EAR</td>
```

```
                          <td className="border-r
border-gray-400 p-0">
                            {/* RIGHT POWER LEVEL Input
*/}
                            <Input
                              id="right_power_level"
                              value={formData.right_powe
r_level}
                              onChange={(e) =>
setFormData({ ...formData, right_power_level:
e.target.value })}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* RIGHT VOLUME Input */}
                            <Input
                              id="right_volume"
                              value={formData.right_volu
me}
                              onChange={(e) =>
setFormData({ ...formData, right_volume: e.target.value
})}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* RIGHT MODEL Input */}
                            <Input
                              id="left_model"
                              value={formData.right_mode
l}
                              onChange={(e) =>
setFormData({ ...formData, right_model: e.target.value
})}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
```

```
                            />
                          </td>
                          <td classNam
e="border-r border-gray-400 p-2">
                            {/* RIGHT Battery Radio
Group */}
                            <RadioGroup
                              value={formData.right_batt
ery_type}
                              onValueChange={(value) =>
setFormData({ ...formData, right_battery_type: value })}
                              className="flex space-x-4
justify-center"
                            >
                              <div className="flex
items-center space-x-1">
                                <RadioGroupItem
value="13" id="right_batt_13" />
                                <Label
htmlFor="right_batt_13" className="font-normal text-
sm">13</Label>
                              </div>
                              <div className="flex
items-center space-x-1">
                                <RadioGroupItem
value="675" id="right_batt_675" />
                                <Label
htmlFor="right_batt_675" className="font-normal text-
sm">675</Label>
                              </div>
                            </RadioGroup>
                          </td>
                          <td className="p-0">
                            {/* RIGHT EARMOLD Input */}
                            <Input
                              id="right_earmold_type"
                              value={formData.right_earm
old_type}
                              onChange={(e) =>
setFormData({ ...formData, right_earmold_type:
e.target.value })}
```

```
                                    className="h-8 border-none
focus-visible:ring-0 text-center"
                                  />
                                </td>
                              </tr>
                            </tbody>
                          </table>

                          <div className="grid grid-cols-2
gap-4">
                            <div className="space-y-4 pr-2">
                              <div className="flex items-
center space-x-4">
                                <Label className="font-bold
text-sm whitespace-nowrap">Number of Hearing Aids
Fit:</Label>
                                <RadioGroup
                                  value={formData.num_hearing_
aids_fit.toString()}
                                  onValueChange={(value) =>
setFormData({ ...formData, num_hearing_aids_fit:
Number.parseInt(value) })}
                                  className="flex space-x-6"
                                >
                                  {['0', '1', '2'].map(val =>
(
                                    <div className="flex
items-center space-x-2" key={val}>
                                      <RadioGroupItem
value={val} id={`num_aids_${val}`} />
                                      <Label
htmlFor={`num_aids_${val}`}>{val}</Label>
                                    </div>
                                  ))}
                                </RadioGroup>
                              </div>

                              {/* Special Device (Now Radio
Group) */}
                              <RadioGroup
```

```
                        value={formData.special_device
}
                        onValueChange={(value) =>
setFormData({ ...formData, special_device: value })}
                        className="space-y-1 pt-2"
                      >
                        <Label className="font-bold
block mb-1 text-sm">Special Device:</Label>
                        <div className="space-y-1 ml-
4">
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem
value="Bone Conductor" id="special_bone" />
                            <Label
htmlFor="special_bone">Bone Conductor (675
battery)</Label>
                          </div>
                          <div className="flex items-
center space-x-2">
                            <RadioGroupItem
value="Body Aid" id="special_body" />
                            <Label
htmlFor="special_body">Body Aid (AA battery)</Label>
                          </div>
                        </div>
                      </RadioGroup>

                    </div>

                    {/* RIGHT Column: Reasons for 0 or
1 aid */}
                    <div className=" p-2 bg-gray-50
rounded-md">
                      <p className="mb-2 text-xs font-
bold">
                        If patient received 1 or 0
hearing aids, select option below:
                      </p>
```

```
                        <div className="flex flex-col
space-y-1">
                          {/* Header for LEFT and RIGHT
columns */}
                          <div className="flex justify-
end space-x-8 pr-1 pt-1 font-bold text-xs">
                            <Label>LEFT</Label>
                            <Label>RIGHT</Label>
                          </div>

                          {/* Mapping the reasons for
fewer aids */}
                          {reasonsForFewerAids.map((reas
on, index) => (
                            <div
                              key={index}
                              className="flex justify-
between items-center text-xs py-1" // Added items-center
for vertical alignment
                            >
                              {/* Reason label */}
                              <Label className="font-
normal w-3/5 whitespace-nowrap">
                                {reason}
                              </Label>

                              {/* LEFT and RIGHT
checkboxes - MODIFIED FOR CENTERING */}
                              {/* This container now
controls the space for the two checkbox columns */}
                              <div className="flex w-2/5
justify-around pr-2">
                                {/* LEFT checkbox
wrapper - Centers the checkbox */}
                                <div className="flex
justify-center w-1/2">
                                  <Checkbox
                                    id={`reason_left_${i
ndex}`}
                                    checked={formData.re
asons?.left?.includes(reason) || false}
```

```
                                    onCheckedChange={(ch
ecked) => {
                                      const updatedLeft
= checked
                                        ?
[...(formData.reasons?.left || []), reason]
                                        :
(formData.reasons?.left || []).filter((r) => r !==
reason);
                                      setFormData({
                                        ...formData,
                                        reasons: {
                                          ...formData.re
asons,
                                          left:
updatedLeft,
                                        },
                                      });
                                    }}
                                  />
                                </div>

                                {/* RIGHT checkbox
wrapper - Centers the checkbox */}
                                <div className="flex
justify-center w-1/2">
                                  <Checkbox
                                    id={`reason_right_${
index}`}
                                    checked={formData.re
asons?.right?.includes(reason) || false}
                                    onCheckedChange={(ch
ecked) => {
                                      const updatedRight
= checked
                                        ?
[...(formData.reasons?.right || []), reason]
                                        :
(formData.reasons?.right || []).filter((r) => r !==
reason);
                                      setFormData({
```

```
                                        ...formData,
                                        reasons: {
                                          ...formData.re
asons,
                                          right:
updatedRight,
                                        },
                                      });
                                    }}
                                  />
                                </div>
                              </div>
                            </div>
                          ))}
                        </div>
                      </div>

                    </div>

                    {/* Comments Section (Moved from old
bottom of 4A) */}
                    <div className="pt-2">
                      <Label htmlFor="fitting_comments"
className="font-bold">Comments:</Label>
                      <Textarea
                        id="fitting_comments"
                        value={formData.fitting_comments
}
                        onChange={(e) => setFormData({
...formData, fitting_comments: e.target.value })}
                        className="min-h-[60px]"
                      />
                    </div>
                    <Separator className="w-full" />
                    <CardTitle className="text-lg">4B.
Fitting Quality Control</CardTitle>
                    <CardContent className="space-y-4">
                      <div>
                        <Label className="font-medium
mb-2">
```

```
                        <strong>Patient clear for
counseling:</strong>
                      </Label>
                      <RadioGroup
                        className="grid grid-cols-2
gap-4"
                        value={formData.patient_clear_
for_counseling ? "Yes" : "No"}
                        onValueChange={(value) =>
                          setFormData({ ...formData,
patient_clear_for_counseling: value === "Yes" })
                        }
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Yes"
id="counseling_yes" />
                          <Label
htmlFor="counseling_yes">Yes</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="No"
id="counseling_no" />
                          <Label
htmlFor="counseling_no">No</Label>
                        </div>
                      </RadioGroup>
                    </div>

                  </CardContent>
                </CardContent>
              </Card>
            </div>
          {/* 5. COUNSELING */}
          <Card>
            <CardHeader>
              <CardTitle className="text-lg">5.
Counseling</CardTitle>
            </CardHeader>
            <CardContent className="space-y-4">
```

```
                <div className="flex items-center space-
x-2">
                  <Checkbox
                    id="completed_counseling"
                    checked={formData.completed_counseli
ng}
                    onCheckedChange={(checked) =>
                      setFormData({ ...formData,
completed_counseling: checked as boolean })
                    }
                  />
                  <Label htmlFor="completed_counseling">
                    Patient completed counseling and
received AfterCare information
                  </Label>
                </div>
                <div className="flex items-center space-
x-2">
                  <Checkbox
                    id="trained_as_student_ambassador"
                    checked={formData.trained_as_student
_ambassador}
                    onCheckedChange={(checked) =>
                      setFormData({ ...formData,
trained_as_student_ambassador: checked as boolean })
                    }
                  />
                  <Label
htmlFor="trained_as_student_ambassador">Patient trained
as a<strong>Student Ambassador</strong></Label>
                </div>
              </CardContent>
            </Card>

            {/* 6. FINAL QUALITY CONTROL */}
            <Card>
              <CardHeader>
                <CardTitle className="text-lg">6. Final
Quality Control</CardTitle>
              </CardHeader>
              <CardContent className="space-y-4">
```

```
                <div className="grid grid-cols-2 gap-4">
                  <div>
                    <Label htmlFor="batteries_13">Number
of batteries provided (13)</Label>
                    <Input
                      id="batteries_13"
                      type="number"
                      placeholder="0"
                      value={formData.batteries_13}
                      onChange={(e) => setFormData({
...formData, batteries_13:
Number.parseInt(e.target.value) })}
                    />
                  </div>
                  <div>
                    <Label
htmlFor="batteries_675">Number of batteries provided
(675)</Label>
                    <Input
                      id="batteries_675"
                      type="number"
                      placeholder="0"
                      value={formData.batteries_675}
                      onChange={(e) =>
                        setFormData({ ...formData,
batteries_675: Number.parseInt(e.target.value) })
                      }
                    />
                  </div>
                </div>

                <div>
                  <Label className="mb-2"><strong>(For
patients 18 & older)</strong></Label>
                  <Label>When wearing your hearing
aid(s) are you
                    satisfied with your hearing?</Label>
                  <RadioGroup
                    className="grid grid-cols-3 gap-4
mt-4"
                    value={formData.final_satisfaction}
```

```
                    onValueChange={(value) =>
setFormData({ ...formData, final_satisfaction: value })}
                  >
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem
value="Unsatisfied" id="final_unsatisfied" />
                      <Label
htmlFor="final_unsatisfied">Unsatisfied</Label>
                    </div>
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="Undecided"
id="final_undecided" />
                      <Label
htmlFor="final_undecided">Undecided</Label>
                    </div>
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="Satisfied"
id="final_satisfied" />
                      <Label
htmlFor="final_satisfied">Satisfied</Label>
                    </div>
                  </RadioGroup>
                </div>

                <div>
                  <Label
htmlFor="final_comments">Comments</Label>
                  <Textarea
                    id="final_comments"
                    value={formData.final_comments}
                    onChange={(e) => setFormData({
...formData, final_comments: e.target.value })}
                  />
                </div>
              </CardContent>
            </Card>
            </form>
          </ScrollArea>
```

```
          <div className="flex justify-end space-x-1 pt-
1">
            <Button type="button" variant="outline"
onClick={requestClose}>
              Cancel
            </Button>
            <Button type="submit"
onClick={handleSubmit}>Submit Phase 2 Form</Button>
          </div>
         </DialogContent>
       </Dialog>

       {/* Confirmation dialog to avoid accidental close
*/}
       <Dialog open={confirmCloseOpen} onOpenChange={(v)
=> setConfirmCloseOpen(v)}>
         <DialogContent className="max-w-md">
           <DialogHeader>
             <DialogTitle>Discard changes?</DialogTitle>
           </DialogHeader>
           <div className="py-4">
             <p className="mb-4">You have unsaved
changes. Are you sure you want to close and discard
them?</p>
             <div className="flex justify-end gap-2">
               <Button variant="outline" onClick={() =>
setConfirmCloseOpen(false)}>Keep editing</Button>
               <Button
onClick={handleConfirmClose}>Discard & Close</Button>
             </div>
           </div>
         </DialogContent>
       </Dialog>

       {/* Submit confirmation: recheck patient data
before proceeding */}
       <Dialog open={submitConfirmOpen} onOpenChange={(v)
=> setSubmitConfirmOpen(v)}>
         <DialogContent className="max-w-md">
           <div className="py-4">
```

```
            <h3 className="text-lg font-semibold mb-
2">Please recheck patient data</h3>
            <p className="mb-4">Before submitting,
please confirm you have reviewed all patient data.
Proceed?</p>
            <div className="flex justify-end gap-2">
              <Button variant="outline" onClick={() =>
setSubmitConfirmOpen(false)}>Cancel</Button>
              <Button
onClick={handleConfirmSubmit}>Confirm & Submit</Button>
            </div>
          </div>
        </DialogContent>
      </Dialog>
    </>
  )
}
```

Phase 3

```
"use client"

import type React from "react"

import { useState, useRef, useEffect } from "react"
import { Dialog, DialogContent, DialogHeader,
DialogTitle } from "@/components/ui/dialog"
import { Button } from "@/components/ui/button"
import { Input } from "@/components/ui/input"
import { Label } from "@/components/ui/label"
import { Textarea } from "@/components/ui/textarea"
import { Select, SelectContent, SelectItem,
SelectTrigger, SelectValue } from
"@/components/ui/select"
import { Checkbox } from "@/components/ui/checkbox"
import { RadioGroup, RadioGroupItem } from
"@/components/ui/radio-group"
```

```tsx
import { Card, CardContent, CardHeader, CardTitle } from
"@/components/ui/card"
import { Separator } from "@/components/ui/separator"
import { ScrollArea } from "@/components/ui/scroll-area"

interface Phase2FormModalProps {
  open: boolean
  onOpenChange: (open: boolean) => void
}

export function Phase2FormModal({ open, onOpenChange }:
Phase2FormModalProps) {
  const initialState = {
    // Registration
    patient_id: "",
    phase2_date: "",
    phase2_city: "",
    patient_type: "",

    // Ear Screening
    ears_clear_for_fitting: "",

    // Otoscopy - Left Ear
    left_wax: false,
    left_infection: false,
    left_perforation: false,
    left_tinnitus: false,
    left_atresia: false,
    left_implant: false,
    left_other: false,

    // Otoscopy - Right Ear
    right_wax: false,
    right_infection: false,
    right_perforation: false,
    right_tinnitus: false,
    right_atresia: false,
    right_implant: false,
    right_other: false,

    medical_recommendation: "",
```

```
    medication_given: [""],
    otoscopy_comments: "",

    // Hearing Screening
    screening_method: "",
    left_ear_result: "",
    right_ear_result: "",
    satisfaction_with_hearing: "",

    // Hearing Aid Fitting
    fitter_name: "",
    num_hearing_aids_fit: 0,
    special_device: "",
    reason_for_fewer_aids: "",
    fitting_comments: "",
    patient_clear_for_counseling: false,
    reasons: {
      left: [] as string[],
      right: [] as string[],
    },

    // Fitting Results - Left Ear
    left_power_level: "",
    left_volume: "",
    left_model: "",
    left_battery_type: "",
    left_earmold_type: "",

    // Fitting Results - Right Ear
    right_power_level: "",
    right_volume: "",
    right_model: "",
    right_battery_type: "",
    right_earmold_type: "",

    // Counseling
    completed_counseling: false,
    received_aftercare_info: false,
    trained_as_student_ambassador: false,

    // Final Quality Control
```

```
    batteries_13: 0,
    batteries_675: 0,
    final_satisfaction: "",
    patient_signature: false,
    parent_guardian_signature: false,
    final_comments: "",
  }

  const [formData, setFormData] = useState({
...initialState })
  const initialRef = useRef({ ...initialState })
  const [confirmCloseOpen, setConfirmCloseOpen] =
useState(false)
  const [submitConfirmOpen, setSubmitConfirmOpen] =
useState(false)

  const fittingRef = useRef<HTMLDivElement | null>(null)
  const scrollAreaRef = useRef<HTMLDivElement |
null>(null)
  const fitterInputRef = useRef<HTMLInputElement |
null>(null)

  const medications = ["Antiseptic", "Analgesic",
"Antifungal", "Antibiotic"]
  const reasonsForFewerAids = ["No Response", "Other",
"Normal Hearing", "Distortion", "Implant",
"Recruitment"]

  // Simple dirty check
  const isFormDirty = () => {
    const current = formData
    const initial = initialRef.current

    for (const key of Object.keys(initial) as (keyof
typeof initial)[]) {
      const a = (initial as any)[key]
      const b = (current as any)[key]

      if (Array.isArray(a) && Array.isArray(b)) {
        const aFiltered = a.filter(Boolean)
        const bFiltered = b.filter(Boolean)
```

```
        if (aFiltered.length !== bFiltered.length)
return true
        if (aFiltered.some((v, i) => v !==
bFiltered[i])) return true
      } else if (typeof a === "string") {
        if ((a || "").trim() !== (b || "").trim())
return true
      } else if (typeof a === "boolean") {
        if (a !== b) return true
      } else if (typeof a === "number") {
        if (a !== b) return true
      } else {
        if (a !== b) return true
      }
    }
    return false
  }

  // Intercept dialog open change to protect unsaved
changes
  const handleDialogOpenChange = (val: boolean) => {
    if (!val) {
      if (isFormDirty()) {
        setConfirmCloseOpen(true)
      } else {
        onOpenChange(false)
        setFormData({ ...initialRef.current })
      }
    } else {
      onOpenChange(true)
    }
  }

  const requestClose = () => {
    if (isFormDirty()) {
      setConfirmCloseOpen(true)
    } else {
      onOpenChange(false)
      setFormData({ ...initialRef.current })
    }
  }
```

```
  const handleConfirmClose = () => {
    setConfirmCloseOpen(false)
    onOpenChange(false)
    setFormData({ ...initialRef.current })
  }

  const OTOSCOPY_KEYS = [
    "left_wax",
    "left_infection",
    "left_perforation",
    "left_tinnitus",
    "left_atresia",
    "left_implant",
    "left_other",
    "right_wax",
    "right_infection",
    "right_perforation",
    "right_tinnitus",
    "right_atresia",
    "right_implant",
    "right_other",
    "otoscopy_comments",
    "medical_recommendation",
    "medication_given",
  ] as (keyof typeof initialState)[]

  // When ears_clear_for_fitting becomes "Yes", hide
otoscopy and scroll to fitting section
  useEffect(() => {
    if (formData.ears_clear_for_fitting === "Yes") {
      // Clear otoscopy related fields to avoid stale
data
      setFormData((prev) => {
        const next = { ...prev } as any
        for (const k of OTOSCOPY_KEYS) {
          if (Array.isArray(next[k])) next[k] = []
          else if (typeof next[k] === "boolean") next[k]
= false
          else next[k] = "" // strings and comments
        }
```

```
        return next
      })

      // small delay to ensure DOM updated, then scroll
fitting into view and focus fitter name
      setTimeout(() => {
        if (fittingRef.current) {
          fittingRef.current.scrollIntoView({ behavior:
"smooth", block: "start" })
        } else if (scrollAreaRef.current) {
          scrollAreaRef.current.scrollTop = 9999
        }
        // focus fitter input if present
        setTimeout(() => {
          if (fitterInputRef.current)
fitterInputRef.current.focus()
        }, 120)
      }, 150)
    }
  }, [formData.ears_clear_for_fitting])

  const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault()
    setSubmitConfirmOpen(true)
  }

  const handleConfirmSubmit = () => {
    console.log("Phase 2 Form Data:", formData)
    // TODO: submit to backend
    onOpenChange(false)
    setFormData({ ...initialRef.current })
    setSubmitConfirmOpen(false)
  }

  return (
    <>
      {/* use intercepted handler so we can show confirm
dialog when closing with dirty state */}
      <Dialog open={open}
onOpenChange={handleDialogOpenChange}>
```

```
        <DialogContent className="sm:max-w-[60vw] sm:h-
[94vh]">
          <DialogHeader>
            <DialogTitle>Phase 2 Hearing Aid Fitting &
Assessment</DialogTitle>
          </DialogHeader>

          {/* attach scroll area ref if component
supports forwarded ref */}
          <ScrollArea className="h-[80vh] pr-4"
ref={scrollAreaRef as any}>
            <form onSubmit={handleSubmit}
className="space-y-6">
              {/* 1. REGISTRATION */}
              <Card>
                <CardHeader>
                  <CardTitle className="text-lg">1.
Registration</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div className="grid grid-cols-4 gap-
4">
                    <div>
                      <Label htmlFor="patient_id">SHF -
ID</Label>
                      <Input
                        id="patient_id"
                        value={formData.patient_id}
                        placeholder="SHF - ID"
                        onChange={(e) => setFormData({
...formData, patient_id: e.target.value })}
                      />
                    </div>
                    <div>
                      <Label htmlFor="phase2_date">Phase
2 Date</Label>
                      <Input
                        id="phase2_date"
                        type="date"
                        value={formData.phase2_date}
```

```
                        onChange={(e) => setFormData({
...formData, phase2_date: e.target.value })}
                        required
                      />
                    </div>
                    <div>
                      <Label htmlFor="phase2_city">Phase
2 City</Label>
                      <Input
                        id="phase2_city"
                        value={formData.phase2_city}
                        placeholder="Phase 2 City"
                        onChange={(e) => setFormData({
...formData, phase2_city: e.target.value })}
                      />
                    </div>
                    <div>
                      <Label>Patient Type</Label>
                      <Select
                        value={formData.patient_type}
                        onValueChange={(value) =>
setFormData({ ...formData, patient_type: value })}
                      >
                        <SelectTrigger className="w-
full">
                          <SelectValue
placeholder="Select type" />
                        </SelectTrigger>
                        <SelectContent>
                          <SelectItem value="Registered
Phase 1">Registered Phase 1</SelectItem>
                          <SelectItem value="Walk-in No
Earmolds">Walk-in No Earmolds</SelectItem>
                        </SelectContent>
                      </Select>
                    </div>
                  </div>
                </CardContent>
              </Card>

              <Separator />
```

```
            {/* 2A. EAR SCREENING */}
            <Card>
              <CardHeader>
                <CardTitle className="text-lg">2A. Ear
Screening</CardTitle>
              </CardHeader>
              <CardContent>
                <div>
                  <Label>Ears Clear for
Fitting</Label>
                  <RadioGroup
                    className="grid grid-cols-3 gap-4
mt-4"
                    value={formData.ears_clear_for_fit
ting}
                    onValueChange={(value) =>
                      // handle change and let effect
do the rest (clear + scroll)
                      setFormData((prev) => ({
...prev, ears_clear_for_fitting: value }))
                    }
                  >
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="Yes"
id="clear_fitting_yes" />
                      <Label
htmlFor="clear_fitting_yes">Yes</Label>
                    </div>
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="No"
id="clear_fitting_no" />
                      <Label
htmlFor="clear_fitting_no">No</Label>
                    </div>
                  </RadioGroup>
                </div>
              </CardContent>
            </Card>
```

```
            {/* 2B. OTOSCOPY - hidden when
ears_clear_for_fitting === "Yes" */}
            {formData.ears_clear_for_fitting !== "Yes"
&& (
              <Card>
                <CardHeader>
                  <CardTitle className="text-lg">2B.
Otoscopy</CardTitle>
                </CardHeader>
                <CardContent className="space-y-4">
                  <div className="grid grid-cols-2
gap-6">
                    <div>
                      <h4 className="font-medium mb-
3">Left Ear</h4>
                      <div className="space-y-2">
                        {[
                          { key: "left_wax", label:
"Wax" },
                          { key: "left_infection",
label: "Infection" },
                          { key: "left_perforation",
label: "Perforation" },
                          { key: "left_tinnitus",
label: "Tinnitus" },
                          { key: "left_atresia",
label: "Atresia" },
                          { key: "left_implant",
label: "Implant" },
                          { key: "left_other", label:
"Other" },
                        ].map(({ key, label }) => (
                          <div key={key}
className="flex items-center space-x-2">
                            <Checkbox
                              id={key}
                              checked={formData[key as
keyof typeof formData] as boolean}
                              onCheckedChange={(checke
d) => setFormData({ ...formData, [key]: checked })}
```

```
                                />
                                <Label
htmlFor={key}>{label}</Label>
                              </div>
                            ))}
                          </div>
                        </div>
                        <div>
                          <h4 className="font-medium mb-
3">Right Ear</h4>
                          <div className="space-y-2">
                            {[
                              { key: "right_wax", label:
"Wax" },
                              { key: "right_infection",
label: "Infection" },
                              { key: "right_perforation",
label: "Perforation" },
                              { key: "right_tinnitus",
label: "Tinnitus" },
                              { key: "right_atresia",
label: "Atresia" },
                              { key: "right_implant",
label: "Implant" },
                              { key: "right_other", label:
"Other" },
                            ].map(({ key, label }) => (
                              <div key={key}
className="flex items-center space-x-2">
                                <Checkbox
                                  id={key}
                                  checked={formData[key as
keyof typeof formData] as boolean}
                                  onCheckedChange={(checke
d) => setFormData({ ...formData, [key]: checked })}
                                />
                                <Label
htmlFor={key}>{label}</Label>
                              </div>
                            ))}
                          </div>
```

```
                    </div>
                  </div>

                  <div>
                    <Label className="mb-4"
htmlFor="medical_recommendation">Medical
Recommendation</Label>
                    <RadioGroup
                      className="grid grid-cols-2 gap-
3"
                      value={formData.medical_recommen
dation}
                      onValueChange={(value) =>
setFormData({ ...formData, medical_recommendation: value
})}
                    >
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem value="Yes"
id="medical_reco_left" />
                        <Label
htmlFor="medical_reco_left">Left</Label>
                      </div>
                      <div className="flex items-
center space-x-2">
                        <RadioGroupItem value="No"
id="medical_reco_right" />
                        <Label
htmlFor="medical_reco_right">Right</Label>
                      </div>
                    </RadioGroup>
                  </div>

                  <div>
                    <Label>Medication Given</Label>
                    <div className="grid grid-cols-2
gap-2 mt-2">
                      {medications.map((med) => (
                        <div key={med} className="flex
items-center space-x-2">
                          <Checkbox
```

```
                                id={`med_${med}`}
                                checked={formData.medicati
on_given.includes(med)}
                                onCheckedChange={(checked)
=> {
                                  if (checked) {
                                    setFormData({
...formData, medication_given:
[...formData.medication_given, med] })
                                  } else {
                                    setFormData({
                                      ...formData,
                                      medication_given:
formData.medication_given.filter((m) => m !== med),
                                    })
                                  }
                                }}
                              />
                              <Label
htmlFor={`med_${med}`}>{med}</Label>
                            </div>
                          ))}
                        </div>
                      </div>

                      <div>
                        <Label htmlFor="otoscopy_comments"
className="mb-4">Comments</Label>
                        <Textarea
                          id="otoscopy_comments"
                          value={formData.otoscopy_comment
s}
                          onChange={(e) => setFormData({
...formData, otoscopy_comments: e.target.value })}
                        />
                      </div>
                    </CardContent>
                  </Card>
                )}

                {/* 3. HEARING SCREENING */}
```

```
            <Card>
              <CardHeader>
                <CardTitle className="text-lg">3.
Hearing Screening</CardTitle>
              </CardHeader>
              <CardContent className="space-y-4">
                <div>
                  <Label className="mb-4">Screening
Method (ONLY for Walk-in patients):</Label>
                  <RadioGroup
                    className="grid grid-cols-2 gap-4
mb-6"
                    value={formData.screening_method}
                    onValueChange={(value) =>
                      setFormData({ ...formData,
screening_method: value })
                    }
                  >
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem
value="Audiogram" id="audiogram" />
                      <Label
htmlFor="audiogram">Audiogram</Label>
                    </div>
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="Voice
Test" id="voice_test" />
                      <Label htmlFor="voice_test">WFA®
Voice Test</Label>
                    </div>
                  </RadioGroup>
                </div>

                <div className="grid grid-cols-2 gap-
4">
                  <div>
                    <Label>Left Ear Result</Label>
                    <RadioGroup
```

```
                        className="grid grid-cols-2 gap-
3 mt-4"
                        value={formData.left_ear_result}
                        onValueChange={(value) =>
setFormData({ ...formData, left_ear_result: value })}
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Pass"
id="left_pass" />
                          <Label
htmlFor="left_pass">Pass</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Fail"
id="left_fail" />
                          <Label
htmlFor="left_fail">Fail</Label>
                        </div>
                      </RadioGroup>
                    </div>
                    <div>
                      <Label>Right Ear Result</Label>
                      <RadioGroup
                        className="grid grid-cols-2 gap-
4 mt-4"
                        value={formData.right_ear_result
}
                        onValueChange={(value) =>
setFormData({ ...formData, right_ear_result: value })}
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Pass"
id="right_pass" />
                          <Label
htmlFor="right_pass">Pass</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
```

```
                          <RadioGroupItem value="Fail"
id="right_fail" />
                          <Label
htmlFor="right_fail">Fail</Label>
                        </div>
                      </RadioGroup>
                    </div>
                  </div>

                  <div>
                    <Label>Satisfaction with Hearing
(18+ if passes)</Label>
                    <RadioGroup
                      className="grid grid-cols-3 gap-4
mt-4"
                      value={formData.satisfaction_with_
hearing}
                      onValueChange={(value) =>
setFormData({ ...formData, satisfaction_with_hearing:
value })}
                    >
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Unsatisfied" id="sat_unsatisfied" />
                        <Label
htmlFor="sat_unsatisfied">Unsatisfied</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Undecided" id="sat_undecided" />
                        <Label
htmlFor="sat_undecided">Undecided</Label>
                      </div>
                      <div className="flex items-center
space-x-2">
                        <RadioGroupItem
value="Satisfied" id="sat_satisfied" />
                        <Label
htmlFor="sat_satisfied">Satisfied</Label>
```

```
                    </div>
                  </RadioGroup>
                </div>
              </CardContent>
            </Card>

            {/* 4A. HEARING AID FITTING */}
            {/* wrap with ref so we can scroll/focus
when ears_clear_for_fitting === "Yes" */}
            <div ref={fittingRef}>
              <Card>
                <CardHeader className="p-4 border-b
border-gray-200">
                  <div className="flex justify-between
items-center">
                    <CardTitle className="text-lg">4A.
Hearing Aid Fitting</CardTitle>
                    <div className="flex items-center
space-x-2 text-sm">
                      <Label htmlFor="fitter_name"
className="font-bold whitespace-nowrap">Fitter
Name:</Label>
                      <Input
                        id="fitter_name"
                        ref={fitterInputRef as any}
                        value={formData.fitter_name}
                        placeholder="Fitter Name"
                        onChange={(e) => setFormData({
...formData, fitter_name: e.target.value })}
                        className="h-7 w-40 p-1
border-b border-t-0 border-l-0 border-r-0 rounded-none
focus-visible:ring-0"
                      />
                    </div>
                  </div>
                </CardHeader>

                <CardContent className="space-y-4 pt-
4">
```

```
                    {/* FIX 2: Implement the Main
Results Table (Combines 4A's old data with 4B's old
data) */}
                    <table className="w-full border-
collapse border border-gray-400 text-sm">
                      <thead className="bg-gray-100">
                        <tr className="border-b border-
gray-400">
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto text-left
w-[10%]">RESULTS</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[20%]">POWER LEVEL</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[20%]">VOLUME</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[20%]">MODEL</th>
                          <th className="font-bold
border-r border-gray-400 p-2 text-black h-auto w-
[10%]">BATTERY</th>
                          <th className="font-bold p-2
text-black h-auto w-[25%]">EARMOLD</th>
                        </tr>
                      </thead>
                      <tbody>
                        {/* LEFT EAR ROW */}
                        <tr className="border-b border-
gray-400">
                          <td className="font-bold
border-r border-gray-400 p-2">LEFT EAR</td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* LEFT POWER LEVEL Input
*/}
                            <Input
                              id="left_power_level"
                              value={formData.left_power
_level}
```

```
                                onChange={(e) =>
setFormData({ ...formData, left_power_level:
e.target.value })}
                                className="h-8 border-none
focus-visible:ring-0 text-center"
                              />
                            </td>
                            <td className="border-r
border-gray-400 p-0">
                              {/* LEFT VOLUME Input */}
                              <Input
                                id="left_volume"
                                value={formData.left_volum
e}
                                onChange={(e) =>
setFormData({ ...formData, left_volume: e.target.value
})}
                                className="h-8 border-none
focus-visible:ring-0 text-center"
                              />
                            </td>
                            <td className="border-r
border-gray-400 p-0">
                              {/* LEFT MODEL Input */}
                              <Input
                                id="left_model"
                                value={formData.left_model
}
                                onChange={(e) =>
setFormData({ ...formData, left_model: e.target.value
})}
                                className="h-8 border-none
focus-visible:ring-0 text-center"
                              />
                            </td>
                            <td className="border-r
border-gray-400 p-2">
                              {/* LEFT Battery Radio Group
*/}
                              <RadioGroup
```

```
                                value={formData.left_batte
ry_type}
                                onValueChange={(value) =>
setFormData({ ...formData, left_battery_type: value })}
                                  className="flex space-x-4
justify-center"
                                >
                                  <div className="flex
items-center space-x-1">
                                    {/* NOTE: Using
RadioGroupItem for standard component look */}
                                    <RadioGroupItem
value="13" id="left_batt_13" />
                                    <Label
htmlFor="left_batt_13" className="font-normal text-
sm">13</Label>
                                  </div>
                                  <div className="flex
items-center space-x-1">
                                    <RadioGroupItem
value="675" id="left_batt_675" />
                                    <Label
htmlFor="left_batt_675" className="font-normal text-
sm">675</Label>
                                  </div>
                                </RadioGroup>
                              </td>
                              <td className="p-0">
                                {/* LEFT EARMOLD Input */}
                                <Input
                                  id="left_earmold_type"
                                  value={formData.left_earmo
ld_type}
                                  onChange={(e) =>
setFormData({ ...formData, left_earmold_type:
e.target.value })}
                                  className="h-8 border-none
focus-visible:ring-0 text-center"
                                />
                              </td>
                            </tr>
```

```
                        {/* RIGHT EAR ROW */}
                        <tr>
                          <td className="font-bold
border-r border-gray-400 p-2">RIGHT EAR</td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* RIGHT POWER LEVEL Input
*/}
                            <Input
                              id="right_power_level"
                              value={formData.right_powe
r_level}
                              onChange={(e) =>
setFormData({ ...formData, right_power_level:
e.target.value })}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* RIGHT VOLUME Input */}
                            <Input
                              id="right_volume"
                              value={formData.right_volu
me}
                              onChange={(e) =>
setFormData({ ...formData, right_volume: e.target.value
})}
                              className="h-8 border-none
focus-visible:ring-0 text-center"
                            />
                          </td>
                          <td className="border-r
border-gray-400 p-0">
                            {/* RIGHT MODEL Input */}
                            <Input
                              id="left_model"
                              value={formData.right_mode
l}
```

```
                                onChange={(e) =>
setFormData({ ...formData, right_model: e.target.value
})}
                                className="h-8 border-none
focus-visible:ring-0 text-center"
                              />
                            </td>
                            <td className="border-r
border-gray-400 p-2">
                              {/* RIGHT Battery Radio
Group */}
                              <RadioGroup
                                value={formData.right_batt
ery_type}
                                onValueChange={(value) =>
setFormData({ ...formData, right_battery_type: value })}
                                className="flex space-x-4
justify-center"
                              >
                                <div className="flex
items-center space-x-1">
                                  <RadioGroupItem
value="13" id="right_batt_13" />
                                  <Label
htmlFor="right_batt_13" className="font-normal text-
sm">13</Label>
                                </div>
                                <div className="flex
items-center space-x-1">
                                  <RadioGroupItem
value="675" id="right_batt_675" />
                                  <Label
htmlFor="right_batt_675" className="font-normal text-
sm">675</Label>
                                </div>
                              </RadioGroup>
                            </td>
                            <td className="p-0">
                              {/* RIGHT EARMOLD Input */}
                              <Input
                                id="right_earmold_type"
```

```
                                value={formData.right_earmold_type}
                                onChange={(e) => setFormData({ ...formData, right_earmold_type: e.target.value })}
                                className="h-8 border-none focus-visible:ring-0 text-center"
                              />
                            </td>
                          </tr>
                        </tbody>
                      </table>

                      <div className="grid grid-cols-2 gap-4">
                        <div className="space-y-4 pr-2">
                          <div className="flex items-center space-x-4">
                            <Label className="font-bold text-sm whitespace-nowrap">Number of Hearing Aids Fit:</Label>
                            <RadioGroup
                              value={formData.num_hearing_aids_fit.toString()}
                              onValueChange={(value) => setFormData({ ...formData, num_hearing_aids_fit: Number.parseInt(value) })}
                              className="flex space-x-6"
                            >
                              {['0', '1', '2'].map(val => (
                                <div className="flex items-center space-x-2" key={val}>
                                  <RadioGroupItem value={val} id={`num_aids_${val}`} />
                                  <Label htmlFor={`num_aids_${val}`}>{val}</Label>
                                </div>
                              ))}
                            </RadioGroup>
                          </div>
```

```
                        {/* Special Device (Now Radio
Group) */}
                        <RadioGroup
                          value={formData.special_device
}
                          onValueChange={(value) =>
setFormData({ ...formData, special_device: value })}
                          className="space-y-1 pt-2"
                        >
                          <Label className="font-bold
block mb-1 text-sm">Special Device:</Label>
                          <div className="space-y-1 ml-
4">
                            <div className="flex items-
center space-x-2">
                              <RadioGroupItem
value="Bone Conductor" id="special_bone" />
                              <Label
htmlFor="special_bone">Bone Conductor (675
battery)</Label>
                            </div>
                            <div className="flex items-
center space-x-2">
                              <RadioGroupItem
value="Body Aid" id="special_body" />
                              <Label
htmlFor="special_body">Body Aid (AA battery)</Label>
                            </div>
                          </div>
                        </RadioGroup>

                      </div>

                      {/* RIGHT Column: Reasons for 0 or
1 aid */}
                      <div className=" p-2 bg-gray-50
rounded-md">
                        <p className="mb-2 text-xs font-
bold">
```

```
                        If patient received 1 or 0
hearing aids, select option below:
                      </p>

                      <div className="flex flex-col
space-y-1">
                        {/* Header for LEFT and RIGHT
columns */}
                        <div className="flex justify-
end space-x-8 pr-1 pt-1 font-bold text-xs">
                          <Label>LEFT</Label>
                          <Label>RIGHT</Label>
                        </div>

                        {/* Mapping the reasons for
fewer aids */}
                        {reasonsForFewerAids.map((reas
on, index) => (
                          <div
                            key={index}
                            className="flex justify-
between items-center text-xs py-1" // Added items-center
for vertical alignment
                          >
                            {/* Reason label */}
                            <Label className="font-
normal w-3/5 whitespace-nowrap">
                              {reason}
                            </Label>

                            {/* LEFT and RIGHT
checkboxes - MODIFIED FOR CENTERING */}
                            {/* This container now
controls the space for the two checkbox columns */}
                            <div className="flex w-2/5
justify-around pr-2">
                              {/* LEFT checkbox
wrapper - Centers the checkbox */}
                              <div className="flex
justify-center w-1/2">
                                <Checkbox
```

```
                                    id={`reason_left_${i
ndex}`}
                                    checked={formData.re
asons?.left?.includes(reason) || false}
                                    onCheckedChange={(ch
ecked) => {
                                      const updatedLeft
= checked
                                        ?
[...(formData.reasons?.left || []), reason]
                                        :
(formData.reasons?.left || []).filter((r) => r !==
reason);
                                      setFormData({
                                        ...formData,
                                        reasons: {
                                          ...formData.re
asons,
                                          left:
updatedLeft,
                                        },
                                      });
                                    }}
                                  />
                                </div>

                                {/* RIGHT checkbox
wrapper - Centers the checkbox */}
                                <div className="flex
justify-center w-1/2">
                                  <Checkbox
                                    id={`reason_right_${
index}`}
                                    checked={formData.re
asons?.right?.includes(reason) || false}
                                    onCheckedChange={(ch
ecked) => {
                                      const updatedRight
= checked
                                        ?
[...(formData.reasons?.right || []), reason]
```

```
                                        :
(formData.reasons?.right || []).filter((r) => r !==
reason);
                                    setFormData({
                                      ...formData,
                                      reasons: {
                                        ...formData.re
asons,
                                        right:
updatedRight,
                                      },
                                    });
                                  }}
                                />
                              </div>
                            </div>
                          </div>
                        ))}
                      </div>
                    </div>

                  </div>

                  {/* Comments Section (Moved from old
bottom of 4A) */}
                  <div className="pt-2">
                    <Label htmlFor="fitting_comments"
className="font-bold">Comments:</Label>
                    <Textarea
                      id="fitting_comments"
                      value={formData.fitting_comments
}
                      onChange={(e) => setFormData({
...formData, fitting_comments: e.target.value })}
                      className="min-h-[60px]"
                    />
                  </div>
                  <Separator className="w-full" />
                  <CardTitle className="text-lg">4B.
Fitting Quality Control</CardTitle>
                  <CardContent className="space-y-4">
```

```
                    <div>
                      <Label className="font-medium
mb-2">
                        <strong>Patient clear for
counseling:</strong>
                      </Label>
                      <RadioGroup
                        className="grid grid-cols-2
gap-4"
                        value={formData.patient_clear_
for_counseling ? "Yes" : "No"}
                        onValueChange={(value) =>
                          setFormData({ ...formData,
patient_clear_for_counseling: value === "Yes" })
                        }
                      >
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="Yes"
id="counseling_yes" />
                          <Label
htmlFor="counseling_yes">Yes</Label>
                        </div>
                        <div className="flex items-
center space-x-2">
                          <RadioGroupItem value="No"
id="counseling_no" />
                          <Label
htmlFor="counseling_no">No</Label>
                        </div>
                      </RadioGroup>
                    </div>

                  </CardContent>
                </CardContent>
              </Card>
            </div>
          {/* 5. COUNSELING */}
          <Card>
            <CardHeader>
```

```
                <CardTitle className="text-lg">5.
Counseling</CardTitle>
              </CardHeader>
              <CardContent className="space-y-4">
                <div className="flex items-center space-
x-2">
                  <Checkbox
                    id="completed_counseling"
                    checked={formData.completed_counseli
ng}
                    onCheckedChange={(checked) =>
                      setFormData({ ...formData,
completed_counseling: checked as boolean })
                    }
                  />
                  <Label htmlFor="completed_counseling">
                    Patient completed counseling and
received AfterCare information
                  </Label>
                </div>
                <div className="flex items-center space-
x-2">
                  <Checkbox
                    id="trained_as_student_ambassador"
                    checked={formData.trained_as_student
_ambassador}
                    onCheckedChange={(checked) =>
                      setFormData({ ...formData,
trained_as_student_ambassador: checked as boolean })
                    }
                  />
                  <Label
htmlFor="trained_as_student_ambassador">Patient trained
as a<strong>Student Ambassador</strong></Label>
                </div>
              </CardContent>
            </Card>

            {/* 6. FINAL QUALITY CONTROL */}
            <Card>
              <CardHeader>
```

```
                <CardTitle className="text-lg">6. Final
Quality Control</CardTitle>
              </CardHeader>
              <CardContent className="space-y-4">
                <div className="grid grid-cols-2 gap-4">
                  <div>
                    <Label htmlFor="batteries_13">Number
of batteries provided (13)</Label>
                    <Input
                      id="batteries_13"
                      type="number"
                      placeholder="0"
                      value={formData.batteries_13}
                      onChange={(e) => setFormData({
...formData, batteries_13:
Number.parseInt(e.target.value) })}
                    />
                  </div>
                  <div>
                    <Label
htmlFor="batteries_675">Number of batteries provided
(675)</Label>
                    <Input
                      id="batteries_675"
                      type="number"
                      placeholder="0"
                      value={formData.batteries_675}
                      onChange={(e) =>
                        setFormData({ ...formData,
batteries_675: Number.parseInt(e.target.value) })
                      }
                    />
                  </div>
                </div>

                <div>
                  <Label className="mb-2"><strong>(For
patients 18 & older)</strong></Label>
                  <Label>When wearing your hearing
aid(s) are you
                    satisfied with your hearing?</Label>
```

```
                  <RadioGroup
                    className="grid grid-cols-3 gap-4
mt-4"
                    value={formData.final_satisfaction}
                    onValueChange={(value) =>
setFormData({ ...formData, final_satisfaction: value })}
                  >
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem
value="Unsatisfied" id="final_unsatisfied" />
                      <Label
htmlFor="final_unsatisfied">Unsatisfied</Label>
                    </div>
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="Undecided"
id="final_undecided" />
                      <Label
htmlFor="final_undecided">Undecided</Label>
                    </div>
                    <div className="flex items-center
space-x-2">
                      <RadioGroupItem value="Satisfied"
id="final_satisfied" />
                      <Label
htmlFor="final_satisfied">Satisfied</Label>
                    </div>
                  </RadioGroup>
                </div>

                <div>
                  <Label
htmlFor="final_comments">Comments</Label>
                  <Textarea
                    id="final_comments"
                    value={formData.final_comments}
                    onChange={(e) => setFormData({
...formData, final_comments: e.target.value })}
                  />
                </div>
```

```
            </CardContent>
          </Card>
          </form>
        </ScrollArea>
        <div className="flex justify-end space-x-1 pt-
1">
          <Button type="button" variant="outline"
onClick={requestClose}>
            Cancel
          </Button>
          <Button type="submit"
onClick={handleSubmit}>Submit Phase 2 Form</Button>
        </div>
       </DialogContent>
      </Dialog>

      {/* Confirmation dialog to avoid accidental close
*/}
      <Dialog open={confirmCloseOpen} onOpenChange={(v)
=> setConfirmCloseOpen(v)}>
       <DialogContent className="max-w-md">
        <DialogHeader>
          <DialogTitle>Discard changes?</DialogTitle>
        </DialogHeader>
        <div className="py-4">
          <p className="mb-4">You have unsaved
changes. Are you sure you want to close and discard
them?</p>
          <div className="flex justify-end gap-2">
            <Button variant="outline" onClick={() =>
setConfirmCloseOpen(false)}>Keep editing</Button>
            <Button
onClick={handleConfirmClose}>Discard & Close</Button>
          </div>
        </div>
       </DialogContent>
      </Dialog>

      {/* Submit confirmation: recheck patient data
before proceeding */}
```

```
      <Dialog open={submitConfirmOpen} onOpenChange={(v)
=> setSubmitConfirmOpen(v)}>
        <DialogContent className="max-w-md">
          <div className="py-4">
            <h3 className="text-lg font-semibold mb-
2">Please recheck patient data</h3>
            <p className="mb-4">Before submitting,
please confirm you have reviewed all patient data.
Proceed?</p>
            <div className="flex justify-end gap-2">
              <Button variant="outline" onClick={() =>
setSubmitConfirmOpen(false)}>Cancel</Button>
              <Button
onClick={handleConfirmSubmit}>Confirm & Submit</Button>
            </div>
          </div>
        </DialogContent>
      </Dialog>
   </>
 )
}
```