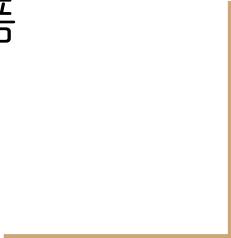


Web Programming

설문조사 웹폼



탁제형 2019975066

Contents

1. Base Source
 2. Upgrade Source
-

Base Source

웹폼 설문조사
Made by Lovable

고객 만족도 설문조사

여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며, 서비스 개선에 활용됩니다.

성함을 알려주세요 *

답변을 입력하세요

전반적인 만족도를 평가해주세요 *

- 매우 만족
- 만족
- 보통
- 불만족
- 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)

- 직관적인 UI/UX
- 빠른 로딩 속도
- 다양한 기능
- 모바일 지원
- 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

답변을 입력하세요

설문 제출하기

설문 작성 시간: 약 3-5분 소요

Base Source

Prompt: 설문조사를 할 수 있는 웹폼을 만들어줘.

설문조사 웹폼을 만들어드리겠습니다.

****디자인 컨셉:****

- Google Forms/Typeform 스타일의 모던하고 깔끔한 디자인
- 청록색/파란색 계열의 신뢰감 있는 색상
- 카드 기반 레이아웃과 부드러운 애니메이션

****구현 기능:****

- 다양한 질문 유형 (객관식, 체크박스, 단답형, 서술형)
- 반응형 디자인
- 폼 유효성 검사
- 제출 완료 메시지

****디자인 요소:****

- Primary: 청록색 그라데이션 ($180\deg$)
- 부드러운 카드 셰도우와 호버 효과
- 낙제한 여백과 가독성 좋은 타이포그래피

The screenshot shows a survey form template with the following sections:

- Header:** 고객 만족도 설문조사
- Text:** 여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며, 서비스 개선에 활용됩니다.
- Section 1: 성함을 알려주세요 ***
Input field: 답변을 입력하세요
- Section 2: 전반적인 만족도를 평가해주세요 ***
Rating scale: 매우 만족 (5) → 만족 (4) → 보통 (3) → 불만족 (2) → 매우 불만족 (1)
- Section 3: 어떤 기능이 가장 유용했나요? (복수 선택 가능)**
Multiple choice: 직관적인 UI/UX, 빠른 응답 속도, 다양한 기능, 모바일 지원, 고객 지원
- Section 4: 개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요**
Input field: 답변을 입력하세요
- Footer:** 설문 제출하기
- 설문 작성 시간: 약 3-5분 소요

Base Source

```

1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5  /* Definition of the design system. All colors, gradients, fonts, etc should be defined here.
6  All colors MUST be HSL.
7 */
8
9  @layer base {
10    :root {
11      --background: 210 40% 98%;
12      --foreground: 222 47% 11%;
13
14      --card: 0 0% 100%;
15      --card-foreground: 222 47% 11%;
16
17      --popover: 0 0% 100%;
18      --popover-foreground: 222 47% 11%;
19
20      --primary: 192 88% 45%;
21      --primary-foreground: 0 0% 100%;
22      --primary-glow: 192 88% 55%;
23
24      --secondary: 192 20% 95%;
25      --secondary-foreground: 192 88% 25%;
26
27      --muted: 210 30% 94%;
28      --muted-foreground: 215 16% 47%;
29
30      --accent: 192 70% 50%;
31      --accent-foreground: 0 0% 100%;
32
33      --destructive: 0 84% 60%;
34      --destructive-foreground: 0 0% 100%;
35
36      --border: 214 32% 91%;
37      --Input: 214 32% 91%;
38      --ring: 192 88% 45%;
39
40      --radius: 0.75rem;
41    }
42
43    --gradient-primary: linear-gradient(135deg, hsl(192 88% 45%), hsl(192 88% 55%));
44    --gradient-card: linear-gradient(180deg, hsl(0 0% 100%), hsl(210 40% 99%));
45    --shadow-card: 0 2px 8px -2px hsl(192 88% 45% / 0.08);
46    --shadow-card-hover: 0 8px 24px -4px hsl(192 88% 45% / 0.12);
47    --transition-smooth: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
48
49    --sidebar-background: 0 0% 98%;
50
51    --sidebar-foreground: 240 5.3% 26.1%;
52
53    --sidebar-primary: 240 5.9% 10%;
54
55    --sidebar-primary-foreground: 0 0% 98%;
56
57    --sidebar-accent: 240 4.8% 95.9%;
58
59    --sidebar-accent-foreground: 240 5.9% 10%;
60
61    --sidebar-border: 220 13% 91%;
62
63    --sidebar-ring: 217.2 91.2% 59.8%;
64
65    .dark {
66      --background: 222 47% 11%;
67      --foreground: 210 40% 98%;
68
69      --card: 222 40% 13%;
70      --card-foreground: 210 40% 98%;
71
72      --popover: 222 40% 13%;
73      --popover-foreground: 210 40% 98%;
74
75      --primary: 192 88% 50%;
76      --primary-foreground: 222 47% 11%;
77      --primary-glow: 192 88% 60%;
78
79      --secondary: 192 20% 28%;
80      --secondary-foreground: 192 88% 70%;
81    }
82
83    --muted: 217 33% 18%;
84    --muted-foreground: 215 20% 65%;
85
86    --accent: 192 70% 45%;
87    --accent-foreground: 210 40% 98%;
88
89    --destructive: 0 63% 31%;
90    --destructive-foreground: 210 40% 98%;
91
92    --border: 217 33% 18%;
93    --Input: 217 33% 18%;
94    --ring: 192 88% 50%;
95
96    --gradient-primary: linear-gradient(135deg, hsl(192 88% 45%), hsl(192 88% 55%));
97    --gradient-card: linear-gradient(180deg, hsl(222 40% 13%), hsl(222 40% 14%));
98    --shadow-card: 0 2px 8px -2px hsl(0 0% 6 / 0.3);
99    --shadow-card-hover: 0 8px 24px -4px hsl(192 88% 50% / 0.2);
100   --sidebar-background: 240 4.3% 95.9%;
101   --sidebar-primary: 224.3 76.3% 48%;
102   --sidebar-primary-foreground: 0 0% 100%;
103   --sidebar-accent: 240 3.7% 15.95;
104   --sidebar-accent-foreground: 240 4.8% 95.9%;
105   --sidebar-border: 240 3.7% 15.95;
106   --sidebar-ring: 217.2 91.2% 59.8%;
107
108  }
109
110  @layer base {
111    *
112    | @apply border-border;
113  }
114
115  body {
116    @apply bg-background text-foreground;
117  }
118

```

Base Source

```
1  <!doctype html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>설문조사 | 간편한 온라인 설문 플랫폼</title>
7      <meta name="description" content="쉽고 빠른 온라인 설문조사 플랫폼. 다양한 질문 유형으로 의견을 수집하고 분석하세요." />
8      <meta name="author" content="Survey Platform" />
9
10     <meta property="og:title" content="설문조사 | 간편한 온라인 설문 플랫폼" />
11     <meta property="og:description" content="쉽고 빠른 온라인 설문조사 플랫폼. 다양한 질문 유형으로 의견을 수집하고 분석하세요." />
12     <meta property="og:type" content="website" />
13     <meta property="og:image" content="https://lovable.dev/opengraph-image-p98pqg.png" />
14
15     <meta name="twitter:card" content="summary_large_image" />
16     <meta name="twitter:site" content="@lovable_dev" />
17     <meta name="twitter:image" content="https://lovable.dev/opengraph-image-p98pqg.png" />
18   </head>
19
20   <body>
21     <div id="root"></div>
22     <script type="module" src="/src/main.tsx"></script>
23   </body>
24 </html>
```

Base Source

```
1 import { SurveyForm } from "@/components/SurveyForm";
2
3 const Index = () => {
4   return <SurveyForm />;
5 };
6
7 export default Index;
```

Base Source

```
1 import { Card } from "@/components/ui/card";
2 import { Input } from "@/components/ui/input";
3 import { Textarea } from "@/components/ui/textarea";
4 import { RadioGroup, RadioGroupItem } from "@/components/ui/radio-group";
5 import { Checkbox } from "@/components/ui/checkbox";
6 import { Label } from "@/components/ui/label";
7 import { cn } from "@/lib/utils";
8
9 interface QuestionOption {
10   id: string;
11   label: string;
12 }
13
14 interface QuestionCardProps {
15   question: string;
16   type: "text" | "textarea" | "radio" | "checkbox";
17   options?: QuestionOption[];
18   required?: boolean;
19   value?: string | string[];
20   onChange?: (value: string | string[]) => void;
21   error?: string;
22 }
23
24 export const QuestionCard = ({  
25   question,  
26   type,  
27   options,  
28   required = false,  
29   value,  
30   onChange,  
31   error,  
32 }: QuestionCardProps) => {  
33   const handleCheckboxChange = (optionId: string, checked: boolean) => {  
34     const currentValues = (value as string[]) || [];
35     const newValues = checked  
      ? [...currentValues, optionId]
36      : currentValues.filter((v) => v !== optionId);
37     onChange?.(newValues);
38   };
39 }
40
41 return [  
42   Card.className`-5 transition-all duration-300 hover:shadow-[inset -2px 0 0 0 1px #e0e0e0] lg:gradient-to-l1-front-to-l2 border-[1px solid #e0e0e0] rounded-[10px] p-5`,  
43   div.className`space-y-4`  
44   div.className`text-lg font-semibold text-card-primary`  
45   (question)  
46   (required ? span.className`text-destructive italic` : span)  
47   (label)  
48
49   (type === "text" ? (  
50     input  
51     value={value as string} || ""  
52     onChange={(e) => onChange?.(e.target.value)}  
53     placeholder="Type something"  
54     className="w-full h-[40px]"  
55     "transition-all duration-300"  
56     error ? "border-destructive focus-visible-ring-destructive"  
57   ) :  
58   (type === "textarea" ? (  
59     textarea  
60     value={value as string} || ""  
61     onChange={(e) => onChange?.(e.target.value)}  
62     placeholder="Type something"  
63     rows={4}  
64     className="w-full h-[100px]"  
65     "transition-all duration-300 resize-none"  
66     error ? "border-destructive focus-visible-ring-destructive"  
67   ) :  
68   (type === "radio" ? options.map((option) => (  
69     radioGroup  
70     value={value as string} || ""  
71     onChange={(val) => onChange?.(val)}  
72   )) : null)  
73 );
74 
```

```
        <div className="space-y-3">
          {options.map((option) => (
            <div key={option.id} className="flex items-center space-x-3">
              <RadioGroupItem
                value={option.id}
                id={option.id}
                className="transition-all duration-200"
              />
              <Label
                htmlFor={option.id}
                className="text-card-foreground cursor-pointer font-normal"
              >
                {option.label}
              </Label>
            </div>
          )));
        </div>
      </RadioGroup>
    )
  }

  {type === "checkbox" && options &> (
    <div className="space-y-3">
      {options.map((option) => (
        <div key={option.id} className="flex items-center space-x-3">
          <Checkbox
            id={option.id}
            checked={(value as string[]).includes(option.id)}
            onChange={(checked) =>
              handleCheckboxChange(option.id, checked as boolean)
            }
            className="transition-all duration-200"
          />
          <Label
            htmlFor={option.id}
            className="text-card-foreground cursor-pointer font-normal"
          >
            {option.label}
          </Label>
        </div>
      )));
    </div>
  )};

```

QuestionCard.tsx

Base Source

```
1 import { useState } from "react";
2 import { Button } from "@/components/ui/button";
3 import { QuestionCard } from "./QuestionCard";
4 import { toast } from "@/hooks/use-toast";
5 import { CheckCircle2 } from "lucide-react";
6
7 interface Question {
8   id: string;
9   question: string;
10  type: "text" | "textarea" | "radio" | "checkbox";
11  options?: { id: string; label: string }[];
12  required?: boolean;
13 }
14
15 const surveyQuestions: Question[] = [
16 {
17   id: "name",
18   question: "성함을 알려주세요",
19   type: "text",
20   required: true,
21 },
22 {
23   id: "satisfaction",
24   question: "전반적인 만족도를 평가해주세요",
25   type: "radio",
26   options: [
27     { id: "very-satisfied", label: "매우 만족" },
28     { id: "satisfied", label: "만족" },
29     { id: "neutral", label: "보통" },
30     { id: "dissatisfied", label: "불만족" },
31     { id: "very-dissatisfied", label: "매우 불만족" },
32   ],
33   required: true,
34 },
35   {
36     id: "features",
37     question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)",
38     type: "checkbox",
39     options: [
40       { id: "ui-design", label: "직관적인 UI/UX" },
41       { id: "speed", label: "빠른 로딩 속도" },
42       { id: "features", label: "다양한 기능" },
43       { id: "mobile", label: "모바일 지원" },
44       { id: "customer-service", label: "고객 지원" },
45     ],
46     required: false,
47   },
48   {
49     id: "feedback",
50     question: "개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요",
51     type: "textarea",
52     required: false,
53   },
54 ];
55
56 export const SurveyForm = () => {
57   const [answers, setAnswers] = useState<Record<string, string | string[]>|{}>();
58   const [errors, setErrors] = useState<Record<string, string>|{}>();
59   const [isSubmitted, setIsSubmitted] = useState(false);
60
61   const handleAnswerChange = (questionId: string, value: string | string[]) => {
62     setAnswers((prev) => ({ ...prev, [questionId]: value }));
63     // Clear error when user starts typing
64     if (errors[questionId]) {
65       setErrors((prev) => {
66         const newErrors = { ...prev };
67         delete newErrors[questionId];
68         return newErrors;
69       });
70     }
71   };
72 }
```

```
73 const validateForm = (): boolean => {
74   const newErrors: Record<string, string> = {};
75
76   surveyQuestions.forEach((question) => {
77     if (question.required) {
78       const answer = answers[question.id];
79       if (answer || (Array.isArray(answer) && answer.length === 0)) {
80         newErrors[question.id] = "필수 항목입니다";
81       } else if (typeof answer === "string" && answer.trim() === "") {
82         newErrors[question.id] = "답변을 입력해주세요";
83       }
84     }
85   });
86
87   setErrors(newErrors);
88   return Object.keys(newErrors).length === 0;
89 }
90
91 const handleSubmit = (e: React.FormEvent) => {
92   e.preventDefault();
93
94   if (!validateForm()) {
95     toast({
96       title: "입력 확인",
97       description: "필수 항목을 모두 입력해주세요.",
98       variant: "destructive",
99     });
100   }
101
102   // Here you would typically send the data to a backend
103   console.log("Survey answers:", answers);
104
105   setIsSubmitted(true);
106   toast({
107     title: "제출 완료",
108     description: "소중한 의견 감사합니다!",
109   });
110 }
111 }
```

SurveyForm.tsx

Base Source

```
113 if (isSubmitted) {
114   return (
115     <div className="min-h-screen flex items-center justify-center p-4 bg-background">
116       <div className="max-w-2xl w-full text-center space-y-6 animate-in-fade-in duration-500">
117         <div className="flex justify-center">
118           <div className="relative">
119             <div className="absolute inset-0 bg-primary/20 blur-xl rounded-full animate-pulse" />
120             <CheckCircle2 className="w-24 h-24 text-primary relative animate-in-zoom-in duration-300" />
121           </div>
122         </div>
123         <div className="space-y-2">
124           <div className="text-3xl font-bold text-foreground">
125             김(행)님!
126           </div>
127           <div className="text-lg text-muted-foreground">
128             좋은 의견은 더 나은 서비스를 만드는 데 활용하겠습니다.
129           <br />
130           소중한 의견은 더 나은 서비스를 만드는 데 활용하겠습니다.
131         </div>
132         <div>
133           <button
134             onClick={() => {
135               setIsSubmitted(false);
136               setAnswers([]);
137               setErrors([]);
138             }}
139             variant="outline"
140             size="lg"
141             className="transition-all duration-200"
142           >
143             새 설문 작성하기
144           </button>
145         </div>
146       </div>
147     );
148   )
149 }
```

```
return (
  <div className="min-h-screen py-12 px-4 lg:background">
    <div className="max-w-3xl mx-auto space-y-8 animate-in duration-300">
      <header className="text-center space-y-4">
        <div className="white-block">
          <div className="text-4xl font-bold bg-gradient-to-r from-primary to-primary-glow bg-clip-text text-transparent">
            고객 헬스 설문조사
          </div>
        </div>
        <div className="h-1 bg-gradient-to-r from-primary to-primary-glow rounded-full mt-2" />
      </div>
      <p className="text-lg text-muted-foreground max-w-2xl mx-auto">
        여러분의 소중한 의견을 들려주세요. 모든 답변은 암명으로 처리되며,
        서비스 개선에 활용됩니다.
      </p>
    </header>

    <form onSubmit={handleSubmit} className="space-y-6">
      {surveyQuestions.map((question, index) => (
        <div key={question.id}>
          <QuestionCard
            question={question.question}
            type={question.type}
            options={question.options}
            required={question.required}
            value={answers[question.id]}
            onChange={(value) => handleAnswerChange(question.id, value)}
            error={errors[question.id]}
          />
        </div>
      )));
    </form>
  </div>
  <div className="animate-in slide-in-from-bottom duration-300" style={{ animationDelay: `${index * 100}ms` }}>
    <div className="pt-4">
      <Button type="submit" size="lg" className="w-full bg-gradient-to-r from-primary to-primary-glow hover:opacity-90 transition-all duration-300 text-primary-foreground font-semibold shadow-lg hover:shadow-xl">
        설문 제출하기
      </Button>
    </div>
  </div>
</div>
<Footer className="text-center text-sm text-muted-foreground pt-8 border-t">
  <p>설문 작성 시간: 약 3-5분 소요</p>
</Footer>
</div>
</div>
);
}
```

SurveyForm.tsx

Upgrade Source

<https://preview--web-poll-forge.lovable.app/>

고객 만족도 설문조사

소중한 의견 감사합니다.

성별을 알려주세요 *

성별을 선택해주세요 *

남성
 여성

나이대를 선택해주세요 *

10대
 20대
 30대
 40대
 50대
 60대 이상

친환경적 인증도를 평가해주세요 *

매우 낮음
 낮음
 보통
 높음
 매우 높음

어떤 기능이 가장 유용했나요? (복수 선택 가능)

직관적인 인터페이스
 빠른 응답 속도
 다양한 기능
 모바일 지원
 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

설문 제출하기

설문 제출하기

설문 제출하기

Upgrade Source I

Prompt: 디자인 컨셉을 블랙/골드 계열의 색상으로 변경해줘

고객 만족도 설문조사

여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며, 서비스 개선에 활용됩니다.

성함을 알려주세요 *

답변을 입력하세요

전반적인 만족도를 평가해주세요 *

- 매우 만족
- 만족
- 보통
- 불만족
- 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)

- 직관적인 UI/UX
- 빠른 로딩 속도
- 다양한 기능
- 모바일 지원
- 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

답변을 입력하세요

설문 제출하기

설문 작성 시간: 약 3-5분 소요

Upgrade Source I

```

1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5  /* Definition of the design system. All colors, gradients, fonts, etc should be defined here.
6  All colors MUST be HSL.
7 */
8
9  @layer base {
10  :root {
11    --background: 0 0% 98%;
12    --foreground: 0 0% 10%;
13
14    --card: 0 0% 100%;
15    --card-foreground: 0 0% 10%;
16
17    --popover: 0 0% 100%;
18    --popover-foreground: 0 0% 10%;
19
20    --primary: 45 90% 50%;
21    --primary-foreground: 0 0% 10%;
22    --primary-glow: 45 95% 60%;
23
24    --secondary: 45 15% 90%;
25    --secondary-foreground: 45 80% 30%;
26
27    --muted: 0 0% 92%;
28    --muted-foreground: 0 0% 45%;
29
30    --accent: 45 85% 55%;
31    --accent-foreground: 0 0% 10%;
32
33    --destructive: 0 84% 60%;
34    --destructive-foreground: 0 0% 100%;
35
36    --border: 0 0% 88%;
37    --input: 0 0% 88%;
38    --ring: 45 90% 50%;
39
40    --radius: 0.75rem;
41  }
42
43  --gradient-primary: linear-gradient(135deg, □hsl(45 90% 50%), □hsl(45 95% 60%));
44  --gradient-card: linear-gradient(180deg, □hsl(0 0% 100%), □hsl(0 0% 98%));
45  --shadow-card: 0 2px 8px -2px □hsl(45 90% 50% / 0.15);
46  --shadow-card-hover: 0 8px 24px -4px □hsl(45 90% 50% / 0.25);
47  --transition-smooth: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
48
49  --sidebar-background: 0 0% 98%;
50
51  --sidebar-foreground: 240 5.3% 26.1%;
52
53  --sidebar-primary: 240 5.9% 10%;
54
55  --sidebar-primary-foreground: 0 0% 98%;
56
57  --sidebar-accent: 240 4.8% 95.9%;
58
59  --sidebar-accent-foreground: 240 5.9% 10%;
60
61  --sidebar-border: 220 13% 91%;
62
63  --sidebar-ring: 217.2 91.2% 59.8%;
64
65  .dark {
66    --background: 0 0% 8%;
67    --foreground: 45 95% 90%;
68
69    --card: 0 0% 12%;
70    --card-foreground: 45 95% 90%;
71
72    --popover: 0 0% 12%;
73    --popover-foreground: 45 95% 90%;
74
75    --primary: 45 95% 55%;
76    --primary-foreground: 0 0% 8%;
77    --primary-glow: 45 100% 65%;
78
79    --secondary: 45 25% 18%;
80    --secondary-foreground: 45 95% 75%;
81
82
83  --muted: 0 0% 18%;
84  --muted-foreground: 0 0% 60%;
85
86  --accent: 45 90% 50%;
87  --accent-foreground: 0 0% 8%;
88
89  --destructive: 0 63% 31%;
90  --destructive-foreground: 45 95% 90%;
91
92  --border: 0 0% 20%;
93  --input: 0 0% 20%;
94  --ring: 45 95% 55%;
95
96  --gradient-primary: linear-gradient(135deg, □hsl(45 90% 50%), □hsl(45 95% 60%));
97  --gradient-card: linear-gradient(180deg, □hsl(0 0% 12%), □hsl(0 0% 14%));
98  --shadow-card: 0 2px 8px -2px □hsl(0 0% 0% / 0.5);
99  --shadow-card-hover: 0 8px 24px -4px □hsl(45 95% 55% / 0.3);
100 --sidebar-background: 240 5.9% 10%;
101 --sidebar-foreground: 240 4.8% 95.9%;
102 --sidebar-primary: 224.3 76.3% 48%;
103 --sidebar-primary-foreground: 0 0% 100%;
104 --sidebar-accent: 240 3.7% 15.9%;
105 --sidebar-accent-foreground: 240 4.8% 95.9%;
106 --sidebar-border: 240 3.7% 15.9%;
107 --sidebar-ring: 217.2 91.2% 59.8%;
108 }
109
110 @layer base {
111   * {
112     @apply border-border;
113   }
114
115   body {
116     @apply bg-background text-foreground;
117   }
118 }

```

Upgrade Source II

Prompt: 바탕도 블랙으로 변경하고 카드레이아웃을
다크그레이로 변경해줘

고객 만족도 설문조사

여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며, 서비스 개선
에 활용됩니다.

성함을 알려주세요 *

진반적인 만족도를 평가해주세요 *

- 매우 만족
- 만족
- 보통
- 불만족
- 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)

- 직관적인 UI/UX
- 빠른 로딩 속도
- 다양한 기능
- 모바일 지원
- 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

설문 제출하기

설문 작성 시간: 약 3-5분 소요

Upgrade Source II

```

1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5  /* Definition of the design system. All colors, gradients, fonts, etc should be defined here.
6  All colors MUST be HSL.
7 */
8
9  @layer base {
10  :root {
11    --background: 0 0% 8%;
12    --foreground: 45 95% 90%;
13
14    --card: 0 0% 15%;
15    --card-foreground: 45 95% 90%;
16
17    --popover: 0 0% 100%;
18    --popover-foreground: 0 0% 10%;
19
20    --primary: 45 90% 50%;
21    --primary-foreground: 0 0% 10%;
22    --primary-glow: 45 95% 60%;
23
24    --secondary: 45 15% 90%;
25    --secondary-foreground: 45 80% 30%;
26
27    --muted: 0 0% 92%;
28    --muted-foreground: 0 0% 45%;
29
30    --accent: 45 85% 55%;
31    --accent-foreground: 0 0% 10%;
32
33    --destructive: 0 84% 60%;
34    --destructive-foreground: 0 0% 100%;
35
36    --border: 0 0% 88%;
37    --input: 0 0% 88%;
38    --ring: 45 90% 50%;
39
40    --radius: 0.75rem;
41
42    --gradient-primary: linear-gradient(135deg, □ hsl(45 90% 50%), □ hsl(45 95% 60%));
43    --gradient-card: linear-gradient(180deg, □ hsl(0 0% 15%), □ hsl(0 0% 17%));
44    --shadow-card: 0 2px 8px -2px □ hsl(45 90% 50% / 0.15);
45    --shadow-card-hover: 0 8px 24px -4px □ hsl(45 90% 50% / 0.25);
46    --transition-smooth: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
47
48    --sidebar-background: 0 0% 98%;
49
50    --sidebar-foreground: 240 5.3% 26.1%;
51
52    --sidebar-primary: 240 5.9% 10%;
53
54    --sidebar-primary-foreground: 0 0% 98%;
55
56    --sidebar-accent: 240 4.8% 95.9%;
57
58    --sidebar-accent-foreground: 240 5.9% 10%;
59
60    --sidebar-border: 220 13% 91%;
61
62    --sidebar-ring: 217.2 91.2% 59.8%;
63  }
64
65  .dark {
66    --background: 0 0% 8%;
67    --foreground: 45 95% 90%;
68
69    --card: 0 0% 12%;
70    --card-foreground: 45 95% 90%;
71
72    --popover: 0 0% 12%;
73    --popover-foreground: 45 95% 90%;
74
75    --primary: 45 95% 55%;
76    --primary-foreground: 0 0% 8%;
77    --primary-glow: 45 108% 65%;
78
79    --secondary: 45 25% 18%;
80    --secondary-foreground: 45 95% 75%;
81
82    --muted: 0 0% 18%;
83    --muted-foreground: 0 0% 60%;
84
85    --accent: 45 90% 50%;
86    --accent-foreground: 0 0% 8%;
87
88    --destructive: 0 63% 31%;
89    --destructive-foreground: 45 95% 90%;
90
91    --border: 0 0% 20%;
92    --input: 0 0% 20%;
93    --ring: 45 95% 55%;
94
95    --gradient-primary: linear-gradient(135deg, □ hsl(45 90% 50%), □ hsl(45 95% 60%));
96    --gradient-card: linear-gradient(180deg, □ hsl(0 0% 12%), □ hsl(0 0% 14%));
97    --shadow-card: 0 2px 8px -2px □ hsl(0 0% 8% / 0.5);
98    --shadow-card-hover: 0 8px 24px -4px □ hsl(45 95% 55% / 0.3);
99    --sidebar-background: 240 5.9% 10%;
100   --sidebar-foreground: 240 4.8% 95.9%;
101   --sidebar-primary: 224.3 76.3% 48%;
102   --sidebar-primary-foreground: 0 0% 100%;
103   --sidebar-accent: 240 3.7% 15.9%;
104   --sidebar-accent-foreground: 240 4.8% 95.9%;
105   --sidebar-border: 240 3.7% 15.9%;
106   --sidebar-ring: 217.2 91.2% 59.8%;
107
108  }
109
110  @layer base {
111  * {
112    @apply border-border;
113  }
114
115  body {
116    @apply bg-background text-foreground;
117  }
118

```

Upgrade Source III

Prompt: 전체 폰트를 Economica 폰트로 변경 해줘

고객 만족도 설문조사

여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며, 서비스 개선에 활용됩니다.

성함을 알려주세요 *

답변을 입력하세요

전반적인 만족도를 평가해주세요 *

매우 만족
 만족
 보통
 불만족
 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)

직관적인 UI/UX
 빠른 로딩 속도
 다양한 기능
 모바일 지원
 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

답변을 입력하세요

설문 제출하기

설문 작성 시간: 약 5~10분 소요

Upgrade Source III

```

1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5  /* Definition of the design system. All colors, gradients, fonts, etc should be defined here.
6  All colors MUST be HSL.
7 */
8
9  @layer base {
10    :root {
11      --background: 0 0% 8%;
12      --foreground: 45 95% 98%;
13
14      --card: 0 0% 15%;
15      --card-foreground: 45 95% 98%;
16
17      --popover: 0 0% 100%;
18      --popover-foreground: 0 0% 10%;
19
20      --primary: 45 98% 50%;
21      --primary-foreground: 0 0% 10%;
22      --primary-glow: 45 95% 60%;
23
24      --secondary: 45 15% 98%;
25      --secondary-foreground: 45 80% 30%;
26
27      --muted: 0 0% 92%;
28      --muted-foreground: 0 0% 45%;
29
30      --accent: 45 85% 55%;
31      --accent-foreground: 0 0% 10%;
32
33      --destructive: 0 84% 60%;
34      --destructive-foreground: 0 0% 100%;
35
36      --border: 0 0% 88%;
37      --input: 0 0% 88%;
38      --ring: 45 90% 50%;
39
40      --radius: 0.75rem;
41    }
42
43    --gradient-primary: linear-gradient(135deg, □ hsl(45 90% 50%), □ hsl(45 95% 60%));
44    --gradient-card: linear-gradient(180deg, □ hsl(0 0% 15%), □ hsl(0 0% 17%));
45    --shadow-card: 0 2px 8px -2px □ hsl(45 98% 50% / 0.15);
46    --shadow-card-hover: 0 8px 24px -4px □ hsl(45 90% 50% / 0.25);
47    --transition-smooth: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
48
49    --sidebar-background: 0 0% 98%;
50
51    --sidebar-foreground: 240 5.3% 26.1%;
52
53    --sidebar-primary: 240 5.9% 10%;
54    --sidebar-primary-foreground: 0 0% 98%;
55
56    --sidebar-accent: 240 4.8% 95.9%;
57
58    --sidebar-accent-foreground: 240 5.9% 10%;
59
60    --sidebar-border: 220 13% 91%;
61
62    --sidebar-ring: 217.2 91.2% 59.8%;
63  }
64
65  .dark {
66    --background: 0 0% 8%;
67    --foreground: 45 95% 90%;
68
69    --card: 0 0% 12%;
70    --card-foreground: 45 95% 90%;
71
72    --popover: 0 0% 12%;
73    --popover-foreground: 45 95% 90%;
74
75    --primary: 45 95% 55%;
76    --primary-foreground: 0 0% 8%;
77    --primary-glow: 45 100% 65%;
78
79    --secondary: 45 25% 18%;
80    --secondary-foreground: 45 95% 75%;
81
82    --muted: 0 0% 18%;
83    --muted-foreground: 0 0% 60%;
84
85    --accent: 45 98% 50%;
86    --accent-foreground: 0 0% 8%;
87
88    --destructive: 0 63% 31%;
89    --destructive-foreground: 45 95% 90%;
90
91    --border: 0 0% 20%;
92    --input: 0 0% 20%;
93    --ring: 45 95% 55%;
94
95    --gradient-primary: linear-gradient(135deg, □ hsl(45 90% 50%), □ hsl(45 95% 60%));
96    --gradient-card: linear-gradient(180deg, □ hsl(0 0% 12%), □ hsl(0 0% 14%));
97    --shadow-card: 0 2px 8px -2px □ hsl(0 0% 0% / 0.5);
98    --shadow-card-hover: 0 8px 24px -4px □ hsl(45 95% 55% / 0.3);
99    --sidebar-background: 240 5.9% 10%;
100   --sidebar-foreground: 240 4.8% 95.9%;
101   --sidebar-primary: 224.3 76.3% 48%;
102   --sidebar-primary-foreground: 0 0% 100%;
103   --sidebar-accent: 240 3.7% 15.9%;
104   --sidebar-accent-foreground: 240 4.8% 95.9%;
105   --sidebar-border: 240 3.7% 15.9%;
106   --sidebar-ring: 217.2 91.2% 59.8%;
107  }
108
109  @layer base {
110    * {
111      @apply border-border;
112    }
113
114    body {
115      @apply bg-background text-foreground font-economica;
116    }
117  }
118

```

Upgrade Source III

```
1  <!doctype html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>설문조사 | 간편한 온라인 설문 플랫폼</title>
7      <meta name="description" content="쉽고 빠른 온라인 설문조사 플랫폼. 다양한 질문 유형으로 의견을 수집하고 분석하세요." />
8      <meta name="author" content="Survey Platform" />
9
10     <link rel="preconnect" href="https://fonts.googleapis.com">
11     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
12     <link href="https://fonts.googleapis.com/css2?family=Economica:wght@400;700&display=swap" rel="stylesheet">
13
14     <meta property="og:title" content="설문조사 | 간편한 온라인 설문 플랫폼" />
15     <meta property="og:description" content="쉽고 빠른 온라인 설문조사 플랫폼. 다양한 질문 유형으로 의견을 수집하고 분석하세요." />
16     <meta property="og:type" content="website" />
17     <meta property="og:image" content="https://lovable.dev/opengraph-image-p98pqg.png" />
18
19     <meta name="twitter:card" content="summary_large_image" />
20     <meta name="twitter:site" content="@lovable_dev" />
21     <meta name="twitter:image" content="https://lovable.dev/opengraph-image-p98pqg.png" />
22 </head>
23
24 <body>
25     <div id="root"></div>
26     <script type="module" src="/src/main.tsx"></script>
27 </body>
28 </html>
```

Upgrade Source III

```

1 import type { Config } from "tailwindcss";
2 |
3 export default {
4   darkMode: ["class"],
5   content: ["./pages/**/*.{ts,tsx}", "./components/**/*.{ts,tsx}", "./app/**/*.{ts,tsx}", "./src/**/*.{ts,tsx}"],
6   prefix: "",
7   theme: {
8     container: {
9       center: true,
10      padding: "2rem",
11      screens: {
12        "2xl": "1400px",
13      },
14    },
15   extend: {
16     colors: {
17       border: "hsl(var(--border))",
18       input: "hsl(var(--input))",
19       ring: "hsl(var(--ring))",
20       background: "hsl(var(--background))",
21       foreground: "hsl(var(--foreground))",
22     },
23     primary: {
24       DEFAULT: "hsl(var(--primary))",
25       foreground: "hsl(var(--primary-foreground))",
26     },
27     secondary: {
28       DEFAULT: "hsl(var(--secondary))",
29       foreground: "hsl(var(--secondary-foreground))",
30     },
31     destructive: {
32       DEFAULT: "hsl(var(--destructive))",
33       foreground: "hsl(var(--destructive-foreground))",
34     },
35     muted: {
36       DEFAULT: "hsl(var(--muted))",
37       foreground: "hsl(var(--muted-foreground))",
38     },
39     accent: {
40       DEFAULT: "hsl(var(--accent))",
41       foreground: "hsl(var(--accent-foreground))",
42     },
43   },
44   popover: {
45     DEFAULT: "hsl(var(--popover))",
46     foreground: "hsl(var(--popover-foreground))",
47   },
48   card: {
49     DEFAULT: "hsl(var(--card))",
50     foreground: "hsl(var(--card-foreground))",
51   },
52   sidebar: {
53     DEFAULT: "hsl(var(--sidebar-background))",
54     foreground: "hsl(var(--sidebar-foreground))",
55     primary: "hsl(var(--sidebar-primary))",
56     "primary-foreground": "hsl(var(--sidebar-primary-foreground))",
57     accent: "hsl(var(--sidebar-accent))",
58     "accent-foreground": "hsl(var(--sidebar-accent-foreground))",
59     border: "hsl(var(--sidebar-border))",
60     ring: "hsl(var(--sidebar-ring))",
61   },
62   fontFamily: {
63     economica: ['Economica', 'sans-serif'],
64   },
65   borderRadius: {
66     lg: "var(--radius)",
67     md: "calc(var(--radius) - 2px)",
68     sm: "calc(var(--radius) - 4px)",
69   },
70   keyframes: {
71     "accordion-down": {
72       from: {
73         height: "0",
74       },
75       to: {
76         height: "var(--radix-accordion-content-height)",
77       },
78     },
79     "accordion-up": {
80       from: {
81         height: "var(--radix-accordion-content-height)",
82       },
83       to: {
84         height: "0",
85       },
86     },
87   },
88   animation: {
89     "accordion-down": "accordion-down 0.2s ease-out",
90     "accordion-up": "accordion-up 0.2s ease-out",
91   },
92   plugins: [require("tailwindcss-animate")],
93 } satisfies Config;

```

Upgrade Source IV

Prompt: "고객 만족도 설문조사"를 그라데이션 하지말고
직관적인 디자인으로 변경해줘

고객 만족도 설문조사

여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며, 서비스 개선에 활용됩니다.

성함을 알려주세요 *

답변을 입력하세요

전반적인 만족도를 평가해주세요 *

- 매우 만족
- 만족
- 보통
- 불만족
- 매우 불만족

어떤 기능이 가장 유용했나요?(복수 선택 가능)

- 직관적인 UI/UX
- 빠른 로딩 속도
- 다양한 기능
- 모바일 지원
- 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

답변을 입력하세요

설문 제출하기

설문 작성 시간: 약 15분 소요

Upgrade Source IV

```

1 import { useState } from "react";
2 import { Button } from "@components/ui/button";
3 import { QuestionCard } from "./QuestionCard";
4 import { toast } from "@hooks/use-toast";
5 import { CheckCircle2 } from "lucide-react";
6 |
7 interface Question {
8   id: string;
9   question: string;
10  type: "text" | "textarea" | "radio" | "checkbox";
11  options?: { id: string; label: string }[];
12  required?: boolean;
13 }
14
15 const surveyQuestions: Question[] = [
16   {
17     id: "name",
18     question: "성함을 알려주세요",
19     type: "text",
20     required: true,
21   },
22   {
23     id: "satisfaction",
24     question: "전반적인 만족도를 평가해주세요",
25     type: "radio",
26     options: [
27       { id: "very-satisfied", label: "매우 만족" },
28       { id: "satisfied", label: "만족" },
29       { id: "neutral", label: "보통" },
30       { id: "dissatisfied", label: "불만족" },
31       { id: "very-dissatisfied", label: "매우 불만족" },
32     ],
33     required: true,
34   },
35   {
36     id: "features",
37     question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)",
38     type: "checkbox",
39     options: [
40       { id: "ui-design", label: "직관적인 UI/UX" },
41       { id: "speed", label: "빠른 로딩 속도" },
42     ],
43     required: false,
44   },
45   {
46     id: "feedback",
47     question: "개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요",
48     type: "textarea",
49     required: false,
50   },
51 ];
52
53
54
55
56 export const SurveyForm = () => {
57   const [answers, setAnswers] = useState<Record<string, string | string[]>({}); // 초기화된 응답 객체
58   const [errors, setErrors] = useState<Record<string, string>({}); // 초기화된 오류 객체
59   const [isSubmitted, setIsSubmitted] = useState(false); // 폼이 제출되었는지 여부
60
61   const handleAnswerChange = (questionId: string, value: string | string[]) => {
62     setAnswers((prev) => ({ ...prev, [questionId]: value }));
63     // Clear error when user starts typing
64     if (!errors[questionId]) {
65       setErrors((prev) => {
66         const newErrors = { ...prev };
67         delete newErrors[questionId];
68         return newErrors;
69       });
70     }
71   }
72
73   const validateForm = (): boolean => {
74     const newErrors: Record<string, string> = {};
75
76     surveyQuestions.forEach((question) => {
77       if (question.required) {
78         const answer = answers[question.id];
79         if (!answer || !Array.isArray(answer) && answer.length === 0) {
80           newErrors[question.id] = "필수 항목입니다";
81         }
82       }
83     })
84
85     if (typeof answer === "string" && answer.trim() === "") {
86       newErrors[question.id] = "답변을 입력해주세요";
87     }
88   }
89
90   setErrors(newErrors);
91   return Object.keys(newErrors).length === 0;
92 }
93
94 const handleSubmit = (e: React.FormEvent) => {
95   e.preventDefault();
96
97   if (!validateForm()) {
98     toast({
99       title: "입력 확인",
100      description: "필수 항목을 모두 입력해주세요.",
101      variant: "destructive",
102    });
103    return;
104  }
105
106  // Here you would typically send the data to a backend
107  console.log("Survey answers:", answers);
108
109  setIsSubmitted(true);
110  toast({
111    title: "제출 완료",
112    description: "소중한 의견 감사합니다!",
113  });
114
115  if (isSubmitted) {
116    return (
117      <div className="min-h-screen flex items-center justify-center p-4 bg-background">
118        <div className="max-w-2xl w-full text-center space-y-6 animate-in fade-in duration-see...
119        <div className="flex justify-center">
120          <div className="relative">
121            <div className="absolute inset-0 bg-primary/20 blur-xl rounded-full animate-pulse" />

```

Upgrade Source IV

```

120   <CheckCircle2 className="w-24 h-24 text-primary relative animate-in zoom-in duration-300" />
121 </div>
122 </div>
123 <div className="space-y-2">
124   <h2 className="text-3xl font-bold text-foreground">
125     감사합니다!
126   </h2>
127   <p className="text-lg text-muted-foreground">
128     설문에 참여해 주셔서 감사합니다.
129     <br />
130   소중한 의견은 더 나은 서비스를 만드는 데 활용하겠습니다.
131   </p>
132 </div>
133 <Button
134   onClick={() => {
135     setIsSubmitted(false);
136     setAnswers({});
137     setErrors({});
138   }}
139   variant="outline"
140   size="lg"
141   className="transition-all duration-200"
142 >
143   새 설문 작성하기
144 </Button>
145 </div>
146 </div>
147 );
148 }

return (
149 <div className="min-h-screen py-12 px-4 bg-background">
150   <div className="max-w-3xl mx-auto space-y-8 animate-in fade-in duration-500">
151     <header className="text-center space-y-4">
152       <h1 className="text-5xl font-bold text-primary">
153         고객 만족도 설문조사
154       </h1>
155       <p className="text-lg text-muted-foreground max-w-2xl mx-auto">
156         여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며,
157         서비스 개선에 활용됩니다.
158       </p>
159     </header>
160     <form onSubmit={handleSubmit} className="space-y-6">
161       {surveyQuestions.map((question, index) => (
162         <div key={question.id} className="animate-in slide-in-from-bottom duration-500" style={{ animationDelay: `${(index * 100) + 50}ms` }}>
163           <QuestionCard
164             question={question}
165             type={question.type}
166             options={question.options}
167             required={question.required}
168             value={answers[question.id]}
169             onChange={value => handleAnswerChange(question.id, value)}
170             error={errors[question.id]}
171           />
172         </div>
173       ))
174     </form>
175     <div className="pt-4">
176       <button type="submit" size="lg" className="w-full bg-gradient-to-r from-primary to-primary-glow hover:opacity-90 transition-all duration-300 text-primary-foreground font-semibold shadow-lg hover:shadow-xl">
177         설문 제출하기
178       </button>
179     </div>
180   </div>
181 </div>
182 <Footer className="text-center text-sm text-muted-foreground pt-8 border-t">
183   <p>설문 작성 시간: 약 3-5분 소요</p>
184 </Footer>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>
190 </div>
191 </div>
192 </div>
193 </div>
194 </div>
195 </div>
196 </div>
197 </div>
198 </div>
199 );

```

Upgrade Source V

Prompt: "여러분의 소중한 의견을 들려주세요. 모든 답변은 익명으로 처리되며, 서비스 개선에 활용됩니다."를 화면상 1줄로 출력되게 변경해줘

고객 만족도 설문조사

소중한 의견 감사합니다. 익명 처리되며 서비스 개선에 활용됩니다.

성함을 알려주세요 *

답변을 입력하세요

전반적인 만족도를 평가해주세요 *

- 매우 만족
- 만족
- 보통
- 불만족
- 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)

- 직관적인 배포
- 빠른 로딩 속도
- 다양한 기능
- 모바일 지원
- 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

답변을 입력하세요

설문 제출하기

설문 작성 시간: 약 3-5분 소요

Upgrade Source V

```

1 import { useState } from "react"; 38
2 import { Button } from "@components/ui/button"; 39
3 import { QuestionCard } from "./QuestionCard"; 40
4 import { toast } from "@hooks/use-toast"; 41
5 import { CheckCircle2 } from "lucide-react"; 42
6
7 interface Question { 43
8   id: string; 44
9   question: string; 45
10  type: "text" | "textarea" | "radio" | "checkbox"; 46
11  options?: { id: string; label: string }[]; 47
12  required?: boolean; 48
13 } 49
14
15 const surveyQuestions: Question[] = [ 50
16 {
17   id: "name", 51
18   question: "성함을 알려주세요", 52
19   type: "text", 53
20   required: true, 54
21 }, 55
22 {
23   id: "satisfaction", 56
24   question: "전반적인 만족도를 평가해주세요", 57
25   type: "radio", 58
26   options: [ 59
27     { id: "very-satisfied", label: "매우 만족" }, 60
28     { id: "satisfied", label: "만족" }, 61
29     { id: "neutral", label: "보통" }, 62
30     { id: "dissatisfied", label: "불만족" }, 63
31     { id: "very-dissatisfied", label: "매우 불만족" }, 64
32   ], 65
33   required: true, 66
34 }, 67
35 {
36   id: "features", 68
37   question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)", 69
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
625
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
1634
1635
1635
1636
1636
1637
1637
1638
1638
1639
1639
1640
1640
1641
1641
1642
1642
1643
1643
1644
1644
1645
1645
1646
1646
1647
1647
```

Upgrade Source V

SurveyForm.tsx

```
199     );
198   
```

```
199     <div>
199       <div>
199         <div>
199           <div>
199             <div>
199               <div>
199                 <div>
199                   <div>
199                     <div>
199                       <div>
199                         <div>
199                           <div>
199                             <div>
199                               <div>
199                                 <div>
199                                   <div>
199                                     <div>
199                                       <div>
199                                         <div>
199                                           <div>
199                                             <div>
199                                               <div>
199                                                 <div>
199                                                   <div>
199                                                     <div>
199                                                       <div>
199                                                         <div>
199                                                           <div>
199                                                             <div>
199                                                               <div>
199                                                                 <div>
199                                                                   <div>
199                                                                     <div>
199                                                                       <div>
199                                                                         <div>
199                                                                           <div>
199                                                                             <div>
199                                                                               <div>
199                                                                                 <div>
199                                                                 <div>
199                                                                 <div>
199         </div>
199       </div>
199     </div>
199   </div>
199 
```

Upgrade Source VI

Prompt: 객관식으로 성별과 나이대를 입력할 수 있게 변경해줘

고객 만족도 설문조사

소중한 의견 감사합니다. 익명 처리되어 서비스 개선에 활용됩니다.

성별을 알려주세요 *

답변을 입력하세요

성별을 선택해주세요 *

남성
 여성
 기타
 등당 안함

나이대를 선택해주세요 *

10대
 20대
 30대
 40대
 50대
 60대 이상

전반적인 만족도를 평가해주세요 *

매우 만족
 만족
 보통
 불만족
 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)

직관적인 인터페이스
 빠른 프로세스 속도
 다양한 기능
 모바일 지원
 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

답변을 입력하세요

질문 제출하기

설문 작성 시간(분) : 분초

Upgrade Source VI

```

1  import { useState } from "react";
2  import { Button } from "@/components/ui/button";
3  import { QuestionCard } from "./QuestionCard";
4  import { toast } from "@/hooks/use-toast";
5  import { CheckCircle2 } from "lucide-react";
6
7  interface Question {
8    id: string;
9    question: string;
10   type: "text" | "textarea" | "radio" | "checkbox";
11   options?: { id: string; label: string }[];
12   required?: boolean;
13 }
14
15 const surveyQuestions: Question[] = [
16   {
17     id: "name",
18     question: "성함을 알려주세요",
19     type: "text",
20     required: true,
21   },
22   {
23     id: "gender",
24     question: "성별을 선택해주세요",
25     type: "radio",
26     options: [
27       { id: "male", label: "남성" },
28       { id: "female", label: "여성" },
29       { id: "other", label: "기타" },
30       { id: "prefer-not-to-say", label: "응답 안함" },
31     ],
32     required: true,
33   },
34   {
35     id: "age",
36     question: "나이대를 선택해주세요",
37     type: "radio",
38   },
39   {
40     id: "10s", label: "10내",
41     id: "20s", label: "20대",
42     id: "30s", label: "30대",
43     id: "40s", label: "40대",
44     id: "50s", label: "50대",
45     id: "60s", label: "60대 이상",
46   ],
47   required: true,
48 },
49   {
50     id: "satisfaction",
51     question: "전반적인 만족도를 평가해주세요",
52     type: "radio",
53     options: [
54       { id: "very-satisfied", label: "매우 만족" },
55       { id: "satisfied", label: "만족" },
56       { id: "neutral", label: "보통" },
57       { id: "dissatisfied", label: "불만족" },
58       { id: "very-dissatisfied", label: "매우 불만족" },
59     ],
60   },
61   {
62     id: "features",
63     question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)",
64     type: "checkbox",
65     options: [
66       { id: "ui-design", label: "직관적인 UI/UX" },
67       { id: "speed", label: "빠른 로딩 속도" },
68       { id: "features", label: "다양한 기능" },
69       { id: "mobile", label: "모바일 지원" },
70       { id: "customer-service", label: "고객 지원" },
71     ],
72   },
73   {
74     id: "feedback",
75     question: "개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요",
76     type: "textarea",
77     required: false,
78   },
79 ],
80 ];
81
82 export const SurveyForm = () => {
83   const [answers, setAnswers] = useState<Record<string, string | string[]>>({});
84   const [errors, setErrors] = useState<Record<string, string>>({});
85   const [isSubmitted, setIsSubmitted] = useState(false);
86
87   const handleAnswerChange = (questionId: string, value: string | string[]) => {
88     setAnswers((prev) => ({ ...prev, [questionId]: value }));
89     // Clear error when user starts typing
90     if (errors[questionId]) {
91       setErrors((prev) => {
92         const newErrors = { ...prev };
93         delete newErrors[questionId];
94         return newErrors;
95       });
96     }
97   };
98
99   const validateForm = (): boolean => {
100     const newErrors: Record<string, string> = {};
101
102     surveyQuestions.forEach((question) => {
103       if (question.required) {
104         const answer = answers[question.id];
105         if (!answer || (!Array.isArray(answer) && answer.length === 0)) {
106           newErrors[question.id] = "필수 항목입니다";
107         } else if (typeof answer === "string" && answer.trim() === "") {
108           newErrors[question.id] = "답변을 입력해주세요";
109         }
110     });
111   };
112
113   return (
114     <div>
115       <h1>Survey Form</h1>
116       <Form>
117         <FormSection>
118           <FormSectionTitle>Personal Information</FormSectionTitle>
119           <FormSectionContent>
120             <FormInput>
121               <FormLabel>Name</FormLabel>
122               <FormControl>{answers.name}</FormControl>
123             </FormInput>
124             <FormInput>
125               <FormLabel>Gender</FormLabel>
126               <FormControl>
127                 {surveyQuestions.map((q) => q.type === "radio" ? (
128                   <FormRadioGroup>
129                     {q.options.map((o) => (
130                       <FormRadio value={o.id}>{o.label}</FormRadio>
131                     ))}
132                   </FormRadioGroup>
133                 ) : null)}
134               </FormControl>
135             </FormInput>
136             <FormInput>
137               <FormLabel>Age</FormLabel>
138               <FormControl>
139                 {surveyQuestions.map((q) => q.type === "radio" ? (
140                   <FormRadioGroup>
141                     {q.options.map((o) => (
142                       <FormRadio value={o.id}>{o.label}</FormRadio>
143                     ))}
144                   </FormRadioGroup>
145                 ) : null)}
146               </FormControl>
147             </FormInput>
148           </FormSectionContent>
149           <FormSectionTitle>Satisfaction</FormSectionTitle>
150           <FormSectionContent>
151             <FormInput>
152               <FormLabel>Overall Satisfaction</FormLabel>
153               <FormControl>
154                 {surveyQuestions.map((q) => q.type === "radio" ? (
155                   <FormRadioGroup>
156                     {q.options.map((o) => (
157                       <FormRadio value={o.id}>{o.label}</FormRadio>
158                     ))}
159                   </FormRadioGroup>
160                 ) : null)}
161               </FormControl>
162             </FormInput>
163           </FormSectionContent>
164           <FormSectionTitle>Feedback</FormSectionTitle>
165           <FormSectionContent>
166             <FormTextarea>
167               <FormLabel>Leave feedback</FormLabel>
168               <FormControl>{answers.feedback}</FormControl>
169             </FormTextarea>
170           </FormSectionContent>
171         </FormSection>
172         <FormFooter>
173           <FormSubmit>Submit</FormSubmit>
174         </FormFooter>
175       </Form>
176     </div>
177   );
178 }

```

Upgrade Source VI

```

110    }
111  });
112
113  setErrors(newErrors);
114  return Object.keys(newErrors).length === 0;
115 };
116
117 const handleSubmit = (e: React.FormEvent) => {
118  e.preventDefault();
119
120  if (!validateForm()) {
121    toast({
122      title: "입력 확인",
123      description: "필수 항목을 모두 입력해주세요.",
124      variant: "destructive",
125    });
126    return;
127  }
128
129 // Here you would typically send the data to a backend
130 console.log("Survey answers:", answers);
131
132 setIsSubmitted(true);
133 toast({
134  title: "제출 완료",
135  description: "소중한 의견 감사합니다!",
136 });
137 };
138
139 if (isSubmitted) {
140  return (
141    <div className="min-h-screen flex items-center justify-center p-4 bg-background">
142      <div className="max-w-2xl w-full text-center space-y-6 animate-in fade-in duration-500">
143        <div className="flex justify-center">
144          <div className="relative">
145            <div className="absolute inset-0 bg-primary/20 blur-xl rounded-full animate-pulse" />
146            <CheckCircle2 className="w-24 h-24 text-primary relative animate-in zoom-in duration-300" />
147          </div>
148        </div>
149        <div className="space-y-3">
150          <h2 className="text-3xl font-bold text-foreground">
151            감사합니다!
152          </h2>
153          <p className="text-lg text-muted-foreground">
154            설문에 참여해 주셔서 감사합니다.
155          <br />
156          소중한 의견은 더 나은 서비스를 만드는데 활용하겠습니다.
157        </p>
158        <div>
159          <button
160            onClick={() => {
161              setIsSubmitted(false);
162              setAnswers({});
163              setErrors({});
164            }}
165            variant="outline"
166            size="lg"
167            className="transition-all duration-200"
168            >
169              새 질문 작성하기
170            </button>
171          </div>
172        </div>
173      </div>
174    </div>
175  );
176  return [
177    <div className="min-h-screen py-12 px-4 bg-background">
178      <div className="max-w-3xl mx-auto space-y-8 animate-in fade-in duration-500" style={{ animationDelay: `${(index * 100) + 100}ms` }}>
179        <header className="text-center space-y-4">
180          <h1 className="text-5xl font-bold text-primary">
181            고객 만족도 설문조사
182          </h1>
183          <p className="text-lg text-muted-foreground">
184            소중한 의견 감사합니다. 익명 처리되어 서비스 개선에 활용됩니다.
185          </p>
186        </header>
187        <form onSubmit={handleSubmit} className="space-y-6">
188          {surveyQuestions.map((question, index) => (
189            <div key={question.id} className="animate-in slide-in-from-bottom duration-500" style={{ animationDelay: `${(index * 100) + 100}ms` }}>
190              <QuestionCard
191                question={question.question}
192                type={question.type}
193                options={question.options}
194                required={question.required}
195                value={answers[question.id]}
196                onChange={(value) => handleAnswerChange(question.id, value)}
197                error={errors[question.id]}
198              />
199            </div>
200          ))
201        </form>
202        <div className="pt-4">
203          <button
204            type="submit"
205            size="lg"
206            className="w-full bg-gradient-to-r from-primary to-primary-glow hover:opacity-90 transition-all duration-300 text-primary-foreground font-semibold shadow-lg hover:shadow-xl" style={{ animationDelay: `${(index * 100) + 100}ms` }}>
207            설문 제출하기
208          </button>
209        </div>
210      </div>
211    </div>
212  </div>
213  </div>
214  </div>
215  </div>
216  </div>
217  </div>
218  </div>
219  <footer className="text-center text-sm text-muted-foreground pt-8 border-t">
220    <p>설문 작성 시간: 약 3-5분 소요</p>
221  </footer>
222  </div>
223  </div>
224  </div>
225  </div>

```

Upgrade Source VII

Prompt: 성별 체크항목에서 기타, 응답 안함 항목을 제거해줘

고객 만족도 설문조사
소중한 의견 감사합니다. 익명 처리되어 서비스 개선에 활용됩니다.

성별을 알려주세요 *

성별을 선택해주세요 *
 남성
 여성

나이대를 선택해주세요 *
 10대
 20대
 30대
 40대
 50대
 60대 이상

전반적인 만족도를 평가해주세요 *
 매우 만족
 만족
 보통
 불만족
 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)
 직관적인 UI/UX
 빠른 로딩 속도
 다양한 기능
 모바일 지원
 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

설문 제출하기

설문 작성 시간 약 1~2분 소요

Upgrade Source VII

```

1 import { useState } from "react";
2 import { Button } from "@/components/ui/button";
3 import { QuestionCard } from "./QuestionCard";
4 import { toast } from "@/hooks/use-toast";
5 import { Checkcircle2 } from "lucide-react";
6
7 interface Question {
8   id: string;
9   question: string;
10  type: "text" | "textarea" | "radio" | "checkbox";
11  options?: { id: string; label: string }[];
12  required?: boolean;
13 }
14
15 const surveyQuestions: Question[] = [
16 {
17   id: "name",
18   question: "성함을 알려주세요",
19   type: "text",
20   required: true,
21 },
22 {
23   id: "gender",
24   question: "성별을 선택해주세요",
25   type: "radio",
26   options: [
27     { id: "male", label: "남성" },
28     { id: "female", label: "여성" },
29   ],
30   required: true,
31 },
32 {
33   id: "age",
34   question: "나이대를 선택해주세요",
35   type: "radio",
36   options: [
37     { id: "10s", label: "10대" },
38     { id: "20s", label: "20대" },
39     { id: "30s", label: "30대" },
40     { id: "40s", label: "40대" },
41     { id: "50s", label: "50대" },
42     { id: "60s", label: "60대 이상" },
43   ],
44   required: true,
45 },
46 {
47   id: "satisfaction",
48   question: "전반적인 만족도를 평가해주세요",
49   type: "radio",
50   options: [
51     { id: "very-satisfied", label: "매우 만족" },
52     { id: "satisfied", label: "만족" },
53     { id: "neutral", label: "보통" },
54     { id: "dissatisfied", label: "불만족" },
55     { id: "very-dissatisfied", label: "매우 불만족" },
56   ],
57   required: true,
58 },
59 {
60   id: "features",
61   question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)",
62   type: "checkbox",
63   options: [
64     { id: "ui-design", label: "직관적인 UI/UX" },
65     { id: "speed", label: "빠른 로딩 속도" },
66     { id: "features", label: "다양한 기능" },
67     { id: "mobile", label: "모바일 지원" },
68     { id: "customer-service", label: "고객 지원" },
69   ],
70   required: false,
71 },
72 ]
73
74
75
76
77
78
79
80 export const SurveyForm = () => {
81   const [answers, setAnswers] = useState<Record<string, string | string[]>>({});
82   const [errors, setErrors] = useState<Record<string, string>>({});
83   const [isSubmitted, setIsSubmitted] = useState(false);
84
85   const handleAnswerChange = (questionId: string, value: string | string[]) => {
86     setAnswers((prev) => ({ ...prev, [questionId]: value }));
87     // Clear error when user starts typing
88     if (!errors[questionId]) {
89       setErrors((prev) => {
90         const newErrors = { ...prev };
91         delete newErrors[questionId];
92         return newErrors;
93       });
94     }
95   };
96
97
98
99
100 const validateForm = (): boolean => {
101   const newErrors: Record<string, string> = {};
102
103   surveyQuestions.forEach((question) => {
104     if (question.required) {
105       const answer = answers[question.id];
106       if (!answer || (Array.isArray(answer) && answer.length === 0)) {
107         newErrors[question.id] = "필수 항목입니다";
108       } else if (typeof answer === "string" && answer.trim() === "") {
109         newErrors[question.id] = "답변을 입력해주세요";
110       }
111     }
112   });
113
114   return Object.keys(newErrors).length === 0;
115 }

```

Upgrade Source VII

SurveyForm.tsx

```
109 });
110
111 setErrors(newErrors);
112 return object.keys(newErrors).length === 0;
113 };
114
115 const handleSubmit = (e: React.FormEvent) => {
116   e.preventDefault();
117
118   if (!validateForm()) {
119     toast({
120       title: "입력 확인",
121       description: "필수 항목을 모두 입력해주세요.",
122       variant: "destructive",
123     });
124     return;
125   }
126
127 // Here you would typically send the data to a backend
128 console.log("Survey answers:", answers);
129
130 setIsSubmitted(true);
131 toast({
132   title: "제출 완료",
133   description: "소중한 의견 감사합니다!",
134 });
135 }
136
137 if (isSubmitted) {
138   return (
139     <div className="min-h-screen flex items-center justify-center p-4 bg-background">
140       <div className="max-w-2xl w-full text-center space-y-6 animate-in fade-in duration-500">
141         <div className="flex justify-center">
142           <div className="relative">
```

```
143           <div className="absolute inset-0 bg-primary/20 blur-xl rounded-full animate-pulse" />
144           <CheckCircle2 className="w-24 h-24 text-primary relative animate-in zoom-in duration-300" />
145         </div>
146       </div>
147       <div className="space-y-2">
148         <h2 className="text-3xl font-bold text-primary">
149           감사합니다!
150         </h2>
151         <p className="text-lg text-muted-foreground">
152           질문에 참여해 주셔서 감사합니다.
153         <br />
154         소중한 의견은 더 나은 서비스를 만드는 데 활용하겠습니다.
155       </p>
156       <button
157         onClick={() => {
158           setIsSubmitted(false);
159           setAnswers({});
160           setErrors({});
161         }}
162         variant="outline"
163         size="lg"
164         className="transition-all duration-200"
165       >
166         새 설문 작성하기
167       </button>
168     </div>
169   </div>
170 </div>
171 );
172 }
173
174 return (
175   <div className="min-h-screen py-12 px-4 bg-background">
176     <div className="max-w-3xl mx-auto space-y-8 animate-in fade-in duration-500">
177       <header className="text-center space-y-4">
178         <h1 className="text-5xl font-bold text-primary">
```

```
179         고객 만족도 설문조사
180       </h1>
181       <p className="text-lg text-muted-foreground">
182         소중한 의견 감사합니다. 이영 처리되어 서비스 개선에 활용됩니다.
183       </p>
184     </header>
185
186     <form onSubmit={handleSubmit} className="space-y-6">
187       <{surveyQuestions.map((question, index)} => (
188         <div
189           key={question.id}
190           className="animate-in slide-in-from-bottom duration-500"
191           style={{ animationDelay: `${index * 100}ms` }}>
192           <QuestionCard
193             question={question.question}
194             type={question.type}
195             options={question.options}
196             required={question.required}
197             value={answers[question.id]}
198             onChange={(value) => handleAnswerChange(question.id, value)}
199             error={errors[question.id]}
200           />
201         </div>
202       ))}>
203     </form>
204   <div className="pt-4">
205     <Button
206       type="submit"
207       size="lg"
208       className="w-full bg-gradient-to-r from-primary to-primary-glow hover:opacity-90
209         transition-all duration-300 text-primary-foreground font-semibold shadow-lg shadow-xl">
210       >
211         설문 제출하기
212       </Button>
213     </div>
214   </div>
215
216 </div>
217
218 <footer className="text-center text-sm text-muted-foreground pt-8 border-t">
219   <p>설문 작성 시간: 약 3-5분 소요</p>
220 </footer>
221 </div>
222 </div>
223 );
```

Upgrade Source VIII

Prompt: "익명 처리되며 서비스 개선에 활용됩니다." 문구만 제거해줘

고객 만족도 설문조사

소중한 의견 감사합니다.

성별을 알려주세요 *

남성
 여성

성별을 선택해주세요 *

남성
 여성

나이대를 선택해주세요 *

10대
 20대
 30대
 40대
 50대
 60대 이상

진법적인 만족도를 정기해주세요 *

매우 만족
 만족
 보통
 불만족
 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)

직관적인 인터페이스
 빠른 로딩 속도
 다양한 기능
 모바일 지원
 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

답변을 양방향하세요.

입력 제출하기

설문작성 시간 약 1분 소요

Upgrade Source VIII

SurveyForm.tsx

```
1 import { useState } from "react";
2 import { Button } from "@/components/ui/button";
3 import { QuestionCard } from "@/components/ui/question-card";
4 import { toast } from "@/hooks/use-toast";
5 import { CheckCircle2 } from "lucide-react";
6
7 interface Question {
8   id: string;
9   question: string;
10  type: "text" | "textarea" | "radio" | "checkbox";
11  options?: { id: string; label: string }[];
12  required?: boolean;
13 }
14
15 const surveyQuestions: Question[] = [
16 {
17   id: "name",
18   question: "성함을 알려주세요",
19   type: "text",
20   required: true,
21 },
22 {
23   id: "gender",
24   question: "성별을 선택해주세요",
25   type: "radio",
26   options: [
27     { id: "male", label: "남성" },
28     { id: "female", label: "여성" },
29   ],
30   required: true,
31 },
32 {
33   id: "age",
34   question: "나이대를 선택해주세요",
35   type: "radio",
36   options: [
37     { id: "10s", label: "10대" },
38     {
39       id: "20s", label: "20대" },
40       id: "30s", label: "30대" },
41       id: "40s", label: "40대" },
42       id: "50s", label: "50대" },
43       id: "60s", label: "60대 이상" },
44     ],
45     required: true,
46   },
47   {
48     id: "satisfaction",
49     question: "전반적인 만족도를 평가해주세요",
50     type: "radio",
51     options: [
52       { id: "very-satisfied", label: "매우 만족" },
53       { id: "satisfied", label: "만족" },
54       { id: "neutral", label: "보통" },
55       { id: "dissatisfied", label: "불만족" },
56       { id: "very-dissatisfied", label: "매우 불만족" },
57     ],
58     required: true,
59   },
60   {
61     id: "features",
62     question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)",
63     type: "checkbox",
64     options: [
65       { id: "ui-design", label: "직관적인 UI/UX" },
66       { id: "speed", label: "빠른 로딩 속도" },
67       { id: "features", label: "다양한 기능" },
68       { id: "mobile", label: "모바일 지원" },
69       { id: "customer-service", label: "고객 지원" },
70     ],
71     required: false,
72   },
73   {
74     id: "feedback",
75     question: "개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요",
76     type: "textarea",
77     required: false,
78   },
79 ]
80
81 export const SurveyForm = () => {
82   const [answers, setAnswers] = useState<Record<string, string | string[]>>({}); // 초기화된 응답 객체
83   const [errors, setErrors] = useState<Record<string, string>>({}); // 초기화된 오류 객체
84   const [isSubmitted, setIsSubmitted] = useState(false); // 폼이 제출되었는지 여부
85
86   const handleAnswerChange = (questionId: string, value: string | string[]) => {
87     setAnswers((prev) => ({ ...prev, [questionId]: value }));
88   }
89
90   // Clear error when user starts typing
91   if (errors[questionId]) {
92     setErrors((prev) => {
93       const newErrors = { ...prev };
94       delete newErrors[questionId];
95       return newErrors;
96     });
97   }
98
99   const validateForm = (): boolean => {
100   const newErrors: Record<string, string> = {};
101
102   surveyQuestions.forEach((question) => {
103     if (question.required) {
104       const answer = answers[question.id];
105       if (!answer || (!Array.isArray(answer) && answer.length === 0)) {
106         newErrors[question.id] = "필수 항목입니다";
107       } else if (typeof answer === "string" && answer.trim() === "") {
108         newErrors[question.id] = "답변을 입력해주세요";
109       }
110     }
111   })
112
113   return Object.keys(newErrors).length === 0;
114 }
115
116 // 테스트 코드
117 // console.log(validateForm());
```

Upgrade Source VIII

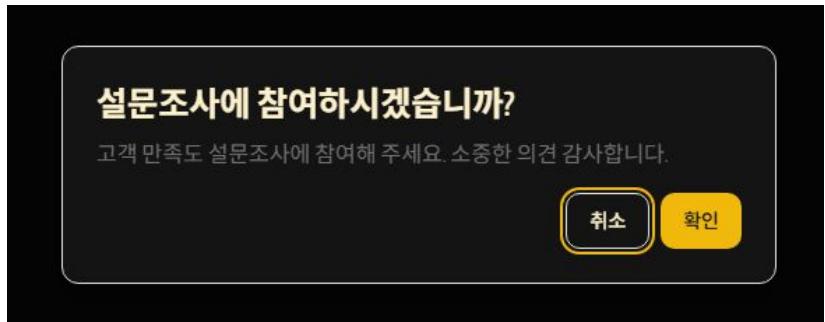
```

109 });
110
111 setErrors(newErrors);
112 return object.keys(newErrors).length === 0;
113 };
114
115 const handleSubmit = (e: React.FormEvent) => {
116   e.preventDefault();
117
118   if (!validateForm()) {
119     toast({
120       title: "입력 확인",
121       description: "필수 항목을 모두 입력해주세요.",
122       variant: "destructive",
123     });
124   }
125
126 // Here you would typically send the data to a backend
127 console.log("Survey answers:", answers);
128
129 setIsSubmitted(true);
130
131 toast({
132   title: "제출 완료",
133   description: "소중한 의견 감사합니다!",
134 });
135
136 if (isSubmitted) {
137   return (
138     <div className="min-h-screen flex items-center justify-center p-4 bg-background">
139       <div className="max-w-2xl w-full text-center space-y-6 animate-in fade-in duration-500">
140         <div className="flex justify-center">
141           <div className="relative">
142             <div className="absolute inset-0 bg-primary/20 blur-xl rounded-full animate-pulse" />
143             <CheckCircle2 className="w-24 h-24 text-primary relative animate-in zoom-in duration-300" />
144           </div>
145         </div>
146
147         <div className="space-y-2">
148           <div className="text-3xl font-bold text-foreground">
149             감사합니다!
150           </div>
151           <p className="text-lg text-muted-foreground">
152             설문에 참여해 주셔서 감사합니다.
153           <br />
154           소중한 의견은 더 나은 서비스를 만드는 데 활용하겠습니다.
155           </p>
156         </div>
157         <button
158           onClick={() => {
159             setIsSubmitted(false);
160             setAnswers({});
161             setErrors({});
162           }}
163           variant="outline"
164           size="lg"
165           className="transition-all duration-200"
166           style={{ border: "1px solid #000", color: "#000", padding: "10px 20px" }}>
167           새 설문 작성하기
168         </button>
169       </div>
170     </div>
171   );
172 }
173
174 if (isSubmitted) {
175   return (
176     <div className="min-h-screen py-12 px-4 bg-background">
177       <div className="max-w-3xl mx-auto space-y-8 animate-in fade-in duration-500">
178         <header className="text-center space-y-4">
179           <h1 className="text-5xl font-bold text-primary">
180             </h1>
181             <p className="text-lg text-muted-foreground">
182               소중한 의견 감사합니다.
183             </p>
184           </header>
185
186           <form onSubmit={handleSubmit} className="space-y-6">
187             {surveyQuestions.map((question, index) => (
188               <div
189                 key={question.id}
190                 className="animate-in slide-in-from-bottom duration-500"
191                 style={{ animationDelay: `${index * 100}ms` }}>
192                 <QuestionCard
193                   question={question.question}
194                   type={question.type}
195                   options={question.options}
196                   required={question.required}
197                   value={answers[question.id]}
198                   onChange={(value) => handleAnswerChange(question.id, value)}
199                   error={errors[question.id]}
200                 />
201               </div>
202             )));
203
204             <div className="pt-4">
205               <button
206                 type="submit"
207                 size="lg"
208                 className="w-full bg-gradient-to-r from-primary to-primary-glow hover:opacity-90
209                 transition-all duration-300 text-primary-foreground font-semibold shadow-lg hover:shadow-xl">
210                 설문 제출하기
211               </button>
212             </div>
213           </form>
214
215         <footer className="text-center text-sm text-muted-foreground pt-8 border-t">
216           <p>설문 작성 시간: 약 3-5분 소요</p>
217         </footer>
218       </div>
219     </div>
220   );
221 }
222
223 );

```

Upgrade Source IX

Prompt: 설문조사 페이지 진입 전, "설문조사에 참여하시겠습니까?" 라고 물어보고 "확인/취소" 버튼을 클릭해서 진입 할 수 있게 변경해줘.



Upgrade Source IX

```

1 ✓ import { useState } from "react";                                38
2 import { Button } from "@/components/ui/button";                  39
3 import { QuestionCard } from "@/QuestionCard";                 40
4 import { toast } from "@/hooks/use-toast";                      41
5 import { CheckCircle2 } from "lucide-react";                   42
6 ✓ import {
7   AlertDialog,
8   AlertDialogAction,
9   AlertDialogCancel,
10  AlertDialogContent,
11  AlertDialogDescription,
12  AlertDialogFooter,
13  AlertDialogHeader,
14  AlertDialogTitle,
15 } from "@/components/ui/alert-dialog";                         51
16
17 ✓ interface Question {                                         53
18   id: string;                                                 54
19   question: string;                                           55
20   type: "text" | "textarea" | "radio" | "checkbox";          56
21   options?: { id: string; label: string }[];                  57
22   required?: boolean;                                         58
23 }
24
25 ✓ const surveyQuestions: Question[] = [                     61
26   {
27     id: "name",
28     question: "성함을 알려주세요",
29     type: "text",
30     required: true,
31   },
32   {
33     id: "gender",
34     question: "성별을 선택해주세요",
35     type: "radio",
36     options: [                                              68
37       { id: "male", label: "남성" },                          69
38       { id: "female", label: "여성" },                        70
39       required: true,                                         71
40     ],
41     required: true,                                         72
42   },
43   {
44     id: "age",
45     question: "나이대를 선택해주세요",
46     type: "radio",
47     options: [                                              73
48       { id: "10s", label: "10대" },
49       { id: "20s", label: "20대" },
50       { id: "30s", label: "30대" },
51       { id: "40s", label: "40대" },
52       { id: "50s", label: "50대" },
53       { id: "60s", label: "60대 이상" },
54     ],
55     required: true,
56   },
57   {
58     id: "satisfaction",
59     question: "전반적인 만족도를 평가해주세요",
60     type: "radio",
61     options: [                                              74
62       { id: "very-satisfied", label: "매우 만족" },
63       { id: "satisfied", label: "만족" },
64       { id: "neutral", label: "보통" },
65       { id: "dissatisfied", label: "불만족" },
66       { id: "very-dissatisfied", label: "매우 불만족" },
67     ],
68     required: true,
69   },
70   {
71     id: "features",
72     question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)", 73
73     type: "checkbox",
74   },
75   {
76     id: "ui-design", label: "직관적인 UI/UX" },
77     label: "빠른 로딩 속도" },
78     label: "다양한 기능" },
79     label: "모바일 지원" },
80     label: "고객 지원" },
81   ],
82   required: false,
83 },
84   {
85     id: "feedback",
86     question: "개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요",
87     type: "textarea",
88     required: false,
89   },
90
91 export const SurveyForm = () => {
92   const [answers, setAnswers] = useState<Record<string, string | string[]>>({});
93   const [errors, setErrors] = useState<Record<string, string>>({});
94   const [isSubmitted, setIsSubmitted] = useState(false);
95   const [hasAgreed, setHasAgreed] = useState(false);
96
97   const handleAnswerChange = (questionId: string, value: string | string[]) => {
98     setAnswers((prev) => ({ ...prev, [questionId]: value }));
99     // Clear error when user starts typing
100    if (errors[questionId]) {
101      setErrors((prev) => {
102        const newErrors = { ...prev };
103        delete newErrors[questionId];
104        return newErrors;
105      });
106    }
107
108  const validateForm = (): boolean => {

```

Upgrade Source IX

```

109 const newErrors: Record<string, string> = {};
110
111 surveyQuestions.forEach((question) => {
112   if (question.required) {
113     const answer = answers[question.id];
114     if (!answer || (Array.isArray(answer) && answer.length === 0)) {
115       newErrors[question.id] = "필수 항목입니다";
116     } else if (typeof answer === "string" && answer.trim() === "") {
117       newErrors[question.id] = "답변을 입력해주세요";
118     }
119   }
120 });
121
122 setErrors(newErrors);
123 return Object.keys(newErrors).length === 0;
124 };
125
126 const handleSubmit = (e: React.FormEvent) => {
127   e.preventDefault();
128
129   if (!validateForm()) {
130     toast({
131       title: "입력 확인",
132       description: "필수 항목을 모두 입력해주세요.",
133       variant: "destructive",
134     });
135     return;
136   }
137
138 // Here you would typically send the data to a backend
139 console.log("Survey answers:", answers);
140
141 setIsSubmitted(true);
142 toast({
143   title: "제출 완료",
144   description: "소중한 의견 감사합니다!",
145 });
146 }
147
148 if (!hasAgreed) {
149   return (
150     <div className="min-h-screen flex items-center justify-center p-4 bg-background">
151       <AlertDialog open={true}>
152         <AlertDialogContent>
153           <AlertDialogHeader>
154             <AlertDialogTitle className="text-2xl">설문조사에 참여하시겠습니까?</AlertDialogTitle>
155             <AlertDialogDescription className="text-base">
156               고객 만족도 설문조사에 참여해 주세요. 소중한 의견 감사합니다.
157             </AlertDialogDescription>
158             <AlertDialogHeader>
159             <AlertDialogFooter>
160               <AlertDialogCancel>취소</AlertDialogCancel>
161               <AlertDialogAction onClick={() => setHasAgreed(true)}>
162                 확인
163               </AlertDialogAction>
164             <AlertDialogFooter>
165             </AlertDialogContent>
166           </AlertDialog>
167         </div>
168       );
169     }
170
171 if (isSubmitted) {
172   return (
173     <div className="min-h-screen flex items-center justify-center p-4 bg-background">
174       <div className="max-w-2xl w-full text-center space-y-6 animate-in fade-in duration-500">
175         <div className="flex justify-center">
176           <div className="relative">
177             <div className="absolute inset-0 bg-primary/0 blur-xl rounded-full animate-pulse" />
178             <CheckCircle2 className="w-24 h-24 text-primary relative animate-in zoom-in duration-300" />

```

Upgrade Source IX

```

179      </div>
180    </div>
181    <div className="space-y-2">
182      <h2 className="text-3xl font-bold text-foreground">
183        감사합니다!
184      </h2>
185      <p className="text-lg text-muted-foreground">
186        설문에 참여해 주셔서 감사합니다.
187        <br />
188        소중한 의견은 더 나은 서비스를 만드는 데 활용하겠습니다.
189      </p>
190    </div>
191    <Button
192      onClick={() => {
193        setIsSubmitted(false);
194        setAnswers({});
195        setErrors({});
196      }}
197      variant="outline"
198      size="lg"
199      className="transition-all duration-200"
200    >
201      새 설문 작성하기
202    </Button>
203  </div>
204);
205
206}
207
208return (
209  <div className="min-h-screen py-12 px-4 bg-background">
210    <div className="max-w-3xl mx-auto space-y-8 animate-in fade-in duration-500">
211      <header className="text-center space-y-4">
212        <h1 className="text-5xl font-bold text-primary">
213          고객 만족도 설문조사
214        </h1>
215        <p className="text-lg text-muted-foreground">
216          소중한 의견 감사합니다.
217        </p>
218      </header>
219      <form onSubmit={handleSubmit} className="space-y-6">
220        {surveyQuestions.map((question, index) => (
221          <div
222            key={question.id}
223            className="animate-in slide-in-from-bottom duration-500"
224            style={{ animationDelay: `${(index * 100) ms}` }}
225          <QuestionCard
226            question={question.question}
227            type={question.type}
228            options={question.options}
229            required={question.required}
230            value={answers[question.id]}
231            onChange={(value) => handleAnswerChange(question.id, value)}
232            error={errors[question.id]}
233          >
234            </div>
235          ))
236        )
237        <div className="pt-4">
238          <button
239            type="submit"
240            size="lg"
241            className="w-full bg-gradient-to-r from-primary to-primary-glow hover:opacity-90
242            transition-all duration-300 text-primary-foreground font-semibold shadow-lg hover:shadow-xl"
243          >
244            설문 제출하기
245          </button>
246        </div>
247      </form>
248    </div>
249  </div>
250);
251
252  <footer className="text-center text-sm text-muted-foreground pt-8 border-t">
253    <p>설문 작성 시간: 약 3-5분 소요</p>
254  </footer>
255</div>
256</div>
257);

```

Upgrade Source X

Prompt: 각 카드마다 응답 할 때 유효성 검사를 실시간으로 해주고, 유효한 답변이 제출 될 때마다 카드 레이아웃에 심플한 애니메이션 효과를 추가해줘

고객 만족도 설문조사
소중한 의견 감사합니다.

성별을 알려주세요*

성별을 선택해주세요*
 남성
 여성

나이대를 선택해주세요*
 10대
 20대
 30대
 40대
 50대
 60대 이상

전반적인 만족도를 평가해주세요*
 매우 만족
 만족
 보통
 불만족
 매우 불만족

어떤 기능이 가장 유용했나요? (복수 선택 가능)
 직관적인 레이아웃
 빠른 응답 속도
 다양한 기능
 모바일 지원
 고객 지원

개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요

설문 제출하기

설문 화장 시간 약 1분 소요

Upgrade Source X

```

1 import { Card } from "@/components/ui/card";
2 import { Input } from "@/components/ui/input";
3 import { Textarea } from "@/components/ui/textarea";
4 import { RadioGroup, RadiogroupItem } from "@/components/ui/radio-group";
5 import { Checkbox } from "@/components/ui/checkbox";
6 import { Label } from "@/components/ui/label";
7 import { cn } from "@lib/utils";
8 import { Checkcircle2 } from "lucide-react";
9
10 interface QuestionOption {
11   id: string;
12   label: string;
13 }
14
15 interface QuestionCardProps {
16   question: string;
17   type: "text" | "textarea" | "radio" | "checkbox";
18   options?: QuestionOption[];
19   required?: boolean;
20   value?: string | string[];
21   onChange?: (value: string | string[]) => void;
22   error?: string;
23   isValid?: boolean;
24 }
25
26 export const QuestionCard = ({
27   question,
28   type,
29   options,
30   required = false,
31   value,
32   onChange,
33   error,
34   isValid = false,
35 }: QuestionCardProps) => {
36   const handlecheckboxChange = (optionId: string, checked: boolean) => {
37     const currentValues = (value as string[]) || [];

```

```

38   const newValues = checked
39     ? [...currentValues, optionId]
40     : currentValues.filter((v) => v !== optionId);
41   onChange?.(newValues);
42 };
43
44 return (
45   <Card className={cn(
46     "p-6 transition-all duration-300 hover:shadow-[var(--shadow-card-hover)] bg-gradient-to-b from-card to-card/95 border-border/50 relative",
47     isValid && "animate-in scale-in duration-200 border-primary/50"
48   )}>
49   {isValid && (
50     <div className="absolute -top-2 -right-2 animate-in zoom-in duration-300">
51       <div className="relative">
52         <div className="absolute inset-0 bg-primary/30 blur-md rounded-full" />
53         <Checkcircle2 className="w-8 h-8 text-primary bg-background rounded-full relative" />
54       </div>
55     </div>
56   )}
57   <div className="space-y-4">
58     <h3 className="text-lg font-semibold text-card-foreground">
59       {question}
60       {(required && <span className="text-destructive ml-1">*</span>)}
61     </h3>
62     ^
63     {type === "text" && (
64       <Input
65         value={(value as string) || ""}
66         onChange={(e) => onChange?.(e.target.value)}
67         placeholder="답변을 입력하세요"
68         className={cn(
69           "transition-all duration-200",
70           error && "border-destructive focus-visible:ring-destructive"
71         )}
72       />

```

Upgrade Source X

```

73      )}
74
75    ^ {type === "textarea" && (
76      <textarea
77        value={(value as string) || ""}
78        onChange={(e) => onChange?.(e.target.value)}
79        placeholder="답변을 입력하세요"
80        rows={4}
81        className={
82          "transition-all duration-200 resize-none",
83          error && "border-destructive focus-visible:ring-destructive"
84        }
85      />
86    )}
87
88    ^ {type === "radio" && options && (
89      <RadioGroup
90        value={(value as string) || ""}
91        onChange={(val) => onChange?.(val)}
92      >
93        <div className="space-y-3">
94          {options.map((option) => (
95            <div key={option.id} className="flex items-center space-x-3">
96              <RadioGroupItem
97                value={option.id}
98                id={option.id}
99                className="transition-all duration-200"
100             />
101            <label
102              htmlFor={option.id}
103              className="text-card-foreground cursor-pointer font-normal"
104            >
105              {option.label}
106            </label>
107          </div>
108        ))};
109      </div>
110    </RadioGroup>
111  )}
112
113  ^ {type === "checkbox" && options && (
114    <div className="space-y-3">
115      {options.map((option) => (
116        <div key={option.id} className="flex items-center space-x-3">
117          <Checkbox
118            id={option.id}
119            checked={(value as string[]).includes(option.id)}
120            onChange={(checked) =>
121              handleCheckboxChange(option.id, checked as boolean)
122            }
123            className="transition-all duration-200"
124          />
125          <Label
126            htmlFor={option.id}
127            className="text-card-foreground cursor-pointer font-normal"
128          >
129            {option.label}
130          </Label>
131        </div>
132      ))});
133    </div>
134  )}
135
136  ^ {error && (
137    <p className="text-sm text-destructive animate-in fade-in duration-200">
138      {error}
139    </p>
140  )}
141  </div>
142  </Card>
143  );
144};

```

Upgrade Source X

SurveyForm.tsx

```
1 import { useState } from "react";
2 import { Button } from "@/components/ui/button";
3 import { QuestionCard } from "./QuestionCard";
4 import { toast } from "@/hooks/use-toast";
5 import { CheckCircle2 } from "lucide-react";
6 import {
7   AlertDialog,
8   AlertDialogAction,
9   AlertDialogCancel,
10  AlertDialogContent,
11  AlertDialogDescription,
12  AlertDialogFooter,
13  AlertDialogHeader,
14  AlertDialogTitle,
15 } from "@/components/ui/alert-dialog";
16
17 interface Question {
18   id: string;
19   question: string;
20   type: "text" | "textarea" | "radio" | "checkbox";
21   options?: { id: string; label: string }[];
22   required?: boolean;
23 }
24
25 const surveyQuestions: Question[] = [
26 {
27   id: "name",
28   question: "성함을 알려주세요",
29   type: "text",
30   required: true,
31 },
32 {
33   id: "gender",
34   question: "성별을 선택해주세요",
35   type: "radio",
36   options: [
37     { id: "male", label: "남성" },
38     { id: "female", label: "여성" },
39   ],
40   required: true,
41 },
42 {
43   id: "age",
44   question: "나이대를 선택해주세요",
45   type: "radio",
46   options: [
47     { id: "10s", label: "10대" },
48     { id: "20s", label: "20대" },
49     { id: "30s", label: "30대" },
50     { id: "40s", label: "40대" },
51     { id: "50s", label: "50대" },
52     { id: "60s", label: "60대 이상" },
53   ],
54   required: true,
55 },
56 {
57   id: "satisfaction",
58   question: "전반적인 만족도를 평가해주세요",
59   type: "radio",
60   options: [
61     { id: "very-satisfied", label: "매우 만족" },
62     { id: "satisfied", label: "만족" },
63     { id: "neutral", label: "보통" },
64     { id: "dissatisfied", label: "불만족" },
65     { id: "very-dissatisfied", label: "매우 불만족" },
66   ],
67   required: true,
68 },
69 {
70   id: "features",
71   question: "어떤 기능이 가장 유용했나요? (복수 선택 가능)",
72   type: "checkbox",
73   options: [
74     { id: "ui-design", label: "직관적인 UI/UX" },
75     { id: "speed", label: "빠른 로딩 속도" },
76     { id: "features", label: "다양한 기능" },
77     { id: "mobile", label: "모바일 지원" },
78     { id: "customer-service", label: "고객 지원" },
79   ],
80   required: false,
81 },
82 {
83   id: "feedback",
84   question: "개선이 필요한 부분이나 추가 의견을 자유롭게 작성해주세요",
85   type: "textarea",
86   required: false,
87 },
88 ];
89
90 export const SurveyForm = () => {
91   const [answers, setAnswers] = useState<Record<string, string | string[]>({}); // 초기화된 응답 객체
92   const [errors, setErrors] = useState<Record<string, string>({}); // 초기화된 오류 객체
93   const [isSubmitted, setIsSubmitted] = useState(false); // 폼이 제출되었는지 여부
94   const [hasAgreed, setHasAgreed] = useState(false); // 개인정보 수집 동의 여부
95   const [validQuestions, setValidQuestions] = useState<Set<string>>(new Set()); // 유효한 질문 목록
96
97   const validateAnswer = (questionId: string, value: string | string[]): boolean => {
98     const question = surveyQuestions.find(q => q.id === questionId);
99     if (!question) return false;
100
101     if (question.required) {
102       if (!value) return false;
103       if (Array.isArray(value) && value.length === 0) return false;
104       if (typeof value === "string" && value.trim() === "") return false;
105     }
106
107     // 필수가 아닌 경우, 값이 있으면 유효
108     if (!question.required && value) {
109       return true;
110     }
111   }
112
113   const handleAnswerChange = (id: string, value: string | string[]) => {
114     setAnswers((prevAnswers) => ({...prevAnswers, [id]: value}));
115     if (validateAnswer(id, value)) {
116       setValidQuestions((prevValidQuestions) => prevValidQuestions.add(id));
117     }
118   }
119
120   const handleFeedbackChange = (value: string) => {
121     setErrors((prevErrors) => ({...prevErrors, feedback: value}));
122   }
123
124   const handleFormSubmit = (e: React.FormEvent) => {
125     e.preventDefault();
126     if (surveyQuestions.every((q) => validateAnswer(q.id, q.options?.find((o) => o.id === answers[q.id])))) {
127       setIsSubmitted(true);
128       // 서버로 전송하는 로직
129     }
130   }
131
132   return (
133     <div>
134       <Form>
135         <FormSection>
136           <FormSectionItem>
137             <FormSectionItemLabel>성명</FormSectionItemLabel>
138             <FormSectionItemInput><Input type="text" value={answers.name} onChange={handleAnswerChange}></Input></FormSectionItemInput>
139           </FormSectionItem>
140           <FormSectionItem>
141             <FormSectionItemLabel>성별</FormSectionItemLabel>
142             <FormSectionItemInput><RadioGroup checked={answers.gender === "male"} value="male" onChange={handleAnswerChange}><Radio value="male"/><Radio value="female"/></RadioGroup></FormSectionItemInput>
143           </FormSectionItem>
144           <FormSectionItem>
145             <FormSectionItemLabel>나이대</FormSectionItemLabel>
146             <FormSectionItemInput><RadioGroup checked={answers.age === "10s"} value="10s" onChange={handleAnswerChange}><Radio value="10s"/><Radio value="20s"/><Radio value="30s"/><Radio value="40s"/><Radio value="50s"/><Radio value="60s"/></RadioGroup></FormSectionItemInput>
147           </FormSectionItem>
148           <FormSectionItem>
149             <FormSectionItemLabel>만족도</FormSectionItemLabel>
150             <FormSectionItemInput><RadioGroup checked={answers.satisfaction === "very-satisfied"} value="very-satisfied" onChange={handleAnswerChange}><Radio value="very-satisfied"/><Radio value="satisfied"/><Radio value="neutral"/><Radio value="dissatisfied"/><Radio value="very-dissatisfied"/></RadioGroup></FormSectionItemInput>
151           </FormSectionItem>
152           <FormSectionItem>
153             <FormSectionItemLabel>기능 유용성</FormSectionItemLabel>
154             <FormSectionItemInput><Checklist checked={answers.features === "ui-design"} value="ui-design" onChange={handleAnswerChange}><ChecklistItem value="ui-design"/><ChecklistItem value="speed"/><ChecklistItem value="features"/><ChecklistItem value="mobile"/><ChecklistItem value="customer-service"/></Checklist></FormSectionItemInput>
155           </FormSectionItem>
156           <FormSectionItem>
157             <FormSectionItemLabel>제작자에게 전달</FormSectionItemLabel>
158             <FormSectionItemInput><Textarea value={answers.feedback} onChange={handleFeedbackChange}></Textarea></FormSectionItemInput>
159           </FormSectionItem>
160         </FormSection>
161         <FormSection>
162           <FormSectionItem>
163             <FormSectionItemLabel>개인정보 수집 동의</FormSectionItemLabel>
164             <FormSectionItemInput><Switch checked={hasAgreed} onChange={() => setHasAgreed(!hasAgreed)}></Switch></FormSectionItemInput>
165           </FormSectionItem>
166         </FormSection>
167       </Form>
168       <FormActions>
169         <FormAction>제출</FormAction>
170       </FormActions>
171     </div>
172   );
173 }
```

Upgrade Source X

```

109  if (Array.isArray(value)) return value.length > 0;
110  if (typeof value === "string") return value.trim().length > 0;
111 }
112
113 return question.required ? true : !!value;
114 };
115
116 const handleAnswerChange = (questionId: string, value: string | string[]) => {
117   setAnswers((prev) => ({ ...prev, [questionId]: value }));
118
119 // 실시간 유효성 검사
120 const isValid = validateAnswer(questionId, value);
121
122 if (isValid) {
123   setValidQuestions((prev) => {
124     const newSet = new Set(prev);
125     newSet.add(questionId);
126     return newSet;
127   });
128 } else {
129   setValidQuestions((prev) => {
130     const newSet = new Set(prev);
131     newSet.delete(questionId);
132     return newSet;
133   });
134 }
135
136 // Clear error when user starts typing
137 if (errors[questionId]) {
138   setErrors((prev) => {
139     const newErrors = { ...prev };
140     delete newErrors[questionId];
141     return newErrors;
142   });
143 }
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
}
const validateForm = (): boolean => {
  const newErrors: Record<string, string> = {};
  surveyQuestions.forEach((question) => {
    if (question.required) {
      const answer = answers[question.id];
      if (!answer || (Array.isArray(answer) && answer.length === 0)) {
        newErrors[question.id] = "필수 항목입니다";
      } else if (typeof answer === "string" && answer.trim() === "") {
        newErrors[question.id] = "답변을 입력해주세요";
      }
    }
  });
  setErrors(newErrors);
  return Object.keys(newErrors).length === 0;
}

const handleSubmit = (e: React.FormEvent) => {
  e.preventDefault();
  if (!validateForm()) {
    toast({
      title: "입력 확인",
      description: "필수 항목을 모두 입력해주세요.",
      variant: "destructive",
    });
    return;
  }
  // Here you would typically send the data to a backend
  console.log("Survey answers:", answers);
  setIsSubmitted(true);
}

toast({
  title: "제출 완료",
  description: "소중한 의견 감사합니다!",
});

if (!hasAgreed) {
  return (
    <div className="min-h-screen flex items-center justify-center p-4 bg-background">
      <AlertDialog open={true}>
        <AlertDialogContent>
          <AlertDialogHeader>
            <AlertDialogTitle className="text-2xl">설문조사에 참여하시겠습니까?</AlertDialogTitle>
            <AlertDialogDescription className="text-base">고객 만족도 설문조사에 참여해 주세요. 소중한 의견 감사합니다.</AlertDialogDescription>
          </AlertDialogHeader>
          <AlertDialogFooter>
            <AlertDialogCancel>취소</AlertDialogCancel>
            <AlertDialogAction onClick={() => setHasAgreed(true)}>확인</AlertDialogAction>
          </AlertDialogFooter>
        </AlertDialogContent>
      </AlertDialog>
    </div>
  );
}

if (isSubmitted) {
  return (
    <div className="min-h-screen flex items-center justify-center p-4 bg-background">
      <div className="max-w-2x w-full text-center space-y-6 animate-in fade-in duration-500">
        <div className="flex justify-center">
          <div className="relative">

```

Upgrade Source X

SurveyForm.tsx

```
195 <div className="absolute inset-0 bg-primary/20 blur-xl rounded-full animate-pulse" />
196 <CheckCircle2 className="w-24 h-24 text-primary relative animate-in zoom-in duration-300" />
197 </div>
198 </div>
199 <div className="space-y-2">
200   <h2 className="text-3xl font-bold text-primary">
201     감사합니다!
202   </h2>
203   <p className="text-lg text-muted-foreground">
204     설문에 참여해 주셔서 감사합니다.
205   <br />
206   <p>소중한 의견은 더 나은 서비스를 만드는 데 활용하겠습니다.</p>
207 </div>
208 <Button
209   onClick={() => {
210     setIsSubmitted(false);
211     setAnswers({});
212     setErrors({});
213   }}
214   variant="outline"
215   size="lg"
216   className="transition-all duration-200"
217 >
218   새 설문 작성하기
219 </Button>
220 </div>
221 </div>
222 );
223
224 return (
225   <div className="min-h-screen py-12 px-4 bg-background">
226     <div className="max-w-3xl mx-auto space-y-8 animate-in fade-in duration-500">
227       <header className="text-center space-y-4">
228         <h1 className="text-5xl font-bold text-primary">
229           고객 만족도 설문조사
230         </h1>
231         <p className="text-lg text-muted-foreground">
232           소중한 의견 감사합니다.
233         </p>
234       </header>
235       <form onSubmit={handleSubmit} className="space-y-6">
236         {surveyQuestions.map((question, index) => (
237           <div key={question.id} className="animate-in slide-in-from-bottom duration-500" style={{ animationDelay: `${(index * 100)ms}` }}>
238             <QuestionCard
239               question={question.question}
240               type={question.type}
241               options={question.options}
242               required={question.required}
243               value={answers[question.id]}
244               onChange={(value) => handleAnswerChange(question.id, value)}
245               error={errors[question.id]}
246               isValid={isValidQuestions.has(question.id)}
247             />
248           </div>
249         )));
250       <div className="pt-4">
251         <button
252           type="submit"
253           size="lg"
254           className="w-full bg-gradient-to-r from-primary to-primary-glow hover:opacity-90 transition-all duration-300 text-primary-foreground font-semibold shadow-lg hover:shadow-2xl">
255           설문 제출하기
256         </button>
257       </div>
258     </form>
259   </div>
260 </div>
261 <Footer className="text-center text-sm text-muted-foreground pt-8 border-t">
262   <p>설문 작성 시간: 약 3-5분 소요</p>
263 </Footer>
264 </div>
265 </div>
266 </div>
267 </div>
268 </div>
269 </div>
270 </div>
271 </div>
272 </div>
273 </div>
274 </div>
275 </div>
276 </div>
277 </div>
278 </div>
279 </div>
280 </div>
281 </div>
282 </div>
283 </div>
284 </div>
285 </div>
286 </div>
287 </div>
288 </div>
289 </div>
290 </div>
291 </div>
292 </div>
293 </div>
294 </div>
295 </div>
296 </div>;
297 );
```

Thank you