

```
import { useState, useRef } from 'react';
import { Card, CardContent } from "@components/ui/card";
import { Button } from "@components/ui/button";
import { Slider } from "@components/ui/slider";
import { Label } from "@components/ui/label";
import { Input } from "@components/ui/input";
import { BarChart, Bar, XAxis, YAxis, Tooltip, ResponsiveContainer } from 'recharts';
```

```
const initialFollowers = [
  { id: 1, name: "Researcher", role: "Researcher", trust: 50 },
  { id: 2, name: "Strategist", role: "Strategist", trust: 50 },
  { id: 3, name: "Executor", role: "Executor", trust: 50 }
];
```

```
const taskType = "Collaboratively create a presentation outline";
const orgContexts = [];
```

```
const getRange = (value) => {
  if (value < 20) return "Very Low";
  if (value < 40) return "Low";
  if (value < 60) return "Mid";
  if (value < 80) return "High";
  return "Very High";
};
```

```
const transcripts = {
  "Very Low|Very Low": `Leader: 'Okay, um... maybe we start with the intro?'
  Researcher: 'You're not sure either, huh? Let's figure it out together.'
  Strategist: 'We need some kind of roadmap, even a rough one.'
  Executor: 'I'll put something together, but we're gonna need more direction.'`,
```

```
  "Very Low|Low": `Leader: 'Anyone have a strong idea on how to begin?'
  Researcher: 'I can throw out some research, but what are we aiming for?'
  Strategist: 'You're trying, but we're missing a compass.'
  Executor: 'I'll wait till the picture's a little clearer.'`,
```

```
  "Very Low|Mid": `Leader: 'Let's take this one step at a time and see where it goes.'
  Researcher: 'Alright, I'll dig up a few relevant sources.'
  Strategist: 'We could build around that if we stay flexible.'
  Executor: 'Okay, I'll get a draft started based on what we've got.'`,
```

```
  "Very Low|High": `Leader: 'We're gonna make something really solid here!'
  Researcher: 'Love your spirit—just need more direction.'
  Strategist: 'If we align that enthusiasm with a plan, we're good.'`
```

Executor: 'I'm in—let's figure out the details as we go.' ,

"Very Low|Very High": `Leader: 'This is going to blow everyone away!'

Researcher: 'That's a bold promise—do we know what we're presenting?'

Strategist: 'The energy's there, let's anchor it with structure.'

Executor: 'I'll follow your lead—just make sure there's something to follow.' ,

"Low|Very Low": `Leader: 'Let's just get something down? Maybe slides?'

Researcher: 'Hmm, that's not much to go on.'

Strategist: 'I think we all need a bit more clarity.'

Executor: 'I'll take a stab in the dark.' ,

"Low|Low": `Leader: 'What's a good place to begin here?'

Researcher: 'You're listening, that's something.'

Strategist: 'Let's agree on a goal so we're not guessing.'

Executor: 'I'm ready once we commit to a plan.' ,

"Low|Mid": `Leader: 'What if we frame it around user needs?'

Researcher: 'Good idea—I'll find examples to back that.'

Strategist: 'Nice start. I'll sketch the outline.'

Executor: 'Cool, I'll start putting together slides.' ,

"Low|High": `Leader: 'I've got a good feeling about this—we can do it.'

Researcher: 'That helps. I'll get right on the data.'

Strategist: 'Let's shape this confidence into a clear outline.'

Executor: 'I'm building as we speak.' ,

"Low|Very High": `Leader: 'We've totally got this. Let's blow them away.'

Researcher: 'Energy's contagious—just need specifics.'

Strategist: 'Let's turn this momentum into something solid.'

Executor: 'I'm moving, but I hope the foundation holds.' ,

"Mid|Very Low": `Leader: 'Here's a rough outline to kick us off.'

Researcher: 'Solid start—maybe speak up a bit more?'

Strategist: 'We're moving, just need a bit more push.'

Executor: 'I'm waiting to see where we land.' ,

"Mid|Low": `Leader: 'Let's get rolling on this basic structure.'

Researcher: 'Got it—I'll gather some references.'

Strategist: 'I can build on this, just tweak as we go.'

Executor: 'Starting the first draft now.' ,

"Mid|Mid": `Leader: 'Here's our outline, we'll polish as we go.'

Researcher: 'Perfect. I've got a few sources already.'

Strategist: 'Structure's clean—I'll strengthen transitions.'

Executor: 'This gives me what I need to get started.'",

"Mid|High": `Leader: 'We've got momentum—let's keep it up.'

Researcher: 'I'm pulling stats right now.'

Strategist: 'You're making it easy to stay aligned.'

Executor: 'Already building off what you laid out.'",

"Mid|Very High": `Leader: 'We're absolutely going to nail this.'

Researcher: 'I trust your lead—I'm ready.'

Strategist: 'Your clarity is helping me go deeper.'

Executor: 'Design is flowing smoothly now.'",

"High|Very Low": `Leader: 'Here's a model that should work.'

Researcher: 'It looks great—just say it with more confidence.'

Strategist: 'We're with you—bring us in more.'

Executor: 'I'll tread carefully until you speak up more.'",

"High|Low": `Leader: 'This structure should guide us.'

Researcher: 'Strong framework—I'll get supporting info.'

Strategist: 'We're off to a good start.'

Executor: 'Working right along with you.'",

"High|Mid": `Leader: 'Everything connects here—check it out.'

Researcher: 'Perfect, I'll match the data to your points.'

Strategist: 'Love this flow—refining as we go.'

Executor: 'Already on slide 3.'",

"High|High": `Leader: 'Let's raise the bar on this one.'

Researcher: 'Already pulling some stellar sources.'

Strategist: 'Clear and smooth—I'll bring in depth.'

Executor: 'Design's crisp, just like your plan.'",

"High|Very High": `Leader: 'We're going to deliver something amazing.'

Researcher: 'I feel it—your plan is airtight.'

Strategist: 'Everything lines up—fine-tuning now.'

Executor: 'Presentation's going to look sharp.'",

"Very High|Very Low": `Leader: 'Here's a full model I mapped out.'

Researcher: 'This is genius—just explain it more clearly.'

Strategist: 'Help us see the vision like you do.'

Executor: 'I'm holding off until I get the green light.'",

"Very High|Low": `Leader: 'Let's run with this structure I built.'

Researcher: 'You've got gold here—I'll support it.'
Strategist: 'Let's tighten it up together.'
Executor: 'Just give me the go, I'm ready.'",

"Very High|Mid": `Leader: 'All the parts are connected here.'
Researcher: 'Looks sharp—I'll back it with evidence.'
Strategist: 'Refining flow as we go.'
Executor: 'Already designing—it's solid.'",

"Very High|High": `Leader: 'This project is dialed in—let's finish strong.'
Researcher: 'Your clarity makes research easy.'
Strategist: 'Polish is going smoothly.'
Executor: 'This is flowing beautifully.'",

"Very High|Very High": `Leader: 'Let's make something we're proud of.'
Researcher: 'Everything is spot on—I'm pushing it even further.'
Strategist: 'We're in perfect sync—I'm loving this.'
Executor: 'Slides look amazing already.'"
};

```
export default function LeaderEffectivenessApp() {
  const [competence, setCompetence] = useState(50);
  const [confidence, setConfidence] = useState(50);
  const [context, setContext] = useState("");
  const [hypothesis, setHypothesis] = useState("High competence and confidence leaders will foster better outline quality and team trust.");
  const [result, setResult] = useState("");
  const [successScore, setSuccessScore] = useState(0);
  const [outlineQuality, setOutlineQuality] = useState(0);
  const [ideaRichness, setIdeaRichness] = useState("");
  const [followers, setFollowers] = useState(initialFollowers);
  const [records, setRecords] = useState([]);
  const [transcript, setTranscript] = useState("");
  const lastInputsRef = useRef({});

  const getIdeaRichness = (score) => {
    if (score >= 80) return "Ideas were diverse and highly creative.";
    if (score >= 60) return "Ideas were varied with occasional creativity.";
    if (score >= 40) return "Ideas were somewhat repetitive.";
    return "Ideas lacked depth and variety.";
  };

  const generateTranscript = (comp, conf) => {
    const compRange = getRange(comp);
```

```

    const confRange = getRange(conf);
    return transcripts[`${compRange}|${confRange}`] || "Team response undefined for this leadership profile.";
  };

```

```

const evaluateTeamEffectiveness = () => {
  const contextScore = 0;
  const competenceScore = (competence / 100) * 40;
  const confidenceScore = (confidence / 100) * 30;
  const totalScore = Math.round(competenceScore + confidenceScore + contextScore);
  const avgTrust = followers.reduce((sum, f) => sum + f.trust, 0) / followers.length;

```

```

  const qualityScore = Math.round((competence * 0.6 + avgTrust * 0.4));
  const richness = getIdeaRichness(qualityScore);
  const transcriptText = generateTranscript(competence, confidence);

```

```

  let effectiveness = "";
  if (totalScore >= 80) effectiveness = "High team effectiveness.";
  else if (totalScore >= 60) effectiveness = "Moderate team effectiveness.";
  else if (totalScore >= 40) effectiveness = "Low team effectiveness.";
  else effectiveness = "Very poor effectiveness.";

```

```

  setSuccessScore(totalScore);
  setOutlineQuality(qualityScore);
  setIdeaRichness(richness);
  setTranscript(transcriptText);
  setResult(effectiveness);

```

```

  return {
    competence,
    confidence,
    context,
    result: effectiveness,
    successScore: totalScore,
    outlineQuality: qualityScore,
    ideaRichness: richness,
  };
};

```

```

const recordInputs = (label) => {
  const evaluation = evaluateTeamEffectiveness();
  const newRecord = {
    ...evaluation,
    label: label || `Record ${records.length + 1}`
  };

```

```

    };
    setRecords(prev => [...prev, newRecord]);
  };

const simulateScenario = (comp, conf, label) => {
  setCompetence(comp);
  setConfidence(conf);
  setTimeout(() => {
    const evaluation = evaluateTeamEffectiveness();
    const newRecord = {
      ...evaluation,
      label
    };
    setRecords(prev => [...prev, newRecord]);
  }, 500);
};

const chartData = (r) => [
  { name: 'Success', value: r.successScore },
  { name: 'Outline Quality', value: r.outlineQuality }
];

return (
  <div className="p-6 max-w-6xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6">
    <div>
      <h1 className="text-2xl font-bold mb-4">Leadership Simulation: Presentation
Creation</h1>
      <Card className="mb-6">
        <CardContent className="space-y-4">
          <Label>Hypothesis:</Label>
          <textarea
            className="w-full p-2 border rounded"
            value={hypothesis}
            onChange={(e) => setHypothesis(e.target.value)}
            rows={2}
          />
          <Label>Leader Competence: {competence}</Label>
          <Slider defaultValue={[50]} max={100} step={1} onValueChange={([v]) =>
setCompetence(v)} />
          <Label>Leader Confidence: {confidence}</Label>
          <Slider defaultValue={[50]} max={100} step={1} onValueChange={([v]) =>
setConfidence(v)} />

          <div className="flex gap-4">

```

```

        <Button onClick={() => evaluateTeamEffectiveness()}>Evaluate</Button>
        <Button variant="outline" onClick={() => recordInputs()}>Record</Button>
      </div>
    </CardContent>
  </Card>
  <Card>
    <CardContent>
      <h2 className="font-semibold mb-2">Results:</h2>
      <p className="mb-1">{result}</p>
      <p className="mb-1">Presentation Outline Quality Score: {outlineQuality}/100</p>
      <p className="mb-1">Idea Richness: {ideaRichness}</p>
      <p className="text-sm text-gray-700">{taskType}</p>
      {transcript && (
        <div className="mt-4 p-2 border rounded bg-gray-50">
          <h3 className="font-semibold mb-1">Post-Simulation Transcript:</h3>
          <pre className="text-sm whitespace-pre-wrap">{transcript}</pre>
        </div>
      )}
    </CardContent>
  </Card>
</div>
<div>
  <h2 className="text-xl font-semibold mb-4">Experiment Tracker</h2>
  {records.map((r, i) => (
    <Card key={i} className="mb-4">
      <CardContent className="space-y-2">
        <h3 className="font-bold">{r.label}</h3>
        <p>Competence: {r.competence}, Confidence: {r.confidence}</p>

        <ResponsiveContainer width="100%" height={200}>
          <BarChart data={chartData(r)}>
            <XAxis dataKey="name" /><YAxis domain={[0, 100]} /><Tooltip />
            <Bar dataKey="value" />
          </BarChart>
        </ResponsiveContainer>
        <p className="text-sm italic">{r.result}</p>
        <p className="text-xs">Outline Quality: {r.outlineQuality}/100</p>
        <p className="text-xs">Idea Richness: {r.ideaRichness}</p>

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

```

```
);  
}
```