import React, { useState } from 'react';

import { Button } from '@/components/ui/button';

import { Card, CardContent, CardHeader, CardTitle } from '@/components/ui/card';

import { Progress } from '@/components/ui/progress';

import { DNAType } from './QuizContainer';

interface Question {

id: string;

text: string;

options: {

text: string;

type: 'correct' | 'blurred' | 'neutral' | 'opt-out';

}[];

}

const getQuestionsForDNA = (dnaType: DNAType): Question[] => {

const baseQuestions = [

{

id: 'Q7',

text: "You're about to launch a new offer in 2 weeks. How would your opposite decide what to focus on first?"

},

{

id: 'Q8',

text: 'A peer gives you tough feedback on your recent launch. How would your opposite respond?'

},

{

id: 'Q9',

text: "You've brought on a new team member who isn't delivering as expected. How would your opposite handle it?"

},

{

id: 'Q10',

text: 'A long-term partnership feels exciting but unclear. How would your opposite approach the next step?'

},

{

id: 'Q11',

text: 'A team member keeps missing deadlines. How would your opposite handle this?'

},

{

id: 'Q12',

text: 'You just wrapped a high-revenue campaign. What would your opposite focus on next?'

}

];

if (dnaType === 'Alchemist') {

return [

{

id: 'Q7',

text: baseQuestions[0].text,

options: [

{ text: 'Identify the core bottlenecks and map out the sequence to remove them', type: 'correct' },

{ text: 'Jump between setup, content, and branding — depending on what feels most urgent', type: 'blurred' },

{ text: 'Ask others what they think should happen first and go with consensus', type: 'neutral' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q8',

text: baseQuestions[1].text,

options: [

{ text: 'Shrug it off publicly but internalize it deeply and avoid the peer next time', type: 'blurred' },

{ text: 'Say thank you, then try to adjust only the parts you agree with emotionally', type: 'neutral' },

{ text: 'Break down the feedback, sort it into categories, and adjust where it makes sense', type: 'correct' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q9',

text: baseQuestions[2].text,

options: [

{ text: 'Overcompensate by taking on their tasks and quietly hoping they improve', type: 'blurred' },

{ text: 'Give vague feedback while focusing on keeping morale high', type: 'neutral' },

{ text: 'Review expectations, pinpoint where the breakdown occurred, and restructure their workflow', type: 'correct' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q10',

text: baseQuestions[3].text,

options: [

{ text: 'Wait to see how things evolve and trust it will clarify naturally', type: 'neutral' },

{ text: 'Say yes quickly, then figure it out as they go', type: 'blurred' },

{ text: 'Clarify objectives, roles, and measurable outcomes before committing', type: 'correct' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q11',

text: baseQuestions[4].text,

options: [

{ text: 'Set up a review system to track accountability and outline next steps', type: 'correct' },

{ text: 'Avoid confrontation and work around them instead', type: 'blurred' },

{ text: 'Assume they need more motivation and give them a pep talk', type: 'neutral' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q12',

text: baseQuestions[5].text,

options: [

{ text: 'Break down what worked and rebuild the system for repeatability', type: 'correct' },

{ text: 'Start planning a new direction without reviewing results', type: 'blurred' },

{ text: 'Celebrate with the team and move on instinctively', type: 'neutral' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

}

];

}

if (dnaType === 'Architect') {

return [

{

id: 'Q7',

text: baseQuestions[0].text,

options: [

{ text: 'Get clear on the emotional pull behind the offer so it feels exciting to build', type: 'correct' },

{ text: 'Focus on fixing small issues even if they are not critical to launch', type: 'blurred' },

{ text: 'Stick to what was originally planned even if the context has changed', type: 'neutral' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q8',

text: baseQuestions[1].text,

options: [

{ text: 'Review the feedback later, but only if it aligns with the data', type: 'neutral' },

{ text: 'Get defensive or withdraw from sharing in future launches', type: 'blurred' },

{ text: 'Reflect on whether there is emotional truth in the feedback, not just facts', type: 'correct' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q9',

text: baseQuestions[2].text,

options: [

{ text: 'Point out the issue once, then assume they will adjust without further input', type: 'neutral' },

{ text: 'Replace the person quickly to avoid more damage', type: 'blurred' },

{ text: 'Talk openly about how the dynamic feels and whether something is misaligned', type: 'correct' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q10',

text: baseQuestions[3].text,

options: [

{ text: 'Explore how aligned it feels and notice what is not being said', type: 'correct' },

{ text: 'Ask for data but delay the decision indefinitely', type: 'neutral' },

{ text: 'Say no out of hesitation or fear of complexity', type: 'blurred' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q11',

text: baseQuestions[4].text,

options: [

{ text: 'Replace them quickly — no tolerance for inconsistency', type: 'blurred' },

{ text: 'Check in to understand what is going on behind the scenes', type: 'correct' },

{ text: 'Add more structure and double their workload to see if they step up', type: 'neutral' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q12',

text: baseQuestions[5].text,

options: [

{ text: 'Dissect every line of performance data to fix bottlenecks', type: 'blurred' },

{ text: 'Pause to reflect on alignment before starting anything else', type: 'correct' },

{ text: 'Jump into the next system while momentum is high', type: 'neutral' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

}

];

}

// Blurred Identity - mix of both styles

return [

{

id: 'Q7',

text: baseQuestions[0].text,

options: [

{ text: 'Identify the core bottlenecks and map out the sequence to remove them', type: 'correct' },

{ text: 'Get clear on the emotional pull behind the offer so it feels exciting to build', type: 'correct' },

{ text: 'Jump between setup, content, and branding — depending on what feels most urgent', type: 'blurred' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q8',

text: baseQuestions[1].text,

options: [

{ text: 'Reflect on whether there is emotional truth in the feedback, not just facts', type: 'correct' },

{ text: 'Break down the feedback, sort it into categories, and adjust where it makes sense', type: 'correct' },

{ text: 'Get defensive or withdraw from sharing in future launches', type: 'blurred' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q9',

text: baseQuestions[2].text,

options: [

{ text: 'Talk openly about how the dynamic feels and whether something is misaligned', type: 'correct' },

{ text: 'Review expectations, pinpoint where the breakdown occurred, and restructure their workflow', type: 'correct' },

{ text: 'Overcompensate by taking on their tasks and quietly hoping they improve', type: 'blurred' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q10',

text: baseQuestions[3].text,

options: [

{ text: 'Clarify objectives, roles, and measurable outcomes before committing', type: 'correct' },

{ text: 'Explore how aligned it feels and notice what is not being said', type: 'correct' },

{ text: 'Say yes quickly, then figure it out as they go', type: 'blurred' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q11',

text: baseQuestions[4].text,

options: [

{ text: 'Check in to understand what is going on behind the scenes', type: 'correct' },

{ text: 'Set up a review system to track accountability and outline next steps', type: 'correct' },

{ text: 'Avoid confrontation and work around them instead', type: 'blurred' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

},

{

id: 'Q12',

text: baseQuestions[5].text,

options: [

{ text: 'Pause to reflect on alignment before starting anything else', type: 'correct' },

{ text: 'Break down what worked and rebuild the system for repeatability', type: 'correct' },

{ text: 'Start planning a new direction without reviewing results', type: 'blurred' },

{ text: 'None of these reflect how my opposite would operate', type: 'opt-out' }

]

}

];

};

interface Props {

defaultDNA: DNAType;

onComplete: (score: number) => void;

recordAnswer: (questionId: string, answer: string) => void;

}

const AwarenessBlock: React.FC<Props> = ({ defaultDNA, onComplete, recordAnswer }) => {

const [currentQuestion, setCurrentQuestion] = useState(0);

const [score, setScore] = useState(0);

const [isAnalyzing, setIsAnalyzing] = useState(false);

const questions = getQuestionsForDNA(defaultDNA);

const handleAnswer = (option: { text: string; type: string }) => {

const questionId = questions[currentQuestion].id;

recordAnswer(questionId, option.type);

const newScore = option.type === 'correct' ? score + 1 : score;

setScore(newScore);

if (currentQuestion < questions.length - 1) {

setTimeout(() => setCurrentQuestion(currentQuestion + 1), 300);

} else {

setIsAnalyzing(true);

setTimeout(() => {

onComplete(newScore);

}, 2000);

}

};

const getAwarenessLevel = (score: number, total: number) => {

if (score === total) return 70;

if (score === total - 1) return 60;

if (score === total - 2) return 50;

if (score === total - 3) return 40;

if (score === total - 4) return 30;

return 20;

};

const progress = ((currentQuestion + 1) / questions.length) \* 100;

if (isAnalyzing) {

const awarenessLevel = getAwarenessLevel(score, questions.length);

return (

<CardContent className="p-12 text-center">

<div className="mb-8">

<div className="animate-spin rounded-full h-16 w-16 border-b-2 border-purple-600 mx-auto mb-4"></div>

<h2 className="text-2xl font-bold text-gray-800 mb-2">

Detecting awareness levels of the opposite…

</h2>

<div className="mt-6">

<div className="text-lg font-semibold mb-2">Awareness Level: {awarenessLevel}%</div>

<Progress value={awarenessLevel} className="w-full h-3" />

</div>

</div>

</CardContent>

);

}

const oppositeType = defaultDNA === 'Architect' ? 'Alchemist' :

defaultDNA === 'Alchemist' ? 'Architect' : 'Opposite';

return (

<div className="p-8">

<CardHeader className="text-center pb-6">

<CardTitle className="text-2xl font-bold text-gray-800 mb-2">

Detecting awareness levels of the opposite…

</CardTitle>

<p className="text-gray-600 mb-4">

How well do you understand the {oppositeType} mindset?

</p>

<div className="mb-4">

<Progress value={progress} className="w-full h-2" />

<p className="text-sm text-gray-600 mt-2">

Question {currentQuestion + 1} of {questions.length}

</p>

</div>

</CardHeader>

<CardContent>

<div className="mb-8">

<h3 className="text-xl font-semibold text-gray-800 mb-6 leading-relaxed">

🧭 {questions[currentQuestion].text}

</h3>

<div className="space-y-3">

{questions[currentQuestion].options.map((option, index) => (

<Button

key={index}

variant="outline"

className="w-full p-6 text-left justify-start hover:scale-[1.02] transition-all duration-200 hover:shadow-md"

onClick={() => handleAnswer(option)}

>

<div className="flex items-center">

<div className={`w-3 h-3 rounded-full mr-4 ${

option.type === 'correct' ? 'bg-green-500' :

option.type === 'blurred' ? 'bg-red-500' :

option.type === 'neutral' ? 'bg-gray-400' : 'bg-yellow-500'

}`}></div>

<span className="text-base">{option.text}</span>

</div>

</Button>

))}

</div>

</div>

</CardContent>

</div>

);

};

export default AwarenessBlock;