import React, { useState } from 'react';

import { Card } from '@/components/ui/card';

import DefaultDNABlock from './DefaultDNABlock';

import AwarenessBlock from './AwarenessBlock';

import PathChoiceBlock from './PathChoiceBlock';

import SubtypeBlock from './SubtypeBlock';

import ValidationBlock from './ValidationBlock';

import ResultsPage from './ResultsPage';

export type DNAType = 'Architect' | 'Alchemist' | 'Blurred';

export type PathType = 'Early' | 'Developed';

export type SubtypeType = 'Master Strategist' | 'Systemised Builder' | 'Internal Analyzer' | 'Ultimate Strategist' |

'Visionary Oracle' | 'Magnetic Perfectionist' | 'Energetic Empath' | 'Ultimate Alchemist' |

'Overthinker' | 'Performer' | 'Self-Forsaker' | 'Self-Betrayer';

export interface QuizState {

currentBlock: number;

defaultDNA: DNAType | null;

awarenessScore: number;

pathChoice: PathType | null;

subtype: SubtypeType | null;

subtypeProgress: number;

answers: Record<string, string>;

}

const QuizContainer = () => {

const [quizState, setQuizState] = useState<QuizState>({

currentBlock: 1,

defaultDNA: null,

awarenessScore: 0,

pathChoice: null,

subtype: null,

subtypeProgress: 0,

answers: {}

});

const updateQuizState = (updates: Partial<QuizState>) => {

setQuizState(prev => ({ ...prev, ...updates }));

};

const recordAnswer = (questionId: string, answer: string) => {

setQuizState(prev => ({

...prev,

answers: { ...prev.answers, [questionId]: answer }

}));

};

const nextBlock = () => {

setQuizState(prev => ({ ...prev, currentBlock: prev.currentBlock + 1 }));

};

const renderCurrentBlock = () => {

switch (quizState.currentBlock) {

case 1:

return (

<DefaultDNABlock

onComplete={(dna) => {

updateQuizState({ defaultDNA: dna });

nextBlock();

}}

recordAnswer={recordAnswer}

/>

);

case 2:

return (

<AwarenessBlock

defaultDNA={quizState.defaultDNA!}

onComplete={(score) => {

updateQuizState({ awarenessScore: score });

nextBlock();

}}

recordAnswer={recordAnswer}

/>

);

case 3:

return (

<PathChoiceBlock

onComplete={(path) => {

updateQuizState({ pathChoice: path });

nextBlock();

}}

/>

);

case 4:

return (

<SubtypeBlock

defaultDNA={quizState.defaultDNA!}

pathChoice={quizState.pathChoice!}

onComplete={(subtype, progress) => {

updateQuizState({ subtype, subtypeProgress: progress });

nextBlock();

}}

recordAnswer={recordAnswer}

/>

);

case 5:

return (

<ValidationBlock

subtype={quizState.subtype!}

onComplete={() => nextBlock()}

recordAnswer={recordAnswer}

/>

);

case 6:

return <ResultsPage quizState={quizState} />;

default:

return <div>Quiz Complete</div>;

}

};

return (

<div className="min-h-screen bg-gradient-to-br from-purple-50 via-blue-50 to-indigo-100 py-8 px-4">

<div className="max-w-4xl mx-auto">

<div className="text-center mb-8">

<h1 className="text-4xl font-bold bg-gradient-to-r from-purple-600 to-blue-600 bg-clip-text text-transparent mb-2">

ENTREPRENEURIAL DNA QUIZ™

</h1>

<p className="text-gray-600">Discover your unique entrepreneurial blueprint</p>

</div>

<Card className="backdrop-blur-sm bg-white/90 shadow-xl border-0">

{renderCurrentBlock()}

</Card>

</div>

</div>

);

};

export default QuizContainer;