import { useState, useEffect } from 'react';

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

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

import { Badge } from '@/components/ui/badge';

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

import { useNavigate, useSearchParams } from 'react-router-dom';

import {

Trophy,

Award,

BookOpen,

CheckCircle,

Clock,

Star,

ArrowLeft,

Download,

Share2,

Calendar,

Target

} from 'lucide-react';

import {

checkCertificateEligibility,

awardCertificate,

getStudentCertificates,

certificateLevels,

type CertificateLevel

} from '@/data/progressiveLessons';

interface StudentCertificate {

id: string;

name: string;

awardedAt: string;

level: string;

}

interface SubjectProgress {

[lessonId: string]: {

completed: boolean;

score: number;

};

}

interface StudentProgress {

[subject: string]: SubjectProgress;

}

export default function CambridgeCertification() {

const navigate = useNavigate();

const [searchParams] = useSearchParams();

const studentName = searchParams.get('student') || localStorage.getItem('studentName') || 'Student';

const [eligibleCertificates, setEligibleCertificates] = useState<CertificateLevel[]>([]);

const [studentCertificates, setStudentCertificates] = useState<StudentCertificate[]>([]);

const [loading, setLoading] = useState(true);

useEffect(() => {

const loadCertificateData = () => {

const eligible = checkCertificateEligibility(studentName);

const awarded = getStudentCertificates(studentName);

setEligibleCertificates(eligible);

setStudentCertificates(awarded);

setLoading(false);

};

loadCertificateData();

}, [studentName]);

const handleClaimCertificate = (certificateId: string) => {

const success = awardCertificate(studentName, certificateId);

if (success) {

// Refresh data

const awarded = getStudentCertificates(studentName);

setStudentCertificates(awarded);

setEligibleCertificates(prev => prev.filter(cert => cert.id !== certificateId));

}

};

const getCertificateProgress = (certificate: CertificateLevel) => {

const progressKey = `progress\_${studentName}`;

const studentProgress: StudentProgress = JSON.parse(localStorage.getItem(progressKey) || '{}');

let completedCount = 0;

certificate.requirements.completedLessons.forEach(lessonId => {

// Check if lesson is completed with required score

Object.values(studentProgress).forEach((subjectProgress: SubjectProgress) => {

if (subjectProgress[lessonId]?.completed &&

subjectProgress[lessonId]?.score >= certificate.requirements.minimumScore) {

completedCount++;

}

});

});

return {

completed: completedCount,

total: certificate.requirements.completedLessons.length,

percentage: Math.round((completedCount / certificate.requirements.completedLessons.length) \* 100)

};

};

if (loading) {

return (

<div className="min-h-screen bg-gradient-to-br from-yellow-50 to-orange-50 flex items-center justify-center">

<div className="text-center">

<Trophy className="w-16 h-16 text-yellow-500 mx-auto mb-4 animate-spin" />

<p className="text-xl text-gray-600">Loading certification data...</p>

</div>

</div>

);

}

return (

<div className="min-h-screen bg-gradient-to-br from-yellow-50 via-orange-50 to-red-50">

{/\* Header \*/}

<header className="bg-white shadow-sm border-b-4 border-yellow-300">

<div className="max-w-6xl mx-auto px-6 py-4">

<div className="flex items-center justify-between">

<div className="flex items-center space-x-3">

<Button

onClick={() => navigate('/dashboard')}

variant="outline"

size="sm"

className="mr-4"

>

<ArrowLeft className="w-4 h-4 mr-2" />

Back to Dashboard

</Button>

<div className="w-12 h-12 bg-gradient-to-r from-yellow-500 to-orange-500 rounded-full flex items-center justify-center">

<Trophy className="w-7 h-7 text-white" />

</div>

<div>

<h1 className="text-2xl font-bold text-gray-800">Cambridge YLE Certification</h1>

<p className="text-sm text-gray-600">{studentName}'s Official Certification Portal</p>

</div>

</div>

</div>

</div>

</header>

<main className="max-w-6xl mx-auto px-6 py-8">

{/\* Student Certificates \*/}

{studentCertificates.length > 0 && (

<div className="mb-8">

<h2 className="text-3xl font-bold text-gray-800 mb-6 flex items-center">

<Award className="w-8 h-8 mr-3 text-yellow-500" />

🎉 Your Cambridge Certificates

</h2>

<div className="grid md:grid-cols-2 gap-6">

{studentCertificates.map((cert: StudentCertificate, index: number) => (

<Card key={index} className="border-2 border-yellow-300 bg-gradient-to-r from-yellow-50 to-orange-50 shadow-lg">

<CardHeader>

<div className="flex items-center justify-between">

<div className="flex items-center space-x-3">

<div className="w-16 h-16 bg-gradient-to-r from-yellow-400 to-orange-500 rounded-full flex items-center justify-center">

<Trophy className="w-8 h-8 text-white" />

</div>

<div>

<CardTitle className="text-xl text-gray-800">Cambridge YLE Flyers</CardTitle>

<p className="text-gray-600">Official Certificate</p>

</div>

</div>

<Badge className="bg-green-500 text-white text-lg px-4 py-2">

Certified ✓

</Badge>

</div>

</CardHeader>

<CardContent>

<div className="space-y-4">

<div className="flex items-center justify-between text-sm">

<span className="text-gray-600">Awarded:</span>

<span className="font-medium">{new Date(cert.awardedAt).toLocaleDateString()}</span>

</div>

<div className="flex items-center justify-between text-sm">

<span className="text-gray-600">Level:</span>

<span className="font-medium">A2 (Grade 4)</span>

</div>

<div className="flex space-x-2 pt-4">

<Button className="flex-1 bg-blue-500 hover:bg-blue-600">

<Download className="w-4 h-4 mr-2" />

Download PDF

</Button>

<Button variant="outline" className="flex-1">

<Share2 className="w-4 h-4 mr-2" />

Share

</Button>

</div>

</div>

</CardContent>

</Card>

))}

</div>

</div>

)}

{/\* Ready to Claim Certificates \*/}

{eligibleCertificates.length > 0 && (

<div className="mb-8">

<h2 className="text-3xl font-bold text-gray-800 mb-6 flex items-center">

<Star className="w-8 h-8 mr-3 text-yellow-500" />

🎊 Ready to Claim!

</h2>

<div className="grid md:grid-cols-2 gap-6">

{eligibleCertificates.map((certificate) => (

<Card key={certificate.id} className="border-2 border-green-300 bg-gradient-to-r from-green-50 to-blue-50 shadow-lg">

<CardHeader>

<div className="flex items-center justify-between">

<div className="flex items-center space-x-3">

<div className="w-16 h-16 bg-gradient-to-r from-green-400 to-blue-500 rounded-full flex items-center justify-center">

<Trophy className="w-8 h-8 text-white" />

</div>

<div>

<CardTitle className="text-xl text-gray-800">{certificate.name}</CardTitle>

<p className="text-gray-600">You've met all requirements!</p>

</div>

</div>

</div>

</CardHeader>

<CardContent>

<p className="text-gray-700 mb-4">{certificate.description}</p>

<Button

onClick={() => handleClaimCertificate(certificate.id)}

className="w-full bg-gradient-to-r from-green-500 to-blue-500 hover:from-green-600 hover:to-blue-600 text-white text-lg py-3"

>

<Award className="w-5 h-5 mr-2" />

Claim Your Certificate!

</Button>

</CardContent>

</Card>

))}

</div>

</div>

)}

{/\* Certificate Progress \*/}

<div className="mb-8">

<h2 className="text-3xl font-bold text-gray-800 mb-6 flex items-center">

<Target className="w-8 h-8 mr-3 text-blue-500" />

📈 Certification Progress

</h2>

<div className="grid md:grid-cols-3 gap-6">

{certificateLevels.map((certificate) => {

const progress = getCertificateProgress(certificate);

const isAwarded = studentCertificates.some((cert: StudentCertificate) => cert.id === certificate.id);

const isEligible = eligibleCertificates.some(cert => cert.id === certificate.id);

return (

<Card key={certificate.id} className={`border-2 ${

isAwarded ? 'border-green-300 bg-green-50' :

isEligible ? 'border-yellow-300 bg-yellow-50' :

'border-gray-200 bg-gray-50'

}`}>

<CardHeader>

<div className="flex items-center space-x-3">

<div className={`w-12 h-12 rounded-full flex items-center justify-center ${

isAwarded ? 'bg-green-500' :

isEligible ? 'bg-yellow-500' :

'bg-gray-400'

}`}>

{isAwarded ? (

<CheckCircle className="w-6 h-6 text-white" />

) : (

<Trophy className="w-6 h-6 text-white" />

)}

</div>

<div>

<CardTitle className="text-lg">{certificate.name.replace('Cambridge YLE ', '')}</CardTitle>

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

{isAwarded ? 'Completed!' : isEligible ? 'Ready!' : `${progress.percentage}% Complete`}

</p>

</div>

</div>

</CardHeader>

<CardContent>

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

<Progress value={isAwarded ? 100 : progress.percentage} className="h-2" />

<div className="text-sm text-gray-600">

<p>Lessons: {progress.completed}/{progress.total}</p>

<p>Min Score: {certificate.requirements.minimumScore}%</p>

</div>

{isAwarded && (

<Badge className="bg-green-500 text-white">

✓ Certificate Earned

</Badge>

)}

{isEligible && !isAwarded && (

<Badge className="bg-yellow-500 text-white">

🎉 Ready to Claim!

</Badge>

)}

</div>

</CardContent>

</Card>

);

})}

</div>

</div>

{/\* Information Section \*/}

<Card className="border-2 border-blue-200 bg-blue-50">

<CardHeader>

<CardTitle className="text-2xl text-blue-800 flex items-center">

<BookOpen className="w-6 h-6 mr-3" />

About Cambridge YLE Certification

</CardTitle>

</CardHeader>

<CardContent>

<div className="grid md:grid-cols-2 gap-6">

<div>

<h4 className="font-bold text-blue-800 mb-3">What is Cambridge YLE?</h4>

<ul className="text-blue-700 space-y-2 text-sm">

<li>• <strong>Young Learners English:</strong> Designed for children aged 7-12</li>

<li>• <strong>International Recognition:</strong> Accepted worldwide</li>

<li>• <strong>Fun & Motivating:</strong> Colorful certificates with shields</li>

<li>• <strong>No Pass/Fail:</strong> Everyone gets recognition for their effort</li>

</ul>

</div>

<div>

<h4 className="font-bold text-blue-800 mb-3">Flyers Level (A2)</h4>

<ul className="text-blue-700 space-y-2 text-sm">

<li>• <strong>Perfect for Grade 4:</strong> Age-appropriate content</li>

<li>• <strong>Real-world Skills:</strong> Practical English for daily life</li>

<li>• <strong>Confidence Building:</strong> Prepares for higher levels</li>

<li>• <strong>Next Step:</strong> Ready for Cambridge Key (KET)</li>

</ul>

</div>

</div>

<div className="mt-6 p-4 bg-white rounded-lg border border-blue-200">

<h4 className="font-bold text-blue-800 mb-2">🎓 How to Earn Your Certificate:</h4>

<ol className="text-blue-700 space-y-1 text-sm">

<li>1. Complete all required Cambridge Flyers lessons</li>

<li>2. Score 90% or higher on each lesson</li>

<li>3. Your certificate will automatically become available</li>

<li>4. Click "Claim Certificate" to receive your official recognition</li>

<li>5. Download and share your achievement!</li>

</ol>

</div>

</CardContent>

</Card>

{/\* No Progress Yet \*/}

{studentCertificates.length === 0 && eligibleCertificates.length === 0 && (

<Card className="border-2 border-orange-200 bg-orange-50 mt-8">

<CardHeader>

<CardTitle className="text-2xl text-orange-800 flex items-center">

<Clock className="w-6 h-6 mr-3" />

Start Your Certification Journey!

</CardTitle>

</CardHeader>

<CardContent>

<p className="text-orange-700 mb-4">

You haven't started working towards your Cambridge YLE Flyers certificate yet.

Complete the foundation lessons first, then unlock the Cambridge preparation content!

</p>

<Button

onClick={() => navigate('/dashboard')}

className="bg-orange-500 hover:bg-orange-600 text-white"

>

<BookOpen className="w-4 h-4 mr-2" />

Start Learning Now

</Button>

</CardContent>

</Card>

)}

</main>

</div>

);

}