'use client'

import { useState, useEffect } from 'react'

import { Button } from "@/components/ui/button"

import { Table, TableBody, TableCell, TableHead, TableHeader, TableRow } from "@/components/ui/table"

import { motion, AnimatePresence } from "framer-motion"

import { GraduationCap, CheckCircle, XCircle, Clock, Info, X, BookOpen, Search, FileText } from 'lucide-react'

import { ScrollArea } from "@/components/ui/scroll-area"

import { Input } from "@/components/ui/input"

import Image from 'next/image'

import { Dialog, DialogContent, DialogHeader, DialogTitle, DialogTrigger } from "@/components/ui/dialog"

import { Select, SelectContent, SelectItem, SelectTrigger, SelectValue } from "@/components/ui/select"

const dummyStudents = [

{

name: "Rahul Sharma",

applicationNumber: "APP001",

course: "Computer Science",

status: "Qualified",

documents: [

{ name: "10th Certificate", type: "PDF", url: "#" },

{ name: "12th Certificate", type: "PDF", url: "#" },

{ name: "Domicile Certificate", type: "PDF", url: "#" },

{ name: "Diploma Certificate", type: "PDF", url: "#" },

{ name: "Aadhar Card", type: "PDF", url: "#" },

{ name: "Family Income Certificate", type: "PDF", url: "#" },

{ name: "Caste Certificate", type: "PDF", url: "#" },

{ name: "Disability Certificate", type: "PDF", url: "#" },

]

},

{

name: "Priya Patel",

applicationNumber: "APP002",

course: "Electrical Engineering",

status: "Pending",

documents: [

{ name: "10th Certificate", type: "PDF", url: "#" },

{ name: "12th Certificate", type: "PDF", url: "#" },

{ name: "Domicile Certificate", type: "PDF", url: "#" },

{ name: "Aadhar Card", type: "PDF", url: "#" },

{ name: "Family Income Certificate", type: "PDF", url: "#" },

]

},

{

name: "Amit Kumar",

applicationNumber: "APP003",

course: "Mechanical Engineering",

status: "Disqualified",

documents: [

{ name: "10th Certificate", type: "PDF", url: "#" },

{ name: "12th Certificate", type: "PDF", url: "#" },

{ name: "Domicile Certificate", type: "PDF", url: "#" },

{ name: "Aadhar Card", type: "PDF", url: "#" },

{ name: "Caste Certificate", type: "PDF", url: "#" },

]

},

{

name: "Sneha Gupta",

applicationNumber: "APP004",

course: "Civil Engineering",

status: "Qualified",

documents: [

{ name: "10th Certificate", type: "PDF", url: "#" },

{ name: "12th Certificate", type: "PDF", url: "#" },

{ name: "Domicile Certificate", type: "PDF", url: "#" },

{ name: "Diploma Certificate", type: "PDF", url: "#" },

{ name: "Aadhar Card", type: "PDF", url: "#" },

{ name: "Family Income Certificate", type: "PDF", url: "#" },

{ name: "Caste Certificate", type: "PDF", url: "#" },

]

},

{

name: "Vikram Singh",

applicationNumber: "APP005",

course: "Chemical Engineering",

status: "Pending",

documents: [

{ name: "10th Certificate", type: "PDF", url: "#" },

{ name: "12th Certificate", type: "PDF", url: "#" },

{ name: "Domicile Certificate", type: "PDF", url: "#" },

{ name: "Aadhar Card", type: "PDF", url: "#" },

{ name: "Family Income Certificate", type: "PDF", url: "#" },

{ name: "Disability Certificate", type: "PDF", url: "#" },

]

},

]

const categories = [

{ name: "Students Applied", icon: GraduationCap, color: "text-blue-500" },

{ name: "Applications Qualified", icon: CheckCircle, color: "text-green-500" },

{ name: "Applications Disqualified", icon: XCircle, color: "text-red-500" },

{ name: "Applications Pending", icon: Clock, color: "text-yellow-500" },

]

const pmsssInfo = `

Launch in the year 2011, PMSSS is a scholarship scheme by the All India Council for Technical Education (AICTE) for the students who are domicile of the UTs of J&K and Ladakh who, after passing Class 12th or equivalent examination, secure admission in government colleges/institutions and other select institutions outside the UTs of J&K and Ladakh through AICTE's counseling process under the supernumerary quota created by the government (except in medical courses). The applicant should have passed the Class 12th examination from JKBOSE or CBSE-affiliated schools located in the UTs of J&K and Ladakh. The family income of the applicant should not exceed ₹8,00,000 per annum. 5000 fresh scholarships are available per annum. 2070 scholarships are for General Degree Courses, 2830 for Professional/Engineering courses, and 100 for Medical Courses.

`

export default function Component() {

const [activeCategory, setActiveCategory] = useState<string | null>(null)

const [showInfo, setShowInfo] = useState(false)

const [isDarkMode, setIsDarkMode] = useState(false)

const [searchTerm, setSearchTerm] = useState('')

const [students, setStudents] = useState(dummyStudents)

useEffect(() => {

const root = window.document.documentElement

root.classList.toggle('dark', isDarkMode)

}, [isDarkMode])

const filteredStudents = students.filter(student =>

(student.name.toLowerCase().includes(searchTerm.toLowerCase()) ||

student.applicationNumber.toLowerCase().includes(searchTerm.toLowerCase()) ||

student.course.toLowerCase().includes(searchTerm.toLowerCase())) &&

(activeCategory === "Students Applied" ||

(activeCategory === "Applications Qualified" && student.status === "Qualified") ||

(activeCategory === "Applications Disqualified" && student.status === "Disqualified") ||

(activeCategory === "Applications Pending" && student.status === "Pending"))

)

const handleStatusChange = (applicationNumber: string, newStatus: string) => {

setStudents(prevStudents =>

prevStudents.map(student =>

student.applicationNumber === applicationNumber

? { ...student, status: newStatus }

: student

)

)

}

return (

<div className={`min-h-screen flex flex-col ${isDarkMode ? 'bg-gray-900' : 'bg-white'} transition-colors duration-500`}>

<header className="bg-gradient-to-br from-purple-600 to-indigo-800 text-white p-4 shadow-lg">

<div className="container mx-auto flex flex-wrap justify-between items-center">

<h1 className="text-3xl font-bold flex items-center">

<BookOpen className="mr-2 h-8 w-8" />

PMSSS

</h1>

<nav className="space-x-2 flex flex-wrap">

{categories.map((category) => (

<Button

key={category.name}

variant="ghost"

className={`text-white hover:bg-white hover:bg-opacity-20 ${activeCategory === category.name ? 'bg-white bg-opacity-20' : ''}`}

onClick={() => {

setActiveCategory(category.name)

setShowInfo(false)

}}

>

<category.icon className={`mr-2 h-4 w-4 ${category.color}`} />

{category.name}

</Button>

))}

</nav>

<div className="flex items-center space-x-2">

<Button

variant="outline"

size="icon"

onClick={() => setIsDarkMode(!isDarkMode)}

className="bg-white bg-opacity-20 text-white hover:bg-opacity-30"

>

{isDarkMode ? '🌞' : '🌙'}

</Button>

<Button

variant="secondary"

className={`${isDarkMode ? 'bg-gray-700 text-white' : 'bg-white text-purple-700'} hover:bg-opacity-90`}

onClick={() => {

setShowInfo(!showInfo)

setActiveCategory(null)

}}

>

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

More Info

</Button>

</div>

</div>

</header>

<main className="flex-grow container mx-auto p-8">

<motion.h2

className={`text-5xl font-bold ${isDarkMode ? 'text-white' : 'text-gray-800'} text-center mb-12`}

initial={{ y: -50, opacity: 0 }}

animate={{ y: 0, opacity: 1 }}

transition={{ duration: 0.5 }}

>

SAG BUREAU

</motion.h2>

<AnimatePresence mode="wait">

{showInfo ? (

<motion.div

key="info"

initial={{ opacity: 0, y: 20 }}

animate={{ opacity: 1, y: 0 }}

exit={{ opacity: 0, y: -20 }}

transition={{ duration: 0.3 }}

className={`${isDarkMode ? 'bg-gray-800' : 'bg-white'} rounded-lg shadow-xl p-6 border ${isDarkMode ? 'border-gray-700' : 'border-gray-200'}`}

>

<div className="flex justify-between items-center mb-4">

<h3 className={`text-2xl font-semibold ${isDarkMode ? 'text-white' : 'text-gray-800'}`}>About PMSSS</h3>

<Button

variant="ghost"

size="icon"

onClick={() => setShowInfo(false)}

className={`${isDarkMode ? 'text-white hover:bg-gray-700' : 'text-gray-800 hover:bg-gray-200'}`}

>

<X className="h-6 w-6" />

</Button>

</div>

<ScrollArea className="h-[60vh] pr-4">

<p className={`${isDarkMode ? 'text-gray-300' : 'text-gray-600'} text-lg leading-relaxed whitespace-pre-wrap`}>{pmsssInfo}</p>

</ScrollArea>

</motion.div>

) : activeCategory ? (

<motion.div

key={activeCategory}

initial={{ opacity: 0, y: 20 }}

animate={{ opacity: 1, y: 0 }}

exit={{ opacity: 0, y: -20 }}

transition={{ duration: 0.3 }}

>

<h3 className={`text-2xl font-semibold ${isDarkMode ? 'text-white' : 'text-gray-800'} mb-4`}>{activeCategory}</h3>

<div className={`${isDarkMode ? 'bg-gray-800' : 'bg-white'} rounded-lg shadow-xl p-6 border ${isDarkMode ? 'border-gray-700' : 'border-gray-200'}`}>

<div className="mb-4 relative">

<Input

type="text"

placeholder="Search by name, application number, or course"

value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)}

className={`pl-10 ${isDarkMode ? 'bg-gray-700 text-white placeholder-gray-400' : 'bg-white text-gray-800 placeholder-gray-400'} border-gray-300`}

/>

<Search className={`absolute left-3 top-1/2 transform -translate-y-1/2 ${isDarkMode ? 'text-gray-400' : 'text-gray-500'}`} />

</div>

<StudentTable

students={filteredStudents}

showStatus={activeCategory === "Students Applied"}

isDarkMode={isDarkMode}

onStatusChange={handleStatusChange}

/>

</div>

</motion.div>

) : (

<motion.div

key="welcome"

initial={{ opacity: 0, y: 20 }}

animate={{ opacity: 1, y: 0 }}

exit={{ opacity: 0, y: -20 }}

transition={{ duration: 0.3 }}

className={`text-center ${isDarkMode ? 'text-white' : 'text-gray-800'}`}

>

<div className="flex justify-center mb-8">

<Image

src="https://hebbkx1anhila5yf.public.blob.vercel-storage.com/Prime-Ministers-Special-Scholarship-Scheme-PMSSS-0UYAF9nu6irstT2CqtIUX5Un7KQ8O0.jpg"

alt="PMSSS Scholarship for Students"

width={300}

height={300}

className="rounded-lg shadow-lg"

/>

</div>

<h3 className="text-3xl font-bold mb-4">Welcome to Prime Minister's Special Scholarship Scheme</h3>

<p className="text-xl">Click on the categories above to view student data or use the More Info button to learn about PMSSS.</p>

</motion.div>

)}

</AnimatePresence>

</main>

</div>

)

}

function StudentTable({ students, showStatus, isDarkMode, onStatusChange }) {

return (

<Table>

<TableHeader>

<TableRow className={isDarkMode ? 'bg-gray-700' : 'bg-gray-100'}>

<TableHead className={isDarkMode ? 'text-white' : 'text-gray-800'}>Name</TableHead>

<TableHead className={isDarkMode ? 'text-white' : 'text-gray-800'}>Application Number</TableHead>

<TableHead className={isDarkMode ? 'text-white' : 'text-gray-800'}>Course</TableHead>

<TableHead className={isDarkMode ? 'text-white' : 'text-gray-800'}>Status</TableHead>

<TableHead className={isDarkMode ? 'text-white' : 'text-gray-800'}>Documents</TableHead>

</TableRow>

</TableHeader>

<TableBody>

{students.map((student, index) => (

<TableRow key={student.applicationNumber} className={index % 2 === 0 ? (isDarkMode ? 'bg-gray-800' : 'bg-gray-50') : (isDarkMode ? 'bg-gray-900' : 'bg-white')}>

<TableCell className={isDarkMode ? 'text-white' : 'text-gray-800'}>{student.name}</TableCell>

<TableCell className={isDarkMode ? 'text-white' : 'text-gray-800'}>{student.applicationNumber}</TableCell>

<TableCell className={isDarkMode ? 'text-white' : 'text-gray-800'}>{student.course}</TableCell>

<TableCell className={isDarkMode ? 'text-white' : 'text-gray-800'}>

<Select

value={student.status}

onValueChange={(value) => onStatusChange(student.applicationNumber, value)}

>

<SelectTrigger className="w-[180px]">

<SelectValue placeholder="Change status" />

</SelectTrigger>

<SelectContent>

<SelectItem value="Qualified">Qualified</SelectItem>

<SelectItem value="Disqualified">Disqualified</SelectItem>

<SelectItem value="Pending">Pending</SelectItem>

</SelectContent>

</Select>

</TableCell>

<TableCell>

<DocumentsDialog student={student} isDarkMode={isDarkMode} />

</TableCell>

</TableRow>

))}

</TableBody>

</Table>

)

}

function Document({ document, isDarkMode }) {

return (

<a

href={document.url}

target="\_blank"

rel="noopener noreferrer"

className={`flex items-center p-2 rounded-md ${

isDarkMode ? 'hover:bg-gray-700' : 'hover:bg-gray-100'

} transition-colors duration-200`}

>

<FileText className={`mr-2 h-5 w-5 ${isDarkMode ? 'text-blue-400' : 'text-blue-600'}`} />

<span className={isDarkMode ? 'text-white' : 'text-gray-800'}>{document.name}</span>

<span className={`ml-auto text-sm ${isDarkMode ? 'text-gray-400' : 'text-gray-500'}`}>

{document.type}

</span>

</a>

)

}

function DocumentsDialog({ student, isDarkMode }) {

return (

<Dialog>

<DialogTrigger asChild>

<Button variant="outline" className={isDarkMode ? 'text-white border-gray-600 hover:bg-gray-700' : 'text-gray-800 border-gray-300 hover:bg-gray-100'}>

View Documents

</Button>

</DialogTrigger>

<DialogContent className={`sm:max-w-[425px] ${isDarkMode ? 'bg-gray-800 text-white' : 'bg-white text-gray-800'}`}>

<DialogHeader>

<DialogTitle className={isDarkMode ? 'text-white' : 'text-gray-800'}>Submitted Documents</DialogTitle>

</DialogHeader>

<div className="py-4">

<h4 className={`mb-4 text-sm font-medium ${isDarkMode ? 'text-gray-400' : 'text-gray-500'}`}>

{student.name} - {student.applicationNumber}

</h4>

<div className="space-y-2">

{student.documents.map((doc, index) => (

<Document key={index} document={doc} isDarkMode={isDarkMode} />

))}

</div>

</div>

</DialogContent>

</Dialog>

)

}