import React, { useState, useEffect } from 'react';

import {

Brain, Sparkles, AlertTriangle, TrendingUp,

Target, BookOpen, MessageSquare, Activity,

ChevronRight, Info, Check, X, Lightbulb,

Stethoscope, FileText, Users, BarChart,

Zap, Shield, Database, RefreshCw

} from 'lucide-react';

const AIAssistantInterface = () => {

const [activeContext, setActiveContext] = useState('valutazione');

const [showAISuggestion, setShowAISuggestion] = useState(false);

const [aiConfidence, setAiConfidence] = useState(0);

const [kbSearchResults, setKbSearchResults] = useState([]);

const [patientData, setPatientData] = useState({

nome: 'Mario Rossi',

età: 45,

diagnosi: 'Lombalgia cronica',

seduta: 5,

sintomi: ['Dolore lombare', 'Rigidità mattutina', 'Limitazione flessione']

});

// Simula suggerimenti IA basati su KB

const getAISuggestions = (context) => {

const suggestions = {

valutazione: {

tests: [

{ nome: 'Test di Lasègue', razionale: 'Escludere componente radicolare', priorità: 'Alta', confidence: 92 },

{ nome: 'Test di Thomas', razionale: 'Valutare flessori anca', priorità: 'Media', confidence: 85 },

{ nome: 'SLUMP Test', razionale: 'Tensione neurale', priorità: 'Media', confidence: 78 }

],

redFlags: [

{ flag: 'Dolore notturno costante', presente: false, azione: 'Monitorare' },

{ flag: 'Deficit neurologici', presente: false, azione: 'OK' }

],

scale: ['Oswestry Disability Index', 'Roland-Morris', 'VAS']

},

trattamento: {

tecniche: [

{ tecnica: 'Mobilizzazione L4-L5', evidenza: 'A', successo: '78%', basatoSu: '145 casi simili' },

{ tecnica: 'Esercizi core stability', evidenza: 'A', successo: '82%', basatoSu: '203 casi simili' },

{ tecnica: 'Educazione posturale', evidenza: 'B', successo: '71%', basatoSu: '89 casi simili' }

],

progressione: {

fase: 'Subacuta',

prossimo: 'Incremento carico 20%',

criteri: 'VAS < 4, ROM > 70%'

}

},

prognosi: {

sedutePreviste: '8-10',

probabilitàSuccesso: '85%',

tempoRecupero: '4-6 settimane',

fattoriPositivi: ['Età < 50', 'Motivazione alta', 'No comorbidità'],

fattoriRischio: ['Cronicità > 3 mesi', 'Lavoro sedentario']

}

};

return suggestions[context] || {};

};

// Componente Valutazione IA

const ValutazioneAI = () => {

const suggestions = getAISuggestions('valutazione');

return (

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

<div className="bg-blue-50 rounded-lg p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<Stethoscope className="w-5 h-5 text-blue-600" />

Test Consigliati da IA

</h3>

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

{suggestions.tests?.map((test, i) => (

<div key={i} className="bg-white rounded p-3 flex justify-between items-start">

<div className="flex-1">

<p className="font-medium">{test.nome}</p>

<p className="text-sm text-gray-600">{test.razionale}</p>

</div>

<div className="text-right">

<span className={`text-xs px-2 py-1 rounded ${

test.priorità === 'Alta' ? 'bg-red-100 text-red-700' :

test.priorità === 'Media' ? 'bg-yellow-100 text-yellow-700' :

'bg-green-100 text-green-700'

}`}>

{test.priorità}

</span>

<p className="text-xs text-gray-500 mt-1">

Confidence: {test.confidence}%

</p>

</div>

</div>

))}

</div>

</div>

<div className="bg-red-50 rounded-lg p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<AlertTriangle className="w-5 h-5 text-red-600" />

Red Flags Check

</h3>

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

{suggestions.redFlags?.map((flag, i) => (

<div key={i} className="flex items-center justify-between">

<span className="text-sm">{flag.flag}</span>

<div className="flex items-center gap-2">

{flag.presente ?

<X className="w-4 h-4 text-red-500" /> :

<Check className="w-4 h-4 text-green-500" />

}

<span className="text-xs text-gray-500">{flag.azione}</span>

</div>

</div>

))}

</div>

</div>

<div className="bg-purple-50 rounded-lg p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<FileText className="w-5 h-5 text-purple-600" />

Scale Valutative Suggerite

</h3>

<div className="flex flex-wrap gap-2">

{suggestions.scale?.map((scala, i) => (

<span key={i} className="px-3 py-1 bg-white rounded-full text-sm">

{scala}

</span>

))}

</div>

</div>

</div>

);

};

// Componente Trattamento IA

const TrattamentoAI = () => {

const suggestions = getAISuggestions('trattamento');

return (

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

<div className="bg-green-50 rounded-lg p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<Target className="w-5 h-5 text-green-600" />

Tecniche Consigliate (basate su KB)

</h3>

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

{suggestions.tecniche?.map((tec, i) => (

<div key={i} className="bg-white rounded p-3">

<div className="flex justify-between items-start mb-2">

<p className="font-medium">{tec.tecnica}</p>

<span className="text-xs px-2 py-1 bg-green-100 text-green-700 rounded">

Evidenza: {tec.evidenza}

</span>

</div>

<div className="flex justify-between text-sm text-gray-600">

<span>Tasso successo: {tec.successo}</span>

<span className="text-xs">Basato su {tec.basatoSu}</span>

</div>

<div className="mt-2 w-full bg-gray-200 rounded-full h-2">

<div

className="bg-green-500 h-2 rounded-full"

style={{ width: tec.successo }}

/>

</div>

</div>

))}

</div>

</div>

<div className="bg-blue-50 rounded-lg p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<TrendingUp className="w-5 h-5 text-blue-600" />

Progressione Suggerita

</h3>

<div className="bg-white rounded p-3">

<p className="text-sm text-gray-600">Fase attuale:</p>

<p className="font-medium">{suggestions.progressione?.fase}</p>

<div className="mt-3 pt-3 border-t">

<p className="text-sm text-gray-600">Prossimo step:</p>

<p className="font-medium text-blue-600">{suggestions.progressione?.prossimo}</p>

<p className="text-xs text-gray-500 mt-1">

Criteri: {suggestions.progressione?.criteri}

</p>

</div>

</div>

</div>

</div>

);

};

// Componente Prognosi IA

const PrognosiAI = () => {

const suggestions = getAISuggestions('prognosi');

return (

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

<div className="bg-indigo-50 rounded-lg p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<Activity className="w-5 h-5 text-indigo-600" />

Predizione Outcome

</h3>

<div className="grid grid-cols-2 gap-4">

<div className="bg-white rounded p-3">

<p className="text-sm text-gray-600">Sedute previste</p>

<p className="text-2xl font-bold text-indigo-600">{suggestions.sedutePreviste}</p>

</div>

<div className="bg-white rounded p-3">

<p className="text-sm text-gray-600">Probabilità successo</p>

<p className="text-2xl font-bold text-green-600">{suggestions.probabilitàSuccesso}</p>

</div>

</div>

<div className="bg-white rounded p-3 mt-3">

<p className="text-sm text-gray-600">Tempo recupero stimato</p>

<p className="font-medium">{suggestions.tempoRecupero}</p>

</div>

</div>

<div className="grid grid-cols-2 gap-4">

<div className="bg-green-50 rounded-lg p-4">

<h4 className="font-medium mb-2 text-green-700">Fattori Positivi</h4>

<ul className="space-y-1">

{suggestions.fattoriPositivi?.map((fattore, i) => (

<li key={i} className="text-sm flex items-center gap-2">

<Check className="w-3 h-3 text-green-500" />

{fattore}

</li>

))}

</ul>

</div>

<div className="bg-yellow-50 rounded-lg p-4">

<h4 className="font-medium mb-2 text-yellow-700">Fattori di Rischio</h4>

<ul className="space-y-1">

{suggestions.fattoriRischio?.map((fattore, i) => (

<li key={i} className="text-sm flex items-center gap-2">

<AlertTriangle className="w-3 h-3 text-yellow-500" />

{fattore}

</li>

))}

</ul>

</div>

</div>

</div>

);

};

// Knowledge Base Search

const KnowledgeBaseSearch = () => {

const [searchQuery, setSearchQuery] = useState('');

const [searchResults, setSearchResults] = useState([

{

titolo: 'Protocollo Lombalgia Cronica - Linee Guida 2024',

tipo: 'protocollo',

rilevanza: 95,

contenuto: 'Fase subacuta: mobilizzazione graduale, esercizi core...'

},

{

titolo: 'Caso Clinico: M, 45 anni, lombalgia da 6 mesi',

tipo: 'caso\_studio',

rilevanza: 88,

contenuto: 'Ottimi risultati con approccio multimodale in 8 sedute...'

},

{

titolo: 'Progressione Esercizi Core Stability',

tipo: 'esercizi',

rilevanza: 82,

contenuto: 'Livello 1: Dead bug, Bird dog. Livello 2: Plank progressioni...'

}

]);

return (

<div className="bg-gray-50 rounded-lg p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<Database className="w-5 h-5 text-gray-600" />

Knowledge Base Ricerca

</h3>

<div className="flex gap-2 mb-3">

<input

type="text"

className="flex-1 p-2 border rounded"

placeholder="Cerca nella KB..."

value={searchQuery}

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

/>

<button className="px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700">

Cerca

</button>

</div>

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

{searchResults.map((result, i) => (

<div key={i} className="bg-white rounded p-3 hover:shadow-md transition-shadow cursor-pointer">

<div className="flex justify-between items-start mb-1">

<p className="font-medium text-sm">{result.titolo}</p>

<span className="text-xs px-2 py-1 bg-blue-100 text-blue-700 rounded">

{result.rilevanza}% match

</span>

</div>

<p className="text-xs text-gray-600">{result.contenuto}</p>

<span className={`text-xs px-2 py-1 rounded mt-2 inline-block ${

result.tipo === 'protocollo' ? 'bg-purple-100 text-purple-700' :

result.tipo === 'caso\_studio' ? 'bg-green-100 text-green-700' :

'bg-yellow-100 text-yellow-700'

}`}>

{result.tipo.replace('\_', ' ')}

</span>

</div>

))}

</div>

</div>

);

};

// Chat Assistant

const ChatAssistant = () => {

const [messages, setMessages] = useState([

{ role: 'ai', text: 'Ciao! Sono il tuo assistente IA. Come posso aiutarti con questo paziente?' },

{ role: 'user', text: 'Il paziente riferisce aumento del dolore dopo gli esercizi' },

{ role: 'ai', text: 'Basandomi sulla KB, potrebbe essere DOMs (indolenzimento muscolare tardivo). Suggerisco: 1) Ridurre intensità del 30%, 2) Aumentare fase riscaldamento, 3) Verificare tecnica esecuzione. Nel 78% dei casi simili, questa strategia ha risolto il problema.' }

]);

const [inputText, setInputText] = useState('');

const sendMessage = () => {

if (inputText.trim()) {

setMessages([...messages,

{ role: 'user', text: inputText },

{ role: 'ai', text: 'Analizzo la tua domanda consultando la KB...' }

]);

setInputText('');

// Simula risposta IA

setTimeout(() => {

setMessages(prev => [

...prev.slice(0, -1),

{ role: 'ai', text: 'Basandomi su 45 casi simili nella KB, suggerisco di verificare la presenza di trigger points miofasciali. Nel 73% dei casi con sintomi simili, il trattamento dei trigger points ha portato miglioramento significativo.' }

]);

}, 1500);

}

};

return (

<div className="bg-white rounded-lg shadow-sm p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<MessageSquare className="w-5 h-5 text-blue-600" />

Assistente IA Chat

</h3>

<div className="h-64 overflow-y-auto mb-3 space-y-2 p-3 bg-gray-50 rounded">

{messages.map((msg, i) => (

<div key={i} className={`flex ${msg.role === 'user' ? 'justify-end' : 'justify-start'}`}>

<div className={`max-w-[80%] p-2 rounded-lg ${

msg.role === 'user'

? 'bg-blue-600 text-white'

: 'bg-white border'

}`}>

{msg.role === 'ai' && <Brain className="w-4 h-4 mb-1 text-blue-600" />}

<p className="text-sm">{msg.text}</p>

</div>

</div>

))}

</div>

<div className="flex gap-2">

<input

type="text"

className="flex-1 p-2 border rounded"

placeholder="Chiedi all'assistente IA..."

value={inputText}

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

onKeyPress={(e) => e.key === 'Enter' && sendMessage()}

/>

<button

onClick={sendMessage}

className="px-4 py-2 bg-blue-600 text-white rounded hover:bg-blue-700"

>

Invia

</button>

</div>

</div>

);

};

return (

<div className="min-h-screen bg-gray-100">

{/\* Header \*/}

<div className="bg-white shadow-sm">

<div className="max-w-7xl mx-auto px-4 py-4">

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

<h1 className="text-2xl font-bold flex items-center gap-2">

<Brain className="w-8 h-8 text-blue-600" />

Sistema IA Assistito - Medicina Ravenna

</h1>

<div className="flex items-center gap-4">

<div className="text-sm">

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

<span className="font-medium ml-1">2.4.1</span>

</div>

<div className="text-sm">

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

<span className="font-medium ml-1 text-green-600">Attivi</span>

</div>

</div>

</div>

</div>

</div>

{/\* Patient Info Bar \*/}

<div className="bg-blue-50 border-b">

<div className="max-w-7xl mx-auto px-4 py-3">

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

<div className="flex gap-6">

<span className="text-sm"><strong>Paziente:</strong> {patientData.nome}</span>

<span className="text-sm"><strong>Età:</strong> {patientData.età}</span>

<span className="text-sm"><strong>Diagnosi:</strong> {patientData.diagnosi}</span>

<span className="text-sm"><strong>Seduta:</strong> #{patientData.seduta}</span>

</div>

<button className="text-sm text-blue-600 hover:underline flex items-center gap-1">

<RefreshCw className="w-3 h-3" />

Aggiorna contesto

</button>

</div>

</div>

</div>

{/\* Main Content \*/}

<div className="max-w-7xl mx-auto px-4 py-6">

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

{/\* Left Column - AI Suggestions \*/}

<div className="col-span-2">

<div className="bg-white rounded-lg shadow p-4 mb-4">

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

<h2 className="text-lg font-semibold flex items-center gap-2">

<Sparkles className="w-5 h-5 text-yellow-500" />

Suggerimenti IA Contestuali

</h2>

<div className="flex gap-2">

<button

onClick={() => setActiveContext('valutazione')}

className={`px-3 py-1 rounded text-sm ${

activeContext === 'valutazione'

? 'bg-blue-600 text-white'

: 'bg-gray-200'

}`}

>

Valutazione

</button>

<button

onClick={() => setActiveContext('trattamento')}

className={`px-3 py-1 rounded text-sm ${

activeContext === 'trattamento'

? 'bg-blue-600 text-white'

: 'bg-gray-200'

}`}

>

Trattamento

</button>

<button

onClick={() => setActiveContext('prognosi')}

className={`px-3 py-1 rounded text-sm ${

activeContext === 'prognosi'

? 'bg-blue-600 text-white'

: 'bg-gray-200'

}`}

>

Prognosi

</button>

</div>

</div>

{activeContext === 'valutazione' && <ValutazioneAI />}

{activeContext === 'trattamento' && <TrattamentoAI />}

{activeContext === 'prognosi' && <PrognosiAI />}

</div>

{/\* Knowledge Base Search \*/}

<KnowledgeBaseSearch />

</div>

{/\* Right Column - Chat & Stats \*/}

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

<ChatAssistant />

{/\* AI Performance Stats \*/}

<div className="bg-white rounded-lg shadow p-4">

<h3 className="font-semibold mb-3 flex items-center gap-2">

<BarChart className="w-5 h-5 text-green-600" />

Performance IA

</h3>

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

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

<span className="text-sm">Accuratezza predizioni</span>

<span className="font-bold text-green-600">87%</span>

</div>

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

<span className="text-sm">Suggerimenti adottati</span>

<span className="font-bold text-blue-600">72%</span>

</div>

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

<span className="text-sm">Tempo risparmiato</span>

<span className="font-bold text-purple-600">-45%</span>

</div>

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

<span className="text-sm">Casi nella KB</span>

<span className="font-bold">12,456</span>

</div>

</div>

</div>

{/\* Learning Status \*/}

<div className="bg-gradient-to-r from-blue-50 to-purple-50 rounded-lg p-4">

<h3 className="font-semibold mb-2 flex items-center gap-2">

<Zap className="w-5 h-5 text-purple-600" />

Apprendimento Continuo

</h3>

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

Il sistema sta apprendendo da questo caso

</p>

<div className="space-y-1">

<div className="flex items-center gap-2">

<div className="w-2 h-2 bg-green-500 rounded-full animate-pulse" />

<span className="text-xs">Pattern riconosciuti: 3</span>

</div>

<div className="flex items-center gap-2">

<div className="w-2 h-2 bg-blue-500 rounded-full animate-pulse" />

<span className="text-xs">Correlazioni trovate: 7</span>

</div>

<div className="flex items-center gap-2">

<div className="w-2 h-2 bg-purple-500 rounded-full animate-pulse" />

<span className="text-xs">KB aggiornamenti: 2 pending</span>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

);

};

export default AIAssistantInterface;