src/components/ChatInterface.tsx

import React, { useState, useRef, useEffect } from 'react';Add commentMore actions

import { Send, Loader2, MessageCircle } from 'lucide-react';

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

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

import { Input } from '@/components/ui/input';

import { Message, UploadedFile } from '../pages/Index';

interface ChatInterfaceProps {

messages: Message[];

onSendMessage: (text: string) => void;

isProcessing: boolean;

uploadedFile: UploadedFile | null;

isDarkMode: boolean;

}

const ChatInterface: React.FC<ChatInterfaceProps> = ({

messages,

onSendMessage,

isProcessing,

uploadedFile,

isDarkMode

}) => {

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

const messagesEndRef = useRef<HTMLDivElement>(null);

const scrollToBottom = () => {

messagesEndRef.current?.scrollIntoView({ behavior: 'smooth' });

};

useEffect(() => {

scrollToBottom();

}, [messages]);

const handleSubmit = (e: React.FormEvent) => {

e.preventDefault();

if (inputText.trim() && !isProcessing && uploadedFile) {

onSendMessage(inputText.trim());

setInputText('');

}

};

const formatTime = (date: Date) => {

return date.toLocaleTimeString([], { hour: '2-digit', minute: '2-digit' });

};

return (

<Card className={`flex flex-col h-full transition-colors duration-300 ${

isDarkMode ? 'bg-gray-800 border-gray-700' : 'bg-white border-gray-200'

}`}>

{/\* Chat Header \*/}

<div className={`p-4 border-b ${isDarkMode ? 'border-gray-700' : 'border-gray-200'}`}>

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

<div className={`p-2 rounded-full ${isDarkMode ? 'bg-blue-600' : 'bg-gradient-to-r from-blue-600 to-indigo-600'}`}>

<MessageCircle className="w-5 h-5 text-white" />

</div>

<div>

<h3 className={`font-semibold ${isDarkMode ? 'text-white' : 'text-gray-900'}`}>

PDF Chat Assistant

</h3>

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

{uploadedFile ? `Chatting about ${uploadedFile.name}` : 'Upload a PDF to start chatting'}

</p>

</div>

</div>

</div>

{/\* Messages Area \*/}

<div className="flex-1 overflow-y-auto p-4 space-y-4">

{!uploadedFile ? (

<div className="flex items-center justify-center h-full">

<div className="text-center">

<MessageCircle className={`w-16 h-16 mx-auto mb-4 ${isDarkMode ? 'text-gray-600' : 'text-gray-400'}`} />

<p className={`text-lg font-medium ${isDarkMode ? 'text-gray-300' : 'text-gray-600'}`}>

Upload a PDF to start chatting

</p>

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

Once you upload a document, I'll be ready to answer your questions about it.

</p>

</div>

</div>

) : messages.length === 0 ? (

<div className="flex items-center justify-center h-full">

<div className="text-center">

<MessageCircle className={`w-16 h-16 mx-auto mb-4 ${isDarkMode ? 'text-gray-600' : 'text-gray-400'}`} />

<p className={`text-lg font-medium ${isDarkMode ? 'text-gray-300' : 'text-gray-600'}`}>

Ready to chat!

</p>

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

Ask me anything about your uploaded PDF.

</p>

</div>

</div>

) : (

<>

{messages.map((message) => (

<div

key={message.id}

className={`flex ${message.isUser ? 'justify-end' : 'justify-start'} animate-fade-in`}

>

<div

className={`

max-w-[80%] p-3 rounded-lg shadow-sm

${message.isUser

? (isDarkMode

? 'bg-blue-600 text-white'

: 'bg-gradient-to-r from-blue-600 to-indigo-600 text-white')

: (isDarkMode

? 'bg-gray-700 text-gray-100'

: 'bg-gray-100 text-gray-900')

}

`}

>

<p className="text-sm leading-relaxed">{message.text}</p>

<p className={`text-xs mt-1 opacity-70`}>

{formatTime(message.timestamp)}

</p>

</div>

</div>

))}

{isProcessing && (

<div className="flex justify-start animate-fade-in">

<div className={`p-3 rounded-lg shadow-sm ${

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

}`}>

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

<Loader2 className={`w-4 h-4 animate-spin ${isDarkMode ? 'text-gray-400' : 'text-gray-500'}`} />

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

Assistant is typing...

</span>

</div>

</div>

</div>

)}

</>

)}

<div ref={messagesEndRef} />

</div>

{/\* Input Area \*/}

<div className={`p-4 border-t ${isDarkMode ? 'border-gray-700' : 'border-gray-200'}`}>

<form onSubmit={handleSubmit} className="flex space-x-2">

<Input

value={inputText}

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

placeholder={uploadedFile ? "Ask a question about your PDF..." : "Upload a PDF first..."}

disabled={!uploadedFile || isProcessing}

className={`flex-1 ${isDarkMode ? 'bg-gray-700 border-gray-600 text-white placeholder-gray-400' : ''}`}

/>

<Button

type="submit"

disabled={!inputText.trim() || !uploadedFile || isProcessing}

size="sm"

className={isDarkMode ? 'bg-blue-600 hover:bg-blue-700' : 'bg-gradient-to-r from-blue-600 to-indigo-600 hover:from-blue-700 hover:to-indigo-700'}

>

<Send className="w-4 h-4" />

</Button>

</form>

</div>

</Card>

);

};

export default ChatInterface;