src/components/FileUpload.tsx

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

import { Upload, File, Check, Loader2 } from 'lucide-react';

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

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

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

interface FileUploadProps {

onFileUpload: (file: File) => void;

uploadedFile: UploadedFile | null;

isProcessing: boolean;

isDarkMode: boolean;

}

const FileUpload: React.FC<FileUploadProps> = ({

onFileUpload,

uploadedFile,

isProcessing,

isDarkMode

}) => {

const [isDragOver, setIsDragOver] = useState(false);

const handleDragOver = useCallback((e: React.DragEvent) => {

e.preventDefault();

setIsDragOver(true);

}, []);

const handleDragLeave = useCallback((e: React.DragEvent) => {

e.preventDefault();

setIsDragOver(false);

}, []);

const handleDrop = useCallback((e: React.DragEvent) => {

e.preventDefault();

setIsDragOver(false);

const files = Array.from(e.dataTransfer.files);

const pdfFile = files.find(file => file.type === 'application/pdf');

if (pdfFile) {

onFileUpload(pdfFile);

}

}, [onFileUpload]);

const handleFileSelect = useCallback((e: React.ChangeEvent<HTMLInputElement>) => {

const file = e.target.files?.[0];

if (file && file.type === 'application/pdf') {

onFileUpload(file);

}

}, [onFileUpload]);

const formatFileSize = (bytes: number) => {

if (bytes === 0) return '0 Bytes';

const k = 1024;

const sizes = ['Bytes', 'KB', 'MB', 'GB'];

const i = Math.floor(Math.log(bytes) / Math.log(k));

return parseFloat((bytes / Math.pow(k, i)).toFixed(2)) + ' ' + sizes[i];

};

return (

<div className="space-y-6">

<Card className={`p-6 transition-colors duration-300 ${

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

}`}>

<h2 className={`text-xl font-semibold mb-4 ${isDarkMode ? 'text-white' : 'text-gray-900'}`}>

Upload PDF Document

</h2>

<div

className={`

border-2 border-dashed rounded-lg p-8 text-center transition-all duration-200

${isDragOver

? (isDarkMode ? 'border-blue-400 bg-blue-900/20' : 'border-blue-400 bg-blue-50')

: (isDarkMode ? 'border-gray-600 hover:border-gray-500' : 'border-gray-300 hover:border-gray-400')

}

${isProcessing ? 'pointer-events-none opacity-50' : 'cursor-pointer'}

`}

onDragOver={handleDragOver}

onDragLeave={handleDragLeave}

onDrop={handleDrop}

onClick={() => document.getElementById('file-input')?.click()}

>

<input

id="file-input"

type="file"

accept=".pdf"

onChange={handleFileSelect}

className="hidden"

disabled={isProcessing}

/>

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

{isProcessing ? (

<Loader2 className={`w-12 h-12 mx-auto animate-spin ${isDarkMode ? 'text-blue-400' : 'text-blue-600'}`} />

) : (

<Upload className={`w-12 h-12 mx-auto ${isDarkMode ? 'text-gray-400' : 'text-gray-500'}`} />

)}

<div>

<p className={`text-lg font-medium ${isDarkMode ? 'text-white' : 'text-gray-900'}`}>

{isProcessing ? 'Processing PDF...' : 'Drop your PDF here or click to browse'}

</p>

<p className={`text-sm mt-1 ${isDarkMode ? 'text-gray-400' : 'text-gray-500'}`}>

{isProcessing ? 'Please wait while we extract content from your document' : 'Only PDF files are supported'}

</p>

</div>

{!isProcessing && (

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

Choose File

</Button>

)}

</div>

</div>

</Card>

{uploadedFile && (

<Card className={`p-6 animate-fade-in transition-colors duration-300 ${

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

}`}>

<div className="flex items-start space-x-4">

<div className={`p-2 rounded-lg ${isDarkMode ? 'bg-green-900' : 'bg-green-100'}`}>

<Check className={`w-5 h-5 ${isDarkMode ? 'text-green-400' : 'text-green-600'}`} />

</div>

<div className="flex-1">

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

Document Uploaded Successfully

</h3>

<div className={`mt-2 p-3 rounded-lg ${isDarkMode ? 'bg-gray-700' : 'bg-gray-50'}`}>

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

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

<span className={`font-medium ${isDarkMode ? 'text-gray-300' : 'text-gray-700'}`}>

{uploadedFile.name}

</span>

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

({formatFileSize(uploadedFile.size)})

</span>

</div>

<p className={`text-sm ${isDarkMode ? 'text-gray-400' : 'text-gray-600'}`}>

{uploadedFile.preview}

</p>

</div>

</div>

</div>

</Card>

)}

</div>

);

};

export default FileUpload;