

```
// src/components/Forms/AddCardForm.jsx
import { useState } from 'react';
import { useNavigate } from 'react-router-dom';
import { Upload, Music as MusicIcon, Link as LinkIcon, X } from 'lucide-react';
import { useAuth } from '../../hooks/useAuth';
import { useCards } from '../../hooks/useCards';
import { uploadImage, uploadAudio } from '../../lib/supabase';

export default function AddCardForm() {
  const { user } = useAuth();
  const { addCard, generateCardId } = useCards();
  const navigate = useNavigate();

  const [title, setTitle] = useState('');
  const [description, setDescription] = useState('');
  const [category, setCategory] = useState('');
  const [image, setImage] = useState(null);
  const [imagePreview, setImagePreview] = useState(null);
  const [musicType, setMusicType] = useState('none'); // 'none', 'url', 'file'
  const [musicUrl, setMusicUrl] = useState('');
  const [musicFile, setMusicFile] = useState(null);
  const [loading, setLoading] = useState(false);
  const [error, setError] = useState('');

  const handleImageChange = (e) => {
    const file = e.target.files[0];
    if (file) {
      setImage(file);
      setImagePreview(URL.createObjectURL(file));
    }
  };

  const handleMusicFileChange = (e) => {
    const file = e.target.files[0];
    if (file) {
      setMusicFile(file);
    }
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    setError('');
    setLoading(true);

    try {
      if (!image) {
        throw new Error('Please upload an image');
      }

      // Upload image
      const imageUrl = await uploadImage(image);
    
```

```
// Upload music file if selected
let musicFileUrl = null;
if (musicType === 'file' && musicFile) {
  musicFileUrl = await uploadAudio(musicFile);
}

// Create card
const cardData = {
  user_id: user.id,
  title,
  description,
  category: category || null,
  image_url: imageUrl,
  music_url: musicType === 'url' ? musicUrl : null,
  music_file_url: musicFileUrl,
  card_id: generateCardId(),
};

await addCard(cardData);
navigate('/');
} catch (err) {
  setError(err.message);
} finally {
  setLoading(false);
}
};

return (
<div className="min-h-screen bg-gradient-to-br from-pink-50 to-purple-50 py-12 px-4">
  <div className="max-w-2xl mx-auto">
    <div className="bg-white rounded-2xl shadow-lg p-8">
      <h1 className="text-3xl font-bold text-gray-900 mb-2">Add New Card</h1>
      <p className="text-gray-600 mb-8">Share a cultural moment with your friends</p>

      {error && (
        <div className="mb-6 p-4 bg-red-50 border border-red-200 text-red-700 rounded-lg">
          {error}
        </div>
      )}
    </div>
    <form onSubmit={handleSubmit} className="space-y-6">
      {/* Image Upload */}
      <div>
        <label className="block text-sm font-medium text-gray-700 mb-2">
          Illustration *
        </label>
        <div className="border-2 border-dashed border-gray-300 rounded-lg p-6 text-center hover:border-pink-500 transition">
          {imagePreview ? (
            <div className="relative">
              <img
                src={imagePreview}
                alt="Preview"
                className="max-h-64 mx-auto rounded-lg"
              />
            </div>
          ) : (
            <div>
              <img alt="Upload icon" />
            </div>
          )}
        </div>
      </div>
    </form>
  </div>
</div>

```

```
<button
  type="button"
  onClick={() => {
    setImage(null);
    setImagePreview(null);
  }}
  className="absolute top-2 right-2 p-2 bg-red-500 text-white rounded-full hover:bg-red-600">
  <X size={16} />
</button>
</div>
) : (
<div>
  <Upload size={48} className="mx-auto text-gray-400 mb-2" />
  <label className="cursor-pointer text-pink-600 hover:text-pink-700 font-medium">
    Choose an image
  <input
    type="file"
    accept="image/*"
    onChange={handleImageChange}
    className="hidden"
  />
  </label>
</div>
)
</div>
</div>

{/* Title */}
<div>
  <label className="block text-sm font-medium text-gray-700 mb-2">
    Title *
  </label>
  <input
    type="text"
    value={title}
    onChange={(e) => setTitle(e.target.value)}
    className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-pink-500 focus:border-transparent"
    placeholder="e.g., Miyazaki's Wind Philosophy"
    required
  />
</div>

{/* Description */}
<div>
  <label className="block text-sm font-medium text-gray-700 mb-2">
    Description *
  </label>
  <textarea
    value={description}
    onChange={(e) => setDescription(e.target.value)}
    rows={5}
    className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-pink-500 focus:border-transparent"
  />
</div>
```

```
pink-500 focus:border-transparent"
    placeholder="Share your thoughts and insights..."
    required
/>
</div>

/* Category */
<div>
    <label className="block text-sm font-medium text-gray-700 mb-2">
        Category (optional)
    </label>
    <input
        type="text"
        value={category}
        onChange={(e) => setCategory(e.target.value)}
        className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-pink-500 focus:border-transparent"
        placeholder="e.g., Film, Literature, Art, Music"
    />
</div>

/* Music Section */
<div>
    <label className="block text-sm font-medium text-gray-700 mb-3">
        Add Music (optional)
    </label>

    <div className="flex gap-3 mb-4">
        <button
            type="button"
            onClick={() => setMusicType('none')}
            className={`${flex-1 py-2 px-4 rounded-lg border-2 transition ${
                musicType === 'none'
                    ? 'border-pink-500 bg-pink-50 text-pink-700'
                    : 'border-gray-300 hover:border-gray-400'
            }`}
        >
            No Music
        </button>
        <button
            type="button"
            onClick={() => setMusicType('url')}
            className={`${flex-1 py-2 px-4 rounded-lg border-2 transition ${
                musicType === 'url'
                    ? 'border-pink-500 bg-pink-50 text-pink-700'
                    : 'border-gray-300 hover:border-gray-400'
            }`}
        >
            <LinkIcon size={16} className="inline mr-1" />
            URL
        </button>
        <button
            type="button"
            onClick={() => setMusicType('file')}
            className={`${flex-1 py-2 px-4 rounded-lg border-2 transition ${

```

```

        musicType === 'file'
          ? 'border-pink-500 bg-pink-50 text-pink-700'
          : 'border-gray-300 hover:border-gray-400'
        }
      >
      <MusicIcon size={16} className="inline mr-1" />
      Upload
    </button>
</div>

{musicType === 'url' && (
  <input
    type="url"
    value={musicUrl}
    onChange={(e) => setMusicUrl(e.target.value)}
    className="w-full px-4 py-3 border border-gray-300 rounded-lg focus:ring-2 focus:ring-pink-500 focus:border-transparent"
    placeholder="Spotify, YouTube, or SoundCloud URL"
  />
)}

{musicType === 'file' && (
  <div className="border-2 border-dashed border-gray-300 rounded-lg p-4 text-center hover:border-pink-500 transition">
    {musicFile ? (
      <div className="flex items-center justify-between">
        <span className="text-sm text-gray-700">{musicFile.name}</span>
        <button
          type="button"
          onClick={() => setMusicFile(null)}
          className="text-red-500 hover:text-red-700"
        >
          <X size={20} />
        </button>
      </div>
    ) : (
      <label className="cursor-pointer">
        <MusicIcon size={32} className="mx-auto text-gray-400 mb-2" />
        <span className="text-pink-600 hover:text-pink-700 font-medium">
          Choose MP3 file
        </span>
        <input
          type="file"
          accept="audio/mpeg, audio/mp3"
          onChange={handleMusicFileChange}
          className="hidden"
        />
      </label>
    )
  </div>
)
}

{/*
  Submit Button */
<button

```

```
        type="submit"
        disabled={loading}
        className="w-full py-4 bg-pink-500 text-white rounded-lg hover:bg-pink-600 transition
disabled:opacity-50 disabled:cursor-not-allowed font-medium text-lg"
      >
    {loading ? 'Creating Card...' : 'Create Card'}
  </button>
</form>
</div>
</div>
</div>
);
}
```