

```
import React, { useState, useEffect, useRef } from 'react';
import { Sparkles, Wand2, PlusCircle, Trash2, UserPlus, Globe, Check } from 'lucide-react';

// --- MOCK DATA & STYLING (for standalone demonstration) ---
const mockUser = {
  id: 101,
  name: 'Alexina B.',
  mission: {
    name: "Urban Greening Initiative",
    description: "To design and implement sustainable green spaces within urban environments, fostering community engagement and increasing local biodiversity.",
  },
};

const colors = { background: 'bg-black', card: 'bg-[#1a1a1a]', text: 'text-white', textMuted: 'text-gray-400', primary: 'text-lime-400', primaryBg: 'bg-lime-400', primaryBgHover: 'hover:bg-lime-500', border: 'border-gray-700' };

// --- HELPER COMPONENTS (required for the form) ---
const Card = ({ children, className = "" }) => ( <div className={`${colors.card} rounded-lg ${className}`}> {children} </div> );
const Input = ({ label, placeholder, type = 'text', value, onChange, required = false, name }) => ( <div> <label className="block text-sm font-medium text-gray-300 mb-1">{label}</label> <input type={type} placeholder={placeholder} value={value} onChange={onChange} required={required} name={name} className="w-full bg-gray-900 border border-gray-700 rounded-lg py-2 px-3 focus:outline-none focus:ring-2 focus:ring-lime-400" /> </div> );
const Textarea = ({ label, placeholder, value, onChange, rows = 3, required = false, name }) => ( <div> <label className="block text-sm font-medium text-gray-300 mb-1">{label}</label> <textarea placeholder={placeholder} value={value} onChange={onChange} required={required} rows={rows} name={name} className="w-full bg-gray-900" /> </div> );
```

```

border border-gray-700 rounded-lg py-2 px-3 focus:outline-none
focus:ring-2 focus:ring-lime-400" /> </div> );
const Button = ({ children, onClick, variant = 'primary', className = '',
type = 'button', disabled = false }) => {
  const baseStyle = 'px-4 py-2 rounded-md font-semibold transition-all
duration-300 flex items-center justify-center gap-2 disabled:opacity-50
disabled:cursor-not-allowed';
  const styles = { primary: `${colors.primaryBg} text-black
${colors.primaryBgHover}`, secondary: `bg-transparent border-2
${colors.border} ${colors.text} hover:bg-gray-800` };
  return ( <button type={type} onClick={onClick} className={`${baseStyle}
${styles[variant]} ${className}`} disabled={disabled}> {children}
</button> );
};

// --- CORE COMPONENT: Global Good Project Creation Form (Form I) ---
const CreateProject = ({ setPage }) => {
  const [step, setStep] = useState('form_i'); // intention step removed
  const [formData, setFormData] = useState({ title: '', completionDate:
'', signupDeadline: '', location: '', contribution: '', bring: '',
provide: '' });
  const [activities, setActivities] = useState([]);
  const [isAddingActivity, setIsAddingActivity] = useState(false);
  const initialNewActivityState = { id: null, title: '', description: '',
date: '', time: '', type: 'In-Person', proof: '' };
  const [newActivity, setNewActivity] =
useState(initialNewActivityState);

  const handleFormChange = (e) => {
    const { name, value } = e.target;
    setFormData(prev => ({ ...prev, [name]: value }));
  };

  const handleActivityChange = (e) => {
    const { name, value } = e.target;
    setNewActivity(prev => ({ ...prev, [name]: value }));
  };

```

```

};

const handleSaveActivity = () => {
    if (!newActivity.title || !newActivity.date || !newActivity.time) {
        alert("Please fill in all activity details.");
        return;
    }
    setActivities(prev => [...prev, { ...newActivity, id: Date.now() }]);
    setNewActivity(initialNewActivityState);
    setIsAddingActivity(false);
};

const handleRemoveActivity = (id) => {
    setActivities(prev => prev.filter(act => act.id !== id));
};

if (step === 'review') {
    return (
        <div className="max-w-3xl mx-auto py-8 px-4">
            <h2 className="text-4xl font-bold text-center mb-4">Review & Publish</h2> <p className="text-center text-gray-400 mb-12">This is a preview of your project details and invite post. Review everything carefully before publishing.</p>
            <Card className="p-6">
                <h3 className="text-xl font-bold text-lime-400 mb-4">Project Overview</h3>
                <div className="space-y-2 text-sm mb-6">
                    <p><span className="font-semibold">Title:</span> {formData.title || 'Your Project Title'}</p>
                    <p><span className="font-semibold">Completion Date:</span> {formData.completionDate || 'Not set'}</p>
                    <p><span className="font-semibold">Contribution to Global Good:</span> {formData.contribution || 'Your contribution details will appear here.'}</p>
                </div>

```

```

        <h3 className="text-xl font-bold text-lime-400 mb-4">Activities</h3>
        <div className="space-y-3">
            {activities.length > 0 ? activities.map(act => (
                <div key={act.id} className="p-3 bg-gray-900 rounded-lg text-sm">
                    <p className="font-bold">{act.title}</p>
                    <p className="text-xs text-gray-400">{act.date} @ {act.time} - ({act.type})</p>
                </div>
            )) : <p className="text-sm text-gray-500">No activities defined.</p>}
        </div>
        <div className="mt-8 pt-6 border-t border-gray-700 flex justify-between items-center">
            <Button variant="secondary" onClick={() => setStep('form_i')}>Back to Edit</Button>
            <Button onClick={() => { alert("Project Published!"); setPage('ProjectHub'); }}>Publish Project</Button>
        </div>
        </Card>
    </div>
);
}

// This is Form I
return (
    <div className="max-w-4xl mx-auto py-8 px-4">
        <h2 className="text-4xl font-bold text-center mb-4">Global Good Project Creation</h2>
        <p className="text-center text-gray-400 mb-12">Build your project from the ground up. Your inputs will shape the final outcome.</p>
        <div className="space-y-8">
            <Card className="p-6"> <h3 className="text-xl font-bold mb-4">Mission Anchor</h3> <div className="bg-gray-900 p-4 rounded-lg"> <p className="text-lime-400 font-semibold">{mockUser.mission.name}</p> <p>

```

```
className="text-sm text-gray-400 mt-1">{mockUser.mission.description}</p>
</div> </Card>

<Card className="p-6">
  <h3 className="text-xl font-bold mb-4">Part 1: Core Logistics</h3>
  <div className="space-y-4">
    <Input name="title" label="Project Title" placeholder="A clear, inspiring name" required onChange={handleFormChange} />
    <div className="grid grid-cols-1 md:grid-cols-2 gap-4">
      <Input name="completionDate" label="By what date will the project be completed?" type="date" required onChange={handleFormChange} />
      <Input name="signupDeadline" label="Signup Deadline (Optional)" type="date" onChange={handleFormChange} />
    </div>
    <Input name="location" label="Location (inc. pin)" placeholder="e.g., City Park, Main Entrance" required onChange={handleFormChange} />
  </div>
</Card>

<Card className="p-6">
  <h3 className="text-xl font-bold mb-4">Part 2: The Heart of the Project</h3>
  <div className="space-y-4">
    <div className="relative">
      <Textarea name="contribution" label="Contribution to Global Good" placeholder="Describe the project's core purpose and positive impact..." rows={5} onChange={handleFormChange} required />
      <button onClick={() => alert("AI is analyzing your project to suggest ways to deepen its Global Good impact, focusing on")}>Submit</button>
    </div>
  </div>
</Card>
```

```

inclusivity and holistic well-being for all life."} className="absolute
top-0 right-0 mt-2 mr-2 p-1 text-gray-400 hover:text-lime-400">
    <Wand2 size={18}/>
</button>
</div>
<Textarea name="bring" label="Participant
Experience: What to Bring" placeholder="e.g., Water bottle, comfortable
shoes" onChange={handleFormChange} />
<Textarea name="provide" label="Participant
Experience: What We'll Provide" placeholder="e.g., Snacks, tools, safety
vests" onChange={handleFormChange} />
</div>
</Card>

<Card className="p-6">
    <h3 className="text-xl font-bold mb-4">Part 3: Activity
Breakdown & Verification Setup</h3>
    <div className="space-y-4">
        {activities.map(act => (
            <div key={act.id} className="flex items-center
justify-between bg-gray-900 p-3 rounded-lg">
                <div>
                    <p
className="font-semibold">{act.title}</p>
                    <p className="text-xs
text-gray-400">{act.date} @ {act.time} <span className={`ml-2 px-2 py-0.5
text-xs rounded-full ${act.type === 'In-Person' ? 'bg-blue-800' :
'bg-purple-800'}`}>{act.type}</span></p>
                </div>
                <Button variant="secondary"
className="!p-2" onClick={() => handleRemoveActivity(act.id)}><Trash2
size={16}/></Button>
            </div>
        ))}
        {isAddingActivity ? (

```

```

                <div className="p-4 border border-dashed
border-gray-700 rounded-lg space-y-4">
                    <Input name="title" label="Activity Title"
placeholder="e.g., Site Preparation" value={newActivity.title}
onChange={handleActivityChange} />
                    <div className="grid grid-cols-1
md:grid-cols-2 gap-4">
                        <Input name="date" label="Activity
Date" type="date" value={newActivity.date} onChange={handleActivityChange}
/>
                        <Input name="time" label="Activity
Time" type="time" value={newActivity.time} onChange={handleActivityChange}
/>
                    </div>
                    <div>
                        <label className="block text-sm
font-medium text-gray-300 mb-2">Activity Type</label>
                        <div className="flex gap-2">
                            <button onClick={() =>
setNewActivity(p => ({...p, type: 'In-Person'}))} className={`flex-1 py-2
text-sm rounded-md flex items-center justify-center gap-2
${newActivity.type === 'In-Person' ? 'bg-lime-400 text-black' :
'bg-gray-800'}`}><UserPlus size={16}/> In-Person</button>
                            <button onClick={() =>
setNewActivity(p => ({...p, type: 'Remote'}))} className={`flex-1 py-2
text-sm rounded-md flex items-center justify-center gap-2
${newActivity.type === 'Remote' ? 'bg-lime-400 text-black' :
'bg-gray-800'}`}><Globe size={16}/> Remote</button>
                        </div>
                    </div>
                    {newActivity.type === 'In-Person' && (
                        <div className="p-3 bg-blue-900/20
text-blue-300 text-sm rounded-lg">Verification will be via QR code scan at
the event.</div>
                    ) }
                    {newActivity.type === 'Remote' && (

```

```

                                <Input name="proof" label="Define Proof
of Contribution" placeholder="e.g., Upload final design file"
value={newActivity.proof} onChange={handleActivityChange} />
                            ) }
                    <div className="flex justify-end gap-2">
                        <Button variant="secondary" onClick={() =>
setIsAddingActivity(false)}>Cancel</Button>
                        <Button
                            onClick={handleSaveActivity}>Save Activity</Button>
                    </div>
                ) : (
                    <Button onClick={() =>
setIsAddingActivity(true)} variant="secondary" className="w-full">
                        <PlusCircle size={16} /> Add Activity
                    </Button>
                )
            </div>
        </Card>

        <Card className="p-6">
            <h3 className="text-xl font-bold mb-4">Part 4:
AI-Assisted Launch Pad</h3>
            <div className="space-y-6">
                <div> <h4 className="font-semibold
mb-2">AI-Generated Project Summary</h4> <div className="p-4 bg-gray-900
rounded-lg min-h-[100px] flex items-center justify-center"> <Button
onClick={() => alert("Generating AI summary...")}><Wand2 size={16} />
Generate Summary</Button> </div> </div>
                <div> <h4 className="font-semibold mb-2">AI
Elevation Tips</h4> <div className="p-4 bg-gray-900 rounded-lg text-sm
text-gray-400"> <p>[AI will provide incremental ideas here after summary
generation]</p> </div> </div>
                <div>

```

```

        <h4 className="font-semibold mb-2">Project
Invite Post</h4> <p className="text-sm text-gray-400 mb-2">This will be
automatically generated for the feed.</p>
        <div className="p-4 bg-gray-900 rounded-lg">
            <div className="bg-black rounded-lg
overflow-hidden">
                <div className="h-40 bg-cover bg-center
flex items-center justify-center text-center p-4" style={{

backgroundImage:
`url(${placeholder.co/800x400/1a1a1a/a3e636?text=${encodeURIComponent(
formData.title || "Your Project")}})` }}></div>
                <div className="p-4"> <h3
className="font-bold text-lg">{formData.title || "Your Project
Title"}</h3> <p className="text-sm text-gray-400 mt-2">A new project has
been launched by {mockUser.name}. Click to view details and join the
mission!</p> </div>
                </div>
            </div>
        </div>
    </div>
    <div className="mt-8 pt-6 border-t border-gray-700 flex
justify-end items-center">
        <Button onClick={() => setStep('review')}>Review &
Publish</Button>
    </div>
</div>
);
};

// --- Main App Shell (for demonstration) ---
export default function App() {
    const [page, setPage] = useState('CreateProject');

    const renderPage = () => {

```

```
switch (page) {
  case 'CreateProject':
    return <CreateProject setPage={setPage} />;
  case 'ProjectHub':
    return <div className="text-center p-8">Navigated to the
Project Hub!</div>;
  default:
    return <CreateProject setPage={setPage} />;
}

return (
  <div className={`${colors.background} ${colors.text} min-h-screen
font-sans`}>
    <main>{renderPage()}</main>
  </div>
);
}
```