

```

import React, { useState, useEffect, useRef } from 'react';
import { Search, User, Compass, BookOpen, MessageSquare, PlusCircle, Star,
Users, Briefcase, Heart, Wind, Share2, Filter, MapPin, MoreVertical,
Trash2, Bookmark, Flag, X, Sparkles, CheckCircle, Edit3, Send,
ChevronDown, ShieldAlert, Zap, ArrowRight, Sun, Users2, Target, Bell,
Video, ThumbsUp, ThumbsDown, Wand2, Image as ImageIcon, Upload, LogOut,
Check, UserPlus, Globe, MessageCircle, Navigation, ChevronLeft,
ChevronRight, UsersRound } from 'lucide-react';

// --- MOCK DATA (REVISED FOR NEW DECENTRALIZED STRUCTURE) ---
const mockUsers = {
  101: {
    id: 101,
    name: 'Alexina B.',
    avatar: 'https://placeholder.co/100x100/a3e636/101010?text=AB',
    followers: 124,
    following: 56,
    followedTags: ['Community Gardens', 'Reforestation'],
    ggCredits: { leader: 6, collaborator: 17, mobilised: 53 },
    activeProject: { id: 7, title: "Leiderdorp Community Cleanup",
location: "Leiderdorp Center" },
    vision: "A world with flourishing ecosystems, teeming with
beautiful nature and wildlife, where humans nurture our planet as one
human family.",
    mission: { name: "Urban Greening Initiative", description: "To
design and implement sustainable green spaces within urban environments,
fostering community engagement and increasing local biodiversity." },
    tags: ["Community Gardens", "Reforestation", "Wildlife Habitats"],
    globalGoodSummary: "Alexina is translating her vision for a
flourishing planet into tangible action through her Urban Greening
Initiative. Her work not only contributes to local ecosystem health but
also aligns with a broader global movement towards sustainable cities. By
mobilizing dozens of volunteers, she is amplifying her impact, creating a
ripple effect of positive change that extends far beyond her individual
projects."
  }
}

```

```
    portfolioSummary: "Her mission is to design and implement sustainable green spaces within urban environments, fostering community engagement and increasing local biodiversity. Alexina's portfolio demonstrates a clear focus on this through her work. By leading projects like 'City Hall Rooftop Garden,' she has proven her ability to execute from concept to completion. Her story from this project highlights a talent for fostering connection and shared purpose among volunteers.",
    pinnedProjects: [6, 3, 4], storyHighlight: 1,
    cvSnapshot: "A seasoned Project Manager with 5 years of experience, Alexina excels in logistics and team coordination. Her background in Environmental Science provides a strong foundation for her goodwill projects, effectively bridging her professional skills with her personal mission.",
    storyPosts: [
      { id: 1, title: "A Garden Takes Root!", project: "City Hall Rooftop Garden", role: "Leader", content: "Seeing the first sprouts in our rooftop garden was a moment of pure magic...", imageUrls: ["https://placeholder.co/800x600/101010/FFFFFF?text=Garden+Sprouts", "https://placeholder.co/800x600/101010/FFFFFF?text=Teamwork", "https://placeholder.co/800x600/101010/FFFFFF?text=Harvest"] },
      { id: 2, title: "Teamwork Makes the Dream Work", project: "Coastal Cleanup", role: "Collaborator", content: "Collaborating with 'Eco Warriors' was an incredible experience...", imageUrls: ["https://placeholder.co/800x600/101010/FFFFFF?text=Cleanup+Team"] },
    ],
    allProjects: [
      { id: 6, title: "Pollinator Havens Project", roleType: 'leader', missionName: "Urban Greening Initiative", imageUrl: "https://placeholder.co/600x400/101010/FFFFFF?text=Pollinator+Haven", tags: ["Wildlife Habitats", "Community Gardens"], date: "2025-08-15" },
      { id: 4, title: "Coastal Cleanup", roleType: 'collaborator', missionName: "Eco Warriors", imageUrl: "https://placeholder.co/600x400/101010/FFFFFF?text=Coastal+Cleanup", tags: ["Ocean Cleanup"], date: "2025-06-30" },
    ],
  },
```

```
    201: { id: 201, name: 'David Kim', mission: { name: "Youth Empowerment"
}, followers: 88, following: 120},
    202: { id: 202, name: 'Sarah Chen', mission: { name: "Tech for All" },
followers: 250, following: 90},
    203: { id: 203, name: 'John S.', mission: { name: "River Savers" },
followers: 45, following: 30},
    204: { id: 204, name: 'Maria Garcia', mission: { name: "Creative
Action" }, followers: 312, following: 150},
    205: { id: 205, name: 'Peter V.', mission: { name: "Clean Canals
Leiden" }, followers: 76, following: 44},
};

let currentUser = mockUsers[101];

const mockFeed = [
  { id: 4, type: 'Project Invite', authorId: 204, title: 'Lend Your
Design Skills for a Local Animal Shelter', description: "Our local animal
shelter needs a new, heartwarming logo and branding guide to help their
furry friends find forever homes. If you're a graphic designer,
illustrator, or branding expert, we'd love your help!", imageUrls:
['https://placeholder.co/800x600/1a1a1a/a3e636?text=Design+for+Paws'], tags:
['Design', 'Animals', 'Community', 'Skills-Based'], location: "San Diego,
CA, USA", remote: true },
  { id: 1, type: 'Project Story', authorId: 201, project: 'Solar Schools
Initiative', title: 'How Teaching Kids About Solar Changed My Life',
description: "Reflecting on my experience with the Solar Schools
Initiative. Seeing children's faces light up when they understood how
sunlight becomes electricity was magical...", imageUrls:
['https://placeholder.co/800x600/1a1a1a/a3e636?text=Solar+Schools',
'https://placeholder.co/800x600/1a1a1a/a3e636?text=Kids+Learning'], tags:
['Education', 'Solar', 'Inspiration', 'Youth'], location: "Amsterdam,
Netherlands" },
  { id: 2, type: 'Project Impact', authorId: 202, project: 'Digital
Literacy Program', title: 'Digital Literacy Program Graduates 45 Seniors',
description: 'Our 8-week digital literacy program successfully helped 45
seniors learn essential computer skills, email, and video calling.'
```

```
imageUrls:
['https://placeholder.co/800x600/1a1a1a/a3e636?text=Digital+Graduates',
'https://placeholder.co/800x600/1a1a1a/a3e636?text=Learning+Together',
'https://placeholder.co/800x600/1a1a1a/a3e636?text=Success!'], metrics:
{'Seniors Graduated': 45, 'Volunteers': 12}, tags: ['Community',
'Seniors', 'Tech', 'Success'], location: "Online" },
  { id: 3, type: 'Project Story', authorId: 203, project: 'River Cleanup
Initiative', title: 'A Human Chain of Good', description: "I met so many
amazing people today and felt a real sense of purpose...", imageUrls:
['https://placeholder.co/800x600/1a1a1a/a3e636?text=Human+Chain'], tags:
['Reflection', 'Teamwork', 'Inspiration', 'Community'], location: "River
Amstel, Amsterdam" },
  { id: 5, type: 'Project Invite', authorId: 205, title: 'Weekend Canal
Cleanup', description: "Join us this weekend to help keep Leiden's
beautiful canals clean! We provide the gear, you provide the enthusiasm.",
imageUrls:
['https://placeholder.co/800x600/1a1a1a/38bdf8?text=Canal+Cleanup'], tags:
['Environment', 'Community', 'Leiden'], location: "Leiden, Netherlands",
remote: false },
];
```

```
const mockLibrary = [
  { id: 1, title: 'Annual Food Drive', mission: "Jane D.'s Mission",
description: 'Organized a city-wide food drive to support local
families.', tags: ['Humanitarian Aid', 'Community'] },
];
```

```
const mockProject = {
  id: 6,
  title: "Pollinator Havens Project",
  mission: { name: "Urban Greening Initiative" },
  leader: { name: "Alexina B." },
  completionDate: "2025-08-15",
  location: "City Park, Community Garden Section",
  collaborators: [ {id: 201, name: "David Kim"}, {id: 202, name: "Sarah
Chen"} ],
```

```

    activities: [
      { id: 1, title: "Site Preparation & Soil Testing", description:
"Clearing the designated area and testing soil quality.", date:
"2025-07-20", time: "10:00 - 13:00", location: "Community Garden",
signedUp: true },
    ]
  };

const mockTopics = ['All', 'Environment', 'Community', 'Education',
'Tech', 'Animals', 'Humanitarian Aid', 'Conservation'];
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', accent:
'#a3e636', border: 'border-gray-700' };

// --- HELPER COMPONENTS ---
const Section = ({ title, children, className = "" }) => ( <div
className={`py-8 ${className}`}> <h2 className={`text-2xl font-bold mb-6
${colors.primary}`}>{title}</h2> {children} </div> );
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, defaultValue, name }) => ( <div> <label className="block
text-sm font-medium text-gray-300 mb-1">{label}</label> <textarea
placeholder={placeholder} value={value} defaultValue={defaultValue}
onChange={onChange} required={required} rows={rows} 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 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> );
};

const InteractiveTag = ({ children, isFollowed, onFollow }) => {
  const [showFollow, setShowFollow] = useState(false);
  return (
    <a href="#" onClick={(e) => e.preventDefault()} onMouseEnter={() =>
setShowFollow(true)} onMouseLeave={() => setShowFollow(false)}
className="relative inline-block bg-gray-800 text-gray-300 text-xs
font-medium mr-2 mb-2 px-2.5 py-1 rounded-full transition-colors
hover:bg-gray-700">
      {children}
      {showFollow && (
        <button onClick={onFollow} className={`absolute -right-1
-top-1 rounded-full p-0.5 transition-opacity ${isFollowed ? 'bg-lime-500'
: 'bg-gray-600'}`}>
          {isFollowed ? <Check size={10} className="text-black"/>
: <PlusCircle size={10}/>}
        </button>
      )}
    </a>
  );
};

const GGCreditDisplay = ({ credits, followers, following, setPage }) => {

```

```

    const creditItems = [ { label: 'Leader', value: credits?.leader || 0 },
    { label: 'Collaborator', value: credits?.collaborator || 0 }, { label:
'Mobilised', value: credits?.mobilised || 0 } ];
    return (
      <div className="mt-2 space-y-1">
        <div onClick={() => setPage('GlobalGood')} className="flex
flex-wrap gap-x-4 gap-y-1 text-sm text-center cursor-pointer group">
          {creditItems.map(item => ( <span key={item.label}
className={` ${colors.textMuted} group-hover:text-white
transition-colors`} > {item.label} <span className={` ${colors.primary}
font-bold`} > {item.value} </span> </span> ))}
        </div>
        <div className="flex flex-wrap gap-x-4 text-sm text-gray-500">
          <span><span className="font-bold text-gray-400">{followers
|| 0}</span> Followers</span>
          <span><span className="font-bold text-gray-400">{following
|| 0}</span> Following</span>
        </div>
      </div>
    );
  };
};

// --- CORE COMPONENTS
const Header = ({ setPage, isLoggedIn, setIsLoggedIn, notifications,
readNotifications }) => {
  const [showCreateMenu, setShowCreateMenu] = useState(false);
  const [showNotifications, setShowNotifications] = useState(false);
  const createMenuRef = useRef(null);
  const notificationMenuRef = useRef(null);
  const hasMission = currentUser && currentUser.mission;
  const unreadNotifications = notifications.filter(n => !n.read).length;

  const handleCreateProjectClick = () => {
    setShowCreateMenu(false);
    if (!hasMission) {
      setPage('CreateMission');
    }
  };
};

```

```

        setTimeout(() => alert("Let's create your Mission first! A
Mission is required before you can start a project."), 100);
    } else { setPage('CreateProject'); }
};

const handleNotificationClick = (notification) => {
    readNotifications(notification.id);
    setPage(notification.link);
    setShowNotifications(false);
};

useEffect(() => {
    const handleClickOutside = (event) => {
        if (createMenuRef.current &&
!createMenuRef.current.contains(event.target)) { setShowCreateMenu(false);
    }

        if (notificationMenuRef.current &&
!notificationMenuRef.current.contains(event.target)) {
setShowNotifications(false); }
    };
    document.addEventListener("mousedown", handleClickOutside);
    return () => document.removeEventListener("mousedown",
handleClickOutside);
}, []);

return (
    <header className={` ${colors.background} sticky top-0 z-50 py-3
px-4 md:px-8 border-b ${colors.border}`}>
        <div className="container mx-auto flex justify-between
items-center">
            <h1 className="text-3xl font-bold tracking-tighter
cursor-pointer" onClick={() => setPage('Feed')}>OHM</h1>
            <nav className="hidden md:flex items-center gap-6 text-sm
font-medium">
                <a href="#" onClick={() => setPage('Feed')}
className={` ${colors.textMuted} hover:${colors.text}`}>Feed</a>

```



```

        <a href="#" onClick={() => setPage('Library')}
className={` ${colors.textMuted} hover:${colors.text}`}>Living Library</a>
        <a href="#" onClick={() => setPage('MyGlobalGood')}
className={` ${colors.textMuted} hover:${colors.text}`}>My Global Good</a>
    </nav>
    <div className="flex items-center gap-4">
        {isLoggedIn ? (
            <>
                <Button variant="secondary" className="!p-2
rounded-full" onClick={() => alert("AI Guide Activated!")}> <Sparkles
size={18} /> </Button>

                <div className="relative" ref={createMenuRef}>
                    <Button onClick={() =>
setShowCreateMenu(!showCreateMenu)} variant="secondary" className="px-3">
<PlusCircle size={20} /> </Button>

                    {showCreateMenu && (
                        <div className="absolute right-0 mt-2
w-48 bg-gray-900 border border-gray-700 rounded-md shadow-lg py-1 z-50">
                            <a href="#" onClick={() => {
setPage('CreateMission'); setShowCreateMenu(false); }} className="block
px-4 py-2 text-sm text-gray-300 hover:bg-gray-800">Create Mission</a>

                            <a href="#"
onClick={handleCreateProjectClick} className="block px-4 py-2 text-sm
text-gray-300 hover:bg-gray-800">Create Project</a>
                        </div>
                    )}
                </div>

                <div className="relative"
ref={notificationMenuRef}>
                    <Button variant="secondary"
className="!px-3 relative" onClick={() =>
setShowNotifications(!showNotifications)}>

                        <Bell size={20} />

                        {unreadNotifications > 0 && <span
className="absolute -top-1 -right-1 h-4 w-4 rounded-full bg-red-500
text-xs flex items-center justify-center">{unreadNotifications}</span>}

```

```

        </Button>
        {showNotifications && (
            <div className="absolute right-0 mt-2
w-80 bg-gray-900 border border-gray-700 rounded-md shadow-lg z-50">
                <div className="p-3 border-b
border-gray-700"><h3 className="font-bold">Notifications</h3></div>
                <div className="max-h-96
overflow-y-auto">
                    {notifications.length > 0 ?
notifications.map(n => (
                        <div key={n.id} onClick={()
=> handleNotificationClick(n)} className={`p-3 border-b border-gray-700
hover:bg-gray-800 cursor-pointer ${!n.read ? 'bg-lime-900/20' : ''}`}>
                            <p
className="text-sm">{n.text}</p>
                            <p className="text-xs
text-gray-500 mt-1">{new Date(n.id).toLocaleTimeString()}</p>
                        </div>
                    )) : <p className="p-4 text-sm
text-gray-500">No new notifications.</p>}
                </div>
            </div>
        )}
    </div>
    <Button onClick={() => setIsLoggedIn(false)}
variant="secondary" className="!px-3">Logout size={20}/></Button>
    </>
    ) : ( <Button onClick={() =>
setPage('Subscribe')}>Subscribe</Button> )}
    </div>
</div>
</header>
);
};

const PostCard = ({ post, setPage }) => {

```

```

const [showMenu, setShowMenu] = useState(false);
const menuRef = useRef(null);
const author = mockUsers[post.authorId];
const isAuthor = author.id === currentUser.id;
const [isFollowed, setIsFollowed] = useState(false);
const [isLiked, setIsLiked] = useState(false);
const [currentImageIndex, setCurrentImageIndex] = useState(0);

const hasMultipleImages = post.imageUrls && post.imageUrls.length > 1;

const nextImage = (e) => {
  e.stopPropagation();
  setCurrentImageIndex((prevIndex) => (prevIndex + 1) %
post.imageUrls.length);
};

const prevImage = (e) => {
  e.stopPropagation();
  setCurrentImageIndex((prevIndex) => (prevIndex - 1 +
post.imageUrls.length) % post.imageUrls.length);
};

useEffect(() => {
  const handleClickOutside = (event) => { if (menuRef.current &&
!menuRef.current.contains(event.target)) { setShowMenu(false); } };
  document.addEventListener("mousedown", handleClickOutside);
  return () => document.removeEventListener("mousedown",
handleClickOutside);
}, []);

const PostTypeLabel = () => {
  const icons = {
    'Project Invite': <Zap size={14} className="inline-block mr-2"
/>,
    'Project Impact': <Sparkles size={14} className="inline-block
mr-2" />,

```

```

        'Project Story': <BookOpen size={14} className="inline-block
mr-2" />,
    };

    return <p className={` ${colors.primary} font-semibold text-sm mb-2
flex items-center`} >{icons[post.type]} {post.type.toUpperCase()}</p>;
    };

    const displayLocation = post.location ? post.location.split(',')[0] :
'Remote';

    return (
        <Card className="p-0 overflow-hidden border-b-4 border-gray-900
rounded-lg">
            <div className="p-4 flex justify-between items-center">
                <div>
                    <a href="#" onClick={() => setPage('MyGlobalGood')}
className="font-semibold text-sm hover:underline">{author.name} <span
className="text-gray-500 font-normal">• "{author.mission.name}"</span></a>
                    <p className="text-xs text-gray-500
mt-1">{displayLocation}</p>
                </div>
                <div className="flex items-center gap-2">
                    {!isAuthor && (
                        <button onClick={() => setIsFollowed(!isFollowed)}
className="text-gray-500 hover:text-lime-400">
                            {isFollowed ? <Check size={20}
className="text-lime-400"/> : <UserPlus size={20}/>}
                        </button>
                    )}
                    <button onClick={() => setShowMenu(true)}
className="text-gray-400 hover:text-white"><MoreVertical size={20}
/></button>
                    {showMenu && (
                        <div ref={menuRef} className="absolute right-4
mt-20 w-56 bg-gray-900 border border-gray-700 rounded-md shadow-lg py-1
z-20">

```

```

        <button onClick={() => { alert("Post
reported."); setShowMenu(false); }} className="w-full flex items-center
px-4 py-2 text-sm text-red-400 hover:bg-gray-800"> <Flag size={16}
className="mr-2" /> Report Post </button>
    </div>
  )}
</div>
</div>

<div className="relative">
  <img src={post.imageUrls[currentImageIndex]}
alt={post.title} className="w-full h-auto aspect-square object-cover" />
  {hasMultipleImages && (
    <>
      <button onClick={prevImage} className="absolute
left-2 top-1/2 -translate-y-1/2 bg-black/50 text-white rounded-full p-1
hover:bg-black/80"><ChevronLeft size={20}/></button>
      <button onClick={nextImage} className="absolute
right-2 top-1/2 -translate-y-1/2 bg-black/50 text-white rounded-full p-1
hover:bg-black/80"><ChevronRight size={20}/></button>
      <div className="absolute bottom-2 left-1/2
-translate-x-1/2 flex gap-1.5">
        {post.imageUrls.map((_, index) => (
          <div key={index} className={`w-1.5 h-1.5
rounded-full ${index === currentImageIndex ? 'bg-white' :
'bg-white/50'}`} ></div>
        ))}
      </div>
    </>
  )}
</div>

<div className="p-4">
  <div className="flex items-center justify-between">
    <div className="flex items-center gap-4">
      <button onClick={() => setIsLiked(!isLiked)}>

```

```

                <Heart size={24} className={`transition-colors
${isLiked ? 'text-red-500 fill-current' : 'hover:text-gray-400'}}`/>
            </button>
            <button><MessageSquare size={24}
className="hover:text-gray-400"/></button>
            <button><Share2 size={24}
className="hover:text-gray-400"/></button>
        </div>
        <button><Bookmark size={24}
className="hover:text-gray-400"/></button>
    </div>

    <div className="mt-4">
        <PostTypeLabel/>
        <h3 className="text-xl font-bold">{post.title}</h3>
        {post.type === 'Project Invite' &&
            <Button onClick={() => setPage('ProjectHubJoin')}
className="w-full my-4">Join Project</Button>
        }
        {post.type === 'Project Impact' &&
            <Button onClick={() => alert("Navigating to Project
Card...")} className="w-full my-4">View Full Project</Button>
        }
        {post.type === 'Project Story' &&
            <p className="text-sm text-gray-400 my-2">
                From the <a href="#" onClick={() =>
alert("Navigating to Project Card...")} className="text-lime-400
hover:underline font-semibold">"{post.project}"</a> project
            </p>
        }
        <p className="text-gray-400 mt-2
text-sm">{post.description}</p>
        <a href="#" className="text-gray-500 text-sm mt-2
block">View all comments</a>
    </div>

```

```

        <div className="mt-4 pt-4 border-t border-gray-800">
          {post.tags.map(tag =>
            <InteractiveTag
              key={tag}

isFollowed={currentUser.followedTags.includes(tag)}
              onFollow={() => alert(`Toggled follow for
${tag}`)}
            >
              {tag}
            </InteractiveTag>
          )}
        </div>
      </div>
    </Card>
  );
};

const Feed = ({ setPage }) => {
  const [activeTab, setActiveTab] = useState('river');
  const [locationSearch, setLocationSearch] = useState("Leiderdorp,
Netherlands");
  const [filterRemote, setFilterRemote] = useState(false);

  const riverContent = mockFeed;

  const opportunitiesContent = mockFeed.filter(post => {
    if (post.type !== 'Project Invite') return false;
    if (filterRemote) return post.remote === true;
    return
post.location?.toLowerCase().includes(locationSearch.split(', ')[0].toLowerCase());
  });

  return (
    <div>

```

```

        <div className="py-4 border-b border-gray-700 sticky top-[65px]
bg-black z-40">
            <div className="container mx-auto px-4">
                <div className="flex gap-4 border-b border-gray-800
mb-4">
                    <button onClick={() => setActiveTab('river')}
className={`py-2 px-1 font-semibold ${activeTab === 'river' ?
'text-lime-400 border-b-2 border-lime-400' : 'text-gray-400'}`}>The
River</button>
                    <button onClick={() =>
setActiveTab('opportunities')} className={`py-2 px-1 font-semibold
${activeTab === 'opportunities' ? 'text-lime-400 border-b-2
border-lime-400' : 'text-gray-400'}`}>Opportunities</button>
                </div>
                {activeTab === 'opportunities' && (
                    <div className="flex flex-col md:flex-row gap-2
items-center">
                        <div className="relative w-full md:flex-1">
                            <MapPin size={18} className="absolute
left-3 top-1/2 -translate-y-1/2 text-gray-500"/>
                            <input
                                type="text"
                                value={locationSearch}
                                onChange={(e) =>
setLocationSearch(e.target.value)}
                                placeholder="Search by city..."
                                className="w-full bg-gray-900 border
border-gray-700 rounded-lg py-2 pl-10 pr-4 focus:outline-none focus:ring-2
focus:ring-lime-400"
                            />
                        </div>
                        <Button variant="secondary" className="!p-2"
onClick={() => setLocationSearch("Leiderdorp, Netherlands")}><Navigation
size={18}/></Button>
                    <div className="flex items-center gap-2 p-1
bg-gray-900 rounded-full border border-gray-700">

```



```

        <button onClick={() =>
setFilterRemote(false)} className={`px-3 py-1 rounded-full text-sm
${!filterRemote ? 'bg-lime-400 text-black' : ''}`}>In-Person</button>
        <button onClick={() =>
setFilterRemote(true)} className={`px-3 py-1 rounded-full text-sm
${filterRemote ? 'bg-lime-400 text-black' : ''}`}>Remote</button>
      </div>
    </div>
  )}
</div>
</div>
<div className="max-w-xl mx-auto py-8 px-4">
  <div className="space-y-4">
    {(activeTab === 'river' ? riverContent :
opportunitiesContent).map(post => (
      <PostCard key={post.id} post={post}
setPage={setPage} />
    ))}
  </div>
</div>
</div>
);
};

const MyGlobalGood = ({ setPage }) => {
  const [view, setView] = useState('public');
  const [activeTab, setActiveTab] = useState('Impact Resume');
  const user = currentUser;
  const getRoleIcon = (roleType) => roleType.includes('leader') ? Star :
Users;

  const ImpactResume = () => {
    const pinnedProjects = user.allProjects.filter(p =>
user.pinnedProjects.includes(p.id));

```

```

    const storyHighlight = user.storyPosts.find(s => s.id ===
user.storyHighlight);
    return (
      <div className="space-y-8">
        <Card className="p-6"> <h3 className="text-lg font-semibold
text-gray-400 mb-2">My Global Good Vision</h3> <p className="text-xl
italic">"{user.vision}"</p> </Card>
        <Card className="p-6"> <h3 className="text-lg font-semibold
text-gray-400 mb-2">Global Good Summary</h3>
<p>{user.globalGoodSummary}</p> <a href="#" onClick={() =>
setPage('GlobalGood')} className="text-sm text-lime-400 font-semibold mt-4
inline-block">See Global Good Overview →</a> </Card>
        <Card className="p-6">
          <h3 className="text-lg font-semibold text-gray-400
mb-2">My Mission</h3> <h2 className="text-2xl font-bold
mb-4">{user.mission?.name}</h2> <p className="text-gray-400 italic
mb-6">"{user.portfolioSummary}"</p>
          <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
            {pinnedProjects.map(item => {
              const Icon = getRoleIcon(item.roleType);
              return (
                <div key={item.id} className="rounded-lg
overflow-hidden border border-gray-700 group cursor-pointer">
                  <div className="relative"> <img
src={item.imageUrl} alt={item.title} className="w-full h-40 object-cover"
/> <div className="absolute top-2 right-2 flex items-center gap-2 text-xs
bg-black/70 text-white p-1 rounded-md"> <Icon size={12} />
<span>{item.roleType}</span></div> </div>
                  <div className="p-4 bg-gray-900"> <h4
className="font-bold truncate">{item.title}</h4> <p className="text-sm
text-lime-400 truncate">{item.missionName}</p> </div>
                </div>
              )
            })}
          </div>
        </Card>
      </div>
    )
  )
}

```

```

        <a href="#" onClick={() => setActiveTab('Portfolio')}
className="text-sm text-lime-400 font-semibold mt-6 inline-block">See Full
Portfolio →</a>
      </Card>
      {storyHighlight && ( <Card className="p-6"> <h3
className="text-lg font-semibold text-gray-400 mb-2">Story Highlight</h3>
<img src={storyHighlight.imageUrls[0]} alt={storyHighlight.title}
className="w-full h-48 object-cover rounded-lg mb-4" /> <h4
className="text-xl font-bold">{storyHighlight.title}</h4> <p
className="text-gray-400 my-2">"{storyHighlight.content}"</p> <a href="#"
onClick={() => setActiveTab('Story')} className="text-sm text-lime-400
font-semibold mt-2 inline-block">See All Project Stories →</a> </Card> )}
      <Card className="p-6"> <h3 className="text-lg font-semibold
text-gray-400 mb-2">CV Snapshot</h3> <p>{user.cvSnapshot}</p> <a href="#"
onClick={() => setPage('FullCV')} className="text-sm text-lime-400
font-semibold mt-4 inline-block">See Full CV →</a> </Card>
      <div className="mt-12 flex justify-center"> <Button
variant="secondary"><Share2 size={16} /> Share Impact Resume</Button>
</div>
    </div>
  );
};

const PortfolioPage = () => {
  const [activeTag, setActiveTag] = useState('All');
  const allUserTags = ['All', ...new Set(user.allProjects.flatMap(p
=> p.tags))];
  const filteredProjects = user.allProjects.filter(p => activeTag ===
'All' || p.tags.includes(activeTag));
  return (
    <div>
      <h2 className="text-3xl font-bold
mb-2">{user.mission?.name}</h2> <p className="text-gray-400
mb-8">{user.mission?.description}</p>
      <div className="mb-6">

```

```

        <p className="text-gray-400 mb-4">Filter by Impact
Tag:</p>

        <div className="flex flex-wrap gap-2">
{allUserTags.map(tag => ( <button key={tag} onClick={() =>
setActiveTag(tag)} className={`px-3 py-1 text-xs rounded-full ${activeTag
=== tag ? 'bg-white text-black' : 'bg-gray-800'}`}>{tag}</button> ))}
</div>

        </div>
        <div className="grid grid-cols-1 md:grid-cols-3 gap-6">
            {filteredProjects.map(item => {
                const Icon = getRoleIcon(item.roleType);
                return (
                    <div key={item.id} className="rounded-lg
overflow-hidden border border-gray-700 group cursor-pointer" onClick={()
=> setPage('ProjectHub')}>
                        <div className="relative"> <img
src={item.imageUrl} alt={item.title} className="w-full h-40 object-cover"
/> <div className="absolute top-2 right-2 flex items-center gap-2 text-xs
bg-black/70 text-white p-1 rounded-md"> <Icon size={12} />
<span>{item.roleType}</span></div> </div>
                            <div className="p-4 bg-gray-900"> <h4
className="font-bold truncate">{item.title}</h4> <p className="text-sm
text-lime-400 truncate">{item.missionName}</p> </div>
                                </div>
                            );
                        })}
                    </div>
                </div>
            );
        };

const StoryPage = () => (
    <div>
        <Card className="mb-8 p-6"> <h2 className="text-2xl font-bold
mb-4">The Story Behind the Stories</h2> <p className="text-gray-400
italic">"[AI-generated narrative summarizing the themes and highlights

```

```

from all the user's stories goes here, creating a compelling personal
journey.]"</p> </Card>

    <div className="space-y-8">
      {user.storyPosts.map(story => (
        <Card key={story.id} className="p-6"> <img
src={story.imageUrls[0]} alt={story.title} className="w-full h-48
object-cover rounded-lg mb-4" /> <h3 className="text-xl
font-bold">{story.title}</h3> <p className="text-sm text-gray-500
my-1">From "{story.project}" ({story.role})</p> <p
className="text-gray-400 mt-2">"{story.content}"</p> </Card>
      )
    )}
    </div>
  </div>
);

const MySuite = () => (
  <div>
    <GlobalGoodPage setPage={setPage} isDashboardView={true} />
    <Card className="mt-8 p-6 cursor-pointer hover:bg-gray-800"
onClick={() => alert("Navigating to full Growth Path page...")}>
      <h3 className="font-bold text-xl mb-2 flex items-center
gap-2 text-lime-400"><Sparkles size={20} /> Guide My Path</h3> <p
className="text-sm text-gray-400">{user.growthPath.summary}</p> <p
className="text-sm text-lime-400 font-semibold mt-4">Continue to AI Guide
-></p>
    </Card>
  </div>
);

const renderActiveTab = () => {
  switch (activeTab) {
    case 'Impact Resume': return <ImpactResume />;
    case 'Portfolio': return <PortfolioPage />;
    case 'Story': return <StoryPage />;
    default: return <ImpactResume />;
  }
}

```

```

};

return (
  <div className="max-w-5xl mx-auto py-8 px-4">
    <div className="flex flex-col md:flex-row items-start
md:items-center justify-between mb-8">
      <div>
        <h2 className="text-4xl font-bold">{user.name}</h2>
        <GGCCreditDisplay credits={user.ggCredits}
followers={user.followers} following={user.following} setPage={setPage} />
      </div>
      <div className="flex items-center gap-4 mt-4 md:mt-0">
        <Button><UserPlus size={16}/> Follow</Button>
        <Button variant="secondary"><Send size={16} />
Message</Button>
        <div className="flex items-center gap-2 p-1 bg-gray-900
rounded-full border border-gray-700">
          <button onClick={() => setView('public')}
className={`px-4 py-1 rounded-full text-sm ${view === 'public' ?
'bg-lime-400 text-black' : ''}`>Public View</button>
          <button onClick={() => setView('private')}
className={`px-4 py-1 rounded-full text-sm ${view === 'private' ?
'bg-lime-400 text-black' : ''}`>My Suite</button>
        </div>
      </div>
    </div>
    {view === 'public' ? (
      <>
        {user.activeProject && (
          <Card className="mb-8 bg-lime-900/30
border-lime-700 p-6">
            <h3 className="text-lg font-semibold
text-lime-300 mb-2">Join {user.name.split(' ')[0]} 's Current Project</h3>
            <div className="flex flex-col sm:flex-row
justify-between items-start sm:items-center">
              <div>

```

```

                <p className="text-xl
font-bold">{user.activeProject.title}</p>
                <p className="text-sm text-gray-400
flex items-center gap-2 mt-1"><MapPin size={14}/>
{user.activeProject.location}</p>
            </div>
            <Button onClick={() =>
setPage('ProjectHubJoin')} className="mt-4 sm:mt-0">
                View & Join
            </Button>
        </div>
    </Card>
    )}
    <div className="flex border-b border-gray-700">
        {[ 'Impact Resume', 'Portfolio', 'Story' ].map(tab =>
( <button key={tab} onClick={() => setActiveTab(tab)} className={`py-2
px-4 font-semibold ${activeTab === tab ? 'text-lime-400 border-b-2
border-lime-400' : 'text-gray-400'} `}> {tab} </button> ))}
    </div>
    <div className="mt-8"> {renderActiveTab()} </div>
</>
    ) : <MySuite />
</div>
);
};

const LivingLibrary = ({ setPage }) => (
    <div className="max-w-5xl mx-auto py-8 px-4">
        <h2 className="text-4xl font-bold text-center mb-4">The Living
Library</h2> <p className="text-center text-gray-400 mb-12">An
ever-growing Inspiration Engine for global good.</p>
        <div className="relative mb-8 max-w-2xl mx-auto"> <Search
className="absolute left-4 top-1/2 -translate-y-1/2 text-gray-500" />
<input type="text" placeholder="Search for projects by title, tag, or
mission..." className="w-full bg-gray-900 border border-gray-700
rounded-lg py-3 pl-12 pr-4 focus:outline-none focus:ring-2
focus:ring-lime-400" /> </div>
    </div>

```

```

    <div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3
gap-6">
      {mockLibrary.map(item => (
        <Card key={item.id} className="flex flex-col p-6">
          <div className="flex-grow"> <p className="text-sm
font-semibold text-lime-400">{item.mission}</p> <h3 className="text-xl
font-bold mt-1 mb-2">{item.title}</h3> <p className="text-gray-400 text-sm
mb-4">{item.description}</p> <div> {item.tags.map(tag => <InteractiveTag
key={tag}>{tag}</InteractiveTag>)} </div> </div>
          <Button className="mt-6 w-full"><PlusCircle size={16}
/> Use as Inspiration</Button>
        </Card>
      ))}
    </div>
  );

const CreateProject = ({ setPage }) => {
  // ... This component is large and remains unchanged from the version
  with the AI planner chat
  return <div>Create Project Placeholder</div>
};

const CreateMission = ({ setPage }) => {
  // ... This component is unchanged
  return <div>Create Mission Placeholder</div>
};

const SubscribeModal = ({ setPage, setIsLoggedIn }) => {
  // ... This component is unchanged
  return <div>Subscribe Placeholder</div>
};

const OnboardingVision = ({ setPage }) => {
  // ... This component is unchanged
  return <div>Onboarding Placeholder</div>
};

const GlobalGoodPage = ({ setPage, isDashboardView = false }) =>
<div>Global Good Page Placeholder</div>;

```



```

const FullCVPage = ({ setPage }) => <div>Full CV Page Placeholder</div>;
const ProjectHub = ({ setPage, project }) => <div>Project Hub
Placeholder</div>;
const ProjectHubJoin = ({ setPage, project }) => <div>Project Hub Join
Placeholder</div>;
const CompleteProject = ({ setPage }) => <div>Complete Project Form
Placeholder</div>;

// --- MAIN APP ---
export default function App() {
  const [page, setPage] = useState('Feed');
  const [isLoggedIn, setIsLoggedIn] = useState(true);
  const [notifications, setNotifications] = useState([
    { id: Date.now() - 10000, text: "David Kim has joined your
'Pollinator Havens Project'.", link: "ProjectHub", read: false },
    { id: Date.now() - 50000, text: "Your 'City Hall Rooftop Garden'
impact post received a new comment.", link: "Feed", read: true }
  ]);

  const readNotifications = (id) => { setNotifications(prev => prev.map(n
=> n.id === id ? { ...n, read: true } : n)); };

  const renderPage = () => {
    if (page === 'Subscribe') { return ( <> <Feed setPage={setPage} />
<SubscribeModal setPage={setPage} setIsLoggedIn={setIsLoggedIn} /> </> );
  }

  switch (page) {
    case 'Feed': return <Feed setPage={setPage} />;
    case 'MyGlobalGood': return <MyGlobalGood setPage={setPage} />;
    case 'Library': return <LivingLibrary setPage={setPage} />;
    case 'CreateProject': return <CreateProject setPage={setPage}
/>;

    case 'CreateMission': return <CreateMission setPage={setPage}
/>;

    case 'Onboarding': return <OnboardingVision setPage={setPage}
/>;
  }
}

```

```

        case 'GlobalGood': return <GlobalGoodPage setPage={setPage} />;
        case 'FullCV': return <FullCVPage setPage={setPage} />;
        case 'ProjectHub': return <ProjectHub setPage={setPage}
project={mockProject} />;
        case 'ProjectHubJoin': return <ProjectHubJoin setPage={setPage}
project={mockProject} />;
        case 'CompleteProject': return <CompleteProject
setPage={setPage} />;
        default: return <Feed setPage={setPage} />;
    }
};

return (
    <div className={` ${colors.background} ${colors.text} min-h-screen
font-sans`} >
        <style>{`
            input[type="date"]::-webkit-calendar-picker-indicator,
            input[type="time"]::-webkit-calendar-picker-indicator {
                filter: invert(1);
            }
        `}</style>
        <Header setPage={setPage} isLoggedIn={isLoggedIn}
setIsLoggedIn={setIsLoggedIn} notifications={notifications}
readNotifications={readNotifications} />
        <main> {renderPage()} </main>
        <footer className={`border-t ${colors.border} py-8 mt-12`} >
            <div className="container mx-auto text-center
text-gray-500">
                <p>OHM for Global Good</p>
                <p className="text-sm mt-2">Building a purpose-driven
economy, one act of goodwill at a time.</p>
            </div>
        </footer>
    </div>
);
}

```

