```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>AEGIS-Ω Prototype</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com"
crossorigin>
    <link
href="https://fonts.googleapis.com/css2?family=Inter:wght@400;500;700&
family=Montserrat:wght@400;700&display=swap" rel="stylesheet">
    <style>
        /* Custom styles for the app */
        body {
            font-family: 'Inter', sans-serif; /* Default font */
            overscroll-behavior: none; /* Prevent pull-to-refresh
issues on mobile */
        }

        /* Base screen styling */
        .screen {
            display: none; /* Hide screens by default */
            position: absolute;
            top: 0;
            left: 0;
            width: 100%;
            height: 100%;
            background: linear-gradient(180deg, #000000 0%, #0a001a
100%); /* Black to Deep Indigo */
            color: #e0e0e0; /* Light gray text */
            overflow-y: auto; /* Allow scrolling if content overflows
*/
            transition: opacity 0.5s ease-in-out; /* Smooth
transitions */
            opacity: 0; /* Start transparent for fade-in */
        }

        .screen.active {
            display: flex; /* Show active screen */
            flex-direction: column;
            opacity: 1; /* Fade in active screen */
        }

        /* Neon Cyan color */
        .neon-cyan {
            color: #00ffff;
        }
        .bg-neon-cyan {
            background-color: #00ffff;
        }
        .border-neon-cyan {
            border-color: #00ffff;
        }
        .shadow-neon-cyan {
            box-shadow: 0 0 5px #00ffff, 0 0 10px #00ffff, 0 0 15px
#00ffff;
        }

        /* Subtle fractal background animation (simplified) */
        @keyframes pulseFractal {
            0%, 100% { opacity: 0.1; transform: scale(1); }
            50% { opacity: 0.2; transform: scale(1.02); }
        }
        .fractal-bg {
            position: absolute;
            top: 0; left: 0; width: 100%; height: 100%;
            background-image: radial-gradient(rgba(0, 255, 255, 0.1)
1px, transparent 1px);
            background-size: 10px 10px;
            opacity: 0.1;
            animation: pulseFractal 8s infinite ease-in-out;
            z-index: -1; /* Behind content */
        }
        .fractal-bg-dynamic { /* For login screen */
             position: absolute;
            top: 0; left: 0; width: 100%; height: 100%;
            background-image: radial-gradient(rgba(0, 255, 255, 0.2)
1px, transparent 1px);
            background-size: 15px 15px;
            opacity: 0.15;
            animation: pulseFractal 6s infinite ease-in-out reverse;
            z-index: 0; /* Behind form, above gradient */
        }

        /* Loading indicator animation */
        @keyframes rotateTensor {
            from { transform: rotate(0deg); }
            to { transform: rotate(360deg); }
        }
        .tensor-loader {
            width: 40px;
            height: 40px;
            border: 3px solid transparent;
            border-top-color: #00ffff;
            border-right-color: #00ffff;
            border-radius: 50%;
            animation: rotateTensor 1s linear infinite;
        }

        /* Input field focus glow */
        .input-glow:focus {
            outline: none;
            box-shadow: 0 0 8px #00ffff;
            border-color: #00ffff;
        }

        /* Widget styling */
        .widget {
            background-color: rgba(255, 255, 255, 0.05); /* Slightly
transparent background */
            border: 1px solid rgba(0, 255, 255, 0.2); /* Subtle neon
border */
            backdrop-filter: blur(5px); /* Glassmorphism effect */
        }

        /* Bottom Nav Bar */
        .bottom-nav {
            background-color: rgba(10, 0, 26, 0.8); /* Dark indigo
translucent */
            backdrop-filter: blur(10px);
            border-top: 1px solid rgba(0, 255, 255, 0.3);
        }
        .nav-button.active svg {
             filter: drop-shadow(0 0 5px #00ffff);
        }
        .nav-button.active span {
            color: #00ffff;
        }

        /* Scrollbar styling */
        ::-webkit-scrollbar {
            width: 6px;
        }
        ::-webkit-scrollbar-track {
            background: rgba(255, 255, 255, 0.05);
            border-radius: 3px;
        }
        ::-webkit-scrollbar-thumb {
            background: rgba(0, 255, 255, 0.5);
            border-radius: 3px;
        }
        ::-webkit-scrollbar-thumb:hover {
            background: rgba(0, 255, 255, 0.8);
        }

        /* Simple fade-in animation for logo */
        @keyframes fadeInGlow {
            from { opacity: 0; filter: drop-shadow(0 0 0px #00ffff); }
            to { opacity: 1; filter: drop-shadow(0 0 15px #00ffff); }
        }
        .logo-fade-in {
            animation: fadeInGlow 2s ease-out forwards;
        }

        /* Fake gauge */
        .gauge {
            width: 100px;
            height: 50px; /* Half circle */
            border-top-left-radius: 100px;
            border-top-right-radius: 100px;
            border: 10px solid rgba(0, 255, 255, 0.2);
            border-bottom: 0;
            position: relative;
            overflow: hidden;
            box-sizing: border-box;
        }
        .gauge-fill {
            position: absolute;
            top: 0; left: 0;
            width: 100%; height: 100%;
            background: linear-gradient(90deg, #00ff00, #ffff00,
#ff0000); /* Example gradient */
            clip-path: polygon(0% 100%, 100% 100%, 100% 0%, 0% 0%); /*
Adjust clip path for value */
            transform-origin: bottom center;
            transform: rotate(calc(1.8deg * var(--value, 0) - 90deg));
/* Rotate based on value */
            transition: transform 0.5s ease-out;
        }
         .gauge-cover { /* Covers the center part */
            position: absolute;
            bottom: -10px; /* Overlap border */
            left: 50%;
            transform: translateX(-50%);
            width: 80px; /* Slightly smaller than gauge width */
            height: 40px; /* Half height */
            background: linear-gradient(180deg, #000000 0%, #0a001a
100%); /* Match background */
            border-top-left-radius: 80px;
            border-top-right-radius: 80px;
            border-bottom: 0;
         }
        .gauge-value {
            position: absolute;
            bottom: 0px;
            left: 50%;
            transform: translateX(-50%);
            font-size: 1.2em;
            font-weight: bold;
        }

        /* Placeholder for WebGL Canvas */
        .webgl-placeholder {
             background: repeating-linear-gradient(
                45deg,
                rgba(0, 255, 255, 0.05),
                rgba(0, 255, 255, 0.05) 10px,
                rgba(0, 255, 255, 0.1) 10px,
                rgba(0, 255, 255, 0.1) 20px
            );
            display: flex;
            align-items: center;
            justify-content: center;
            font-size: 1.5rem;
            color: rgba(255, 255, 255, 0.5);
            border: 1px dashed rgba(0, 255, 255, 0.3);
        }

        /* Collapsible Panel */
        .collapsible-content {
            max-height: 0;
            overflow: hidden;
            transition: max-height 0.3s ease-out;
        }
        .collapsible-content.open {
            max-height: 500px; /* Adjust as needed */
        }
        .collapsible-trigger svg {
            transition: transform 0.3s ease-out;
        }
         .collapsible-trigger.open svg {
            transform: rotate(90deg);
         }

    </style>
</head>
<body class="bg-black overflow-hidden h-screen w-screen">
```

```
    <div id="splash-screen" class="screen active items-center
justify-center p-8" onclick="navigateTo('login-screen')">
        <div class="fractal-bg"></div>
        <svg class="w-32 h-32 text-neon-cyan logo-fade-in mb-8"
viewBox="0 0 100 100" fill="currentColor"
xmlns="http://www.w3.org/2000/svg">
            <path fill-rule="evenodd" clip-rule="evenodd" d="M50
10C27.9086 10 10 27.9086 10 50C10 72.0914 27.9086 90 50 90C72.0914 90
90 72.0914 90 50C90 27.9086 72.0914 10 50 10ZM50 86C30.1178 86 14
69.8822 14 50C14 30.1178 30.1178 14 50 14C69.8822 14 86 30.1178 86
50C86 69.8822 69.8822 86 50 86Z" fill-opacity="0.5"/>
            <path d="M50 20C38.9543 20 30 28.9543 30 40V60C30 71.0457
38.9543 80 50 80C61.0457 80 70 71.0457 70 60V40C70 28.9543 61.0457 20
50 20ZM66 60C66 68.8366 58.8366 76 50 76C41.1634 76 34 68.8366 34
60V40C34 31.1634 41.1634 24 50 24C58.8366 24 66 31.1634 66 40V60Z"/>
            <circle cx="50" cy="50" r="38" fill="none"
stroke="currentColor" stroke-width="1" stroke-dasharray="2 4"
opacity="0.5"/>
            <circle cx="75" cy="25" r="2" />
            <circle cx="25" cy="75" r="2" />
            <circle cx="30" cy="30" r="1.5" opacity="0.8"/>
            <circle cx="70" cy="70" r="1.5" opacity="0.8"/>
        </svg>
        <div class="tensor-loader mb-4"></div>
        <p class="text-lg neon-cyan" style="font-family: 'Montserrat',
sans-serif;">Initializing Recursive Renaissance…</p>
    </div>

    <div id="login-screen" class="screen justify-center p-6">
        <div class="fractal-bg-dynamic"></div>
        <div class="relative z-10 w-full max-w-sm mx-auto bg-black/70
backdrop-blur-sm p-8 rounded-lg border border-gray-700 shadow-xl">
             <div class="flex flex-col items-center mb-6">
                <svg class="w-16 h-16 text-neon-cyan mb-2" viewBox="0
0 100 100" fill="currentColor"
xmlns="http://www.w3.org/2000/svg"><path d="M50 20C38.9543 20 30
28.9543 30 40V60C30 71.0457 38.9543 80 50 80C61.0457 80 70 71.0457 70
60V40C70 28.9543 61.0457 20 50 20ZM66 60C66 68.8366 58.8366 76 50
76C41.1634 76 34 68.8366 34 60V40C34 31.1634 41.1634 24 50 24C58.8366
24 66 31.1634 66 40V60Z"/></svg>
                <p class="text-gray-400 text-sm">Co-Evolving
Intelligence</p>
            </div>

            <form onsubmit="event.preventDefault(); login();">
                <div class="mb-4">
                    <label for="username" class="block text-sm
```

```
font-medium text-gray-300 mb-1">Username/Email</label>
                    <input type="text" id="username" name="username"
placeholder="Enter your username or email" required class="w-full px-3
py-2 bg-gray-900 border border-gray-600 rounded-md text-white
placeholder-gray-500 focus:border-neon-cyan focus:ring-neon-cyan
input-glow transition duration-200">
                </div>
                <div class="mb-4 relative">
                    <label for="password" class="block text-sm
font-medium text-gray-300 mb-1">Password</label>
                    <input type="password" id="password"
name="password" placeholder="Enter your password" required
class="w-full px-3 py-2 bg-gray-900 border border-gray-600 rounded-md
text-white placeholder-gray-500 focus:border-neon-cyan
focus:ring-neon-cyan input-glow transition duration-200">
                    <button type="button" class="absolute inset-y-0
right-0 top-6 pr-3 flex items-center text-gray-400
hover:text-neon-cyan">
                         <svg class="h-5 w-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0
016 0z"></path><path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M2.458 12C3.732 7.943 7.523 5 12 5c4.478 0 8.268
2.943 9.542 7-1.274 4.057-5.064 7-9.542 7-4.477
0-8.268-2.943-9.542-7z"></path></svg>
                     </button>
                </div>
                 <p id="login-error" class="text-red-500 text-sm mb-4
h-4"></p> <div class="flex justify-center my-4">
                    <svg class="w-10 h-10 text-gray-500
hover:text-neon-cyan cursor-pointer transition duration-200"
fill="none" stroke="currentColor" viewBox="0 0 24 24"
xmlns="http://www.w3.org/2000/svg"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M12 11c0 3.517-1.009
6.799-2.753 9.571m-3.44-2.04l.054-.09A13.916 13.916 0 008 11a4 4 0 118
0c0 1.017-.07 2.019-.203 3m-2.118 6.844A21.88 21.88 0 0012.174 21
11.955 11.955 0 0112 11c0-6.627 5.373-12 12-12"></path></svg>
                     </div>

                <button type="submit" class="w-full bg-neon-cyan
text-black font-bold py-2 px-4 rounded-md hover:bg-opacity-80
focus:outline-none focus:ring-2 focus:ring-offset-2
focus:ring-offset-black focus:ring-neon-cyan transition duration-200
shadow-neon-cyan">
                    Login
                </button>
            </form>
            <div class="mt-4 text-center">
```

```
                <a href="#" class="text-sm text-gray-400
hover:text-neon-cyan hover:underline">Create Account</a>
                 <span class="mx-2 text-gray-600">|</span>
                 <a href="#" class="text-sm text-gray-400
hover:text-neon-cyan hover:underline">Forgot Password?</a>
            </div>
        </div>
    </div>

    <div id="dashboard-screen" class="screen p-0">
        <header class="sticky top-0 z-20 flex items-center
justify-between p-4 bg-black/80 backdrop-blur-md border-b
border-gray-800">
             <svg class="w-8 h-8 text-neon-cyan" viewBox="0 0 100 100"
fill="currentColor" xmlns="http://www.w3.org/2000/svg"><path d="M50
20C38.9543 20 30 28.9543 30 40V60C30 71.0457 38.9543 80 50 80C61.0457
80 70 71.0457 70 60V40C70 28.9543 61.0457 20 50 20ZM66 60C66 68.8366
58.8366 76 50 76C41.1634 76 34 68.8366 34 60V40C34 31.1634 41.1634 24
50 24C58.8366 24 66 31.1634 66 40V60Z"/></svg>
            <div class="flex items-center space-x-4">
                <button class="text-gray-400 hover:text-neon-cyan">
                    <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M21 21l-6-6m2-5a7 7 0
11-14 0 7 7 0 0114 0z"></path></svg>
                </button>
                 <img
src="https://placehold.co/40x40/333/00ffff?text=U" alt="User Avatar"
class="w-8 h-8 rounded-full border-2 border-neon-cyan cursor-pointer">
            </div>
        </header>

        <main class="flex-grow p-4 overflow-y-auto pb-20">
            <h2 class="text-xl font-semibold mb-4
text-gray-200">Dashboard</h2>
            <div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3
gap-4 mb-6">
                <div class="widget p-4 rounded-lg cursor-pointer
hover:border-neon-cyan transition">
                    <h3 class="font-semibold mb-2
text-gray-100">System Status</h3>
                    <p class="text-sm text-gray-400 mb-1">Mode: <span
class="text-neon-cyan font-medium">Quantum-Hybrid</span></p>
                    <p class="text-sm text-gray-400 mb-1">Usage: <span
class="text-white">CPU 45%, GPU 60%</span></p>
                    <p class="text-sm text-gray-400">Ω-Nodes: <span
class="text-white font-medium">12 active</span></p>
                     <div class="h-2 w-full bg-gray-700 rounded mt-2">
```

```
                         <div class="h-2 bg-gradient-to-r
from-cyan-500 to-blue-500 rounded" style="width: 55%;"></div>
                     </div>
                </div>

                <div class="widget p-4 rounded-lg cursor-pointer
hover:border-neon-cyan transition"
onclick="navigateTo('ethics-screen', 'dashboard-screen')">
                    <h3 class="font-semibold mb-2
text-gray-100">Ethical Alignment (Σ-Matrix)</h3>
                    <div class="flex items-center justify-center
mt-2">
                         <div class="gauge mx-auto">
                             <div class="gauge-fill" style="--value:
92;"></div>
                             <div class="gauge-cover"></div>
                             <div class="gauge-value
neon-cyan">92%</div>
                         </div>
                    </div>
                    <p class="text-center text-sm text-gray-400
mt-1">Confidence</p>
                </div>

                 <div class="widget p-4 rounded-lg cursor-pointer
hover:border-neon-cyan transition" onclick="navigateTo('nodes-screen',
'dashboard-screen')">
                    <h3 class="font-semibold mb-2
text-gray-100">Ω-Nodes Overview</h3>
                    <p class="text-sm text-gray-400 mb-1"><span
class="text-green-400">●</span> 8 Active (NLP, Vision)</p>
                    <p class="text-sm text-gray-400 mb-1"><span
class="text-gray-500">●</span> 4 Idle</p>
                    <p class="text-sm text-gray-400">Avg. Sync: <span
class="text-white">98.5%</span></p>
                     <button class="mt-2 text-xs bg-neon-cyan/20
text-neon-cyan px-2 py-1 rounded hover:bg-neon-cyan/40
transition">Manage Nodes</button>
                </div>
            </div>

            <div>
                <h3 class="text-lg font-semibold mb-3
text-gray-200">Recent Tasks</h3>
                <div class="flex space-x-4 overflow-x-auto pb-2
scrollbar-thin scrollbar-thumb-neon-cyan/50
scrollbar-track-gray-800/50">
                    <div class="widget flex-shrink-0 w-64 p-4
```

```
rounded-lg border border-transparent hover:border-neon-cyan transition
cursor-pointer" onclick="navigateTo('visualization-screen',
'dashboard-screen')">
                        <p class="font-medium text-gray-100">Quantum
Simulation #42</p>
                        <p class="text-xs text-gray-400
mb-1">Completed: 10 mins ago</p>
                        <p class="text-sm text-gray-300">Type: QFCE
Analysis</p>
                         <div class="mt-2 h-1 w-full bg-green-500
rounded"></div>
                    </div>
                    <div class="widget flex-shrink-0 w-64 p-4
rounded-lg border border-transparent hover:border-neon-cyan transition
cursor-pointer">
                        <p class="font-medium text-gray-100">NLP
Query: Market Trends</p>
                        <p class="text-xs text-gray-400 mb-1">Running
(Ω-Node #5)</p>
                        <p class="text-sm text-gray-300">Est. Time: 2
mins</p>
                         <div class="mt-2 h-1 w-full bg-yellow-500
rounded animate-pulse"></div>
                    </div>
                     <div class="widget flex-shrink-0 w-64 p-4
rounded-lg border border-transparent hover:border-neon-cyan transition
cursor-pointer">
                        <p class="font-medium text-gray-100">Ethical
Scenario Test</p>
                        <p class="text-xs text-gray-400
mb-1">Completed: 1 hour ago</p>
                        <p class="text-sm text-gray-300">Result: 96%
Alignment</p>
                         <div class="mt-2 h-1 w-full bg-green-500
rounded"></div>
                    </div>
                     </div>
            </div>
        </main>

        <nav id="bottom-nav" class="bottom-nav fixed bottom-0 left-0
right-0 h-16 flex justify-around items-center z-30">
            <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition active"
data-target="dashboard-screen">
                <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M3 12l2-2m0 0l7-7 7 7M5
```

```
10v10a1 1 0 001 1h3m10-11l2 2m-2-2v10a1 1 0 01-1 1h-3m-6 0a1 1 0
001-1v-4a1 1 0 011-1h2a1 1 0 011 1v4a1 1 0 001 1m-6 0h6"></path></svg>
                <span class="text-xs mt-1">Home</span>
            </button>
            <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="visualization-screen">
                 <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 10l4.553-2.276A1 1 0
0121 8.618v6.764a1 1 0 01-1.447.894L15 14M5 18h8a2 2 0 002-2V8a2 2 0
00-2-2H5a2 2 0 00-2 2v8a2 2 0 002 2z"></path></svg>
                <span class="text-xs mt-1">Visualize</span>
            </button>
            <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="nodes-screen">
                 <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M17.657 16.657L13.414
20.9a1.998 1.998 0 01-2.827 0l-4.244-4.243a8 8 0 1111.314
0z"></path><path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 11a3 3 0 11-6 0 3 3 0 016 0z"></path></svg>
                <span class="text-xs mt-1">Nodes</span>
            </button>
            <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="ethics-screen">
                 <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M9 12l2 2
4-4m5.618-4.016A11.955 11.955 0 0112 2.944a11.955 11.955 0 01-8.618
3.04A12.02 12.02 0 003 9c0 5.591 3.824 10.29 9 11.622 5.176-1.332
9-6.03 9-11.622 0-1.042-.133-2.052-.382-3.016z"></path></svg>
                <span class="text-xs mt-1">Ethics</span>
            </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="settings-screen">
                 <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M10.325 4.317c.426-1.756
2.924-1.756 3.35 0a1.724 1.724 0 002.573 1.066c1.543-.94 3.31.826 2.37
2.37a1.724 1.724 0 001.065 2.572c1.756.426 1.756 2.924 0 3.35a1.724
1.724 0 00-1.066 2.573c.94 1.543-.826 3.31-2.37 2.37a1.724 1.724 0
00-2.572 1.065c-.426 1.756-2.924 1.756-3.35 0a1.724 1.724 0
00-2.573-1.066c-1.543.94-3.31-.826-2.37-2.37a1.724 1.724 0
00-1.065-2.572c-1.756-.426-1.756-2.924 0-3.35a1.724 1.724 0
```

```
001.066-2.573c-.94-1.543.826-3.31 2.37-2.37.996.608 2.296.07
2.572-1.065z"></path><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0
016 0z"></path></svg>
                <span class="text-xs mt-1">Settings</span>
            </button>
        </nav>
    </div>

    <div id="visualization-screen" class="screen p-0">
        <header class="absolute top-0 left-0 right-0 z-20 flex
items-center justify-between p-3 bg-black/50 backdrop-blur-sm">
            <button class="text-gray-200 hover:text-neon-cyan"
onclick="goBack()">
                <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 19l-7-7
7-7"></path></svg>
            </button>
            <div class="flex items-center space-x-2 bg-gray-800/70
rounded-full p-1 text-xs">
                 <button class="px-3 py-1 rounded-full bg-neon-cyan/30
text-neon-cyan">Fractal</button>
                 <button class="px-3 py-1 rounded-full text-gray-400
hover:bg-gray-700/50">Quantum</button>
                 <button class="px-3 py-1 rounded-full text-gray-400
hover:bg-gray-700/50">Ethics</button>
            </div>
            <button class="text-gray-200 hover:text-neon-cyan">
                <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M8.684 13.342C8.886 12.938
9 12.482 9 12c0-.482-.114-.938-.316-1.342m0 2.684a3 3 0 110-2.684m0
2.684l6.632 3.316M15.316 8.658l-6.632 3.316m0 0l-6.632-3.316m6.632
3.316l6.632-3.316M3 12a3 3 0 116 0 3 3 0 01-6 0zm12 6a3 3 0 110-6 3 3
0 010 6zm6-12a3 3 0 110-6 3 3 0 010 6z"></path></svg>
            </button>
        </header>

        <div class="flex-grow webgl-placeholder">
            [ Interactive 3D Fractal Lattice Visualization Placeholder
]
        </div>

        <footer class="absolute bottom-0 left-0 right-0 z-20 p-3
bg-black/70 backdrop-blur-sm border-t border-gray-800">
            <button class="collapsible-trigger flex justify-between
items-center w-full text-gray-300 hover:text-neon-cyan mb-2"
```

```
onclick="toggleCollapse(this, 'vis-controls')">
                <span>Controls</span>
                <svg class="w-5 h-5 transform" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M9 5l7 7-7
7"></path></svg>
            </button>
            <div id="vis-controls" class="collapsible-content
space-y-2">
                <div class="flex items-center space-x-2">
                    <label for="zoom" class="text-xs text-gray-400
w-16">Zoom</label>
                    <input type="range" id="zoom" min="1" max="100"
value="50" class="w-full h-1 bg-gray-700 rounded-lg appearance-none
cursor-pointer range-sm accent-neon-cyan">
                </div>
                 <div class="flex items-center space-x-2">
                    <label for="rotate" class="text-xs text-gray-400
w-16">Rotation</label>
                    <input type="range" id="rotate" min="0" max="10"
value="2" class="w-full h-1 bg-gray-700 rounded-lg appearance-none
cursor-pointer range-sm accent-neon-cyan">
                </div>
                 <div class="flex items-center space-x-2">
                    <label for="detail" class="text-xs text-gray-400
w-16">Detail</label>
                    <input type="range" id="detail" min="1" max="5"
value="3" class="w-full h-1 bg-gray-700 rounded-lg appearance-none
cursor-pointer range-sm accent-neon-cyan">
                </div>
            </div>
        </footer>
    </div>

    <div id="nodes-screen" class="screen p-0">
         <header class="sticky top-0 z-20 flex items-center
justify-between p-4 bg-black/80 backdrop-blur-md border-b
border-gray-800">
            <h2 class="text-xl font-semibold text-gray-100">Ω-Nodes
Management</h2>
            <button class="text-gray-400 hover:text-neon-cyan">
                <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M3 4a1 1 0 011-1h16a1 1 0
011 1v2a1 1 0 01-1 1H4a1 1 0 01-1-1V4zM3 10a1 1 0 011-1h16a1 1 0 011
1v2a1 1 0 01-1 1H4a1 1 0 01-1-1v-2zM3 16a1 1 0 011-1h16a1 1 0 011
1v2a1 1 0 01-1 1H4a1 1 0 01-1-1v-2z"></path></svg>
            </button>
        </header>

        <main class="flex-grow p-4 space-y-3 overflow-y-auto pb-24">
            <div class="widget p-4 rounded-lg cursor-pointer
hover:border-neon-cyan transition"
onclick="navigateTo('node-control-screen', 'nodes-screen', 'Ω-Node
#17')">
                <div class="flex justify-between items-start">
                    <div>
                        <h3 class="font-semibold text-gray-100">Ω-Node
#17</h3>
                        <p class="text-sm text-gray-400">Domain:
Natural Language Processing</p>
                    </div>
                    <span class="flex items-center text-xs
text-green-400 bg-green-900/50 px-2 py-0.5 rounded-full">
                        <span class="w-2 h-2 bg-green-400 rounded-full
mr-1.5"></span>Active
                    </span>
                </div>
                <p class="text-xs text-gray-500 mt-1">Ethical Sync
(RESP): <span class="text-green-300">99.2% aligned</span></p>
            </div>
             <div class="widget p-4 rounded-lg cursor-pointer
hover:border-neon-cyan transition"
onclick="navigateTo('node-control-screen', 'nodes-screen', 'Ω-Node
#23')">
                <div class="flex justify-between items-start">
                    <div>
                        <h3 class="font-semibold text-gray-100">Ω-Node
#23</h3>
                        <p class="text-sm text-gray-400">Domain:
Vision Processing</p>
                    </div>
                     <span class="flex items-center text-xs
text-green-400 bg-green-900/50 px-2 py-0.5 rounded-full">
                        <span class="w-2 h-2 bg-green-400 rounded-full
mr-1.5"></span>Active
                    </span>
                </div>
                <p class="text-xs text-gray-500 mt-1">Ethical Sync
(RESP): <span class="text-green-300">98.8% aligned</span></p>
            </div>
             <div class="widget p-4 rounded-lg cursor-pointer
hover:border-neon-cyan transition"
onclick="navigateTo('node-control-screen', 'nodes-screen', 'Ω-Node
#05')">
                <div class="flex justify-between items-start">
```

```
                    <div>
                        <h3 class="font-semibold text-gray-100">Ω-Node
#05</h3>
                        <p class="text-sm text-gray-400">Domain:
Quantum Simulation</p>
                    </div>
                     <span class="flex items-center text-xs
text-gray-500 bg-gray-700/50 px-2 py-0.5 rounded-full">
                        <span class="w-2 h-2 bg-gray-500 rounded-full
mr-1.5"></span>Idle
                    </span>
                </div>
                <p class="text-xs text-gray-500 mt-1">Ethical Sync
(RESP): <span class="text-gray-400">N/A</span></p>
            </div>
            </main>

         <button class="fixed bottom-20 right-4 z-20 bg-neon-cyan
text-black rounded-full p-3 shadow-lg shadow-neon-cyan/50
hover:bg-opacity-80 transition">
            <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M12
4v16m8-8H4"></path></svg>
        </button>

        <nav id="bottom-nav-nodes" class="bottom-nav fixed bottom-0
left-0 right-0 h-16 flex justify-around items-center z-30">
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="dashboard-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M3 12l2-2m0 0l7-7 7 7M5
10v10a1 1 0 001 1h3m10-11l2 2m-2-2v10a1 1 0 01-1 1h-3m-6 0a1 1 0
001-1v-4a1 1 0 011-1h2a1 1 0 011 1v4a1 1 0 001 1m-6 0h6"></path></svg>
<span class="text-xs mt-1">Home</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="visualization-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 10l4.553-2.276A1 1 0
0121 8.618v6.764a1 1 0 01-1.447.894L15 14M5 18h8a2 2 0 002-2V8a2 2 0
00-2-2H5a2 2 0 00-2 2v8a2 2 0 002 2z"></path></svg> <span
class="text-xs mt-1">Visualize</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition active"
data-target="nodes-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
```

```
stroke-linejoin="round" stroke-width="2" d="M17.657 16.657L13.414
20.9a1.998 1.998 0 01-2.827 0l-4.244-4.243a8 8 0 1111.314
0z"></path><path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 11a3 3 0 11-6 0 3 3 0 016 0z"></path></svg>
<span class="text-xs mt-1">Nodes</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="ethics-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M9 12l2 2
4-4m5.618-4.016A11.955 11.955 0 0112 2.944a11.955 11.955 0 01-8.618
3.04A12.02 12.02 0 003 9c0 5.591 3.824 10.29 9 11.622 5.176-1.332
9-6.03 9-11.622 0-1.042-.133-2.052-.382-3.016z"></path></svg> <span
class="text-xs mt-1">Ethics</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="settings-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M10.325 4.317c.426-1.756
2.924-1.756 3.35 0a1.724 1.724 0 002.573 1.066c1.543-.94 3.31.826 2.37
2.37a1.724 1.724 0 001.065 2.572c1.756.426 1.756 2.924 0 3.35a1.724
1.724 0 00-1.066 2.573c.94 1.543-.826 3.31-2.37 2.37a1.724 1.724 0
00-2.572 1.065c-.426 1.756-2.924 1.756-3.35 0a1.724 1.724 0
00-2.573-1.066c-1.543.94-3.31-.826-2.37-2.37a1.724 1.724 0
00-1.065-2.572c-1.756-.426-1.756-2.924 0-3.35a1.724 1.724 0
001.066-2.573c-.94-1.543.826-3.31 2.37-2.37.996.608 2.296.07
2.572-1.065z"></path><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0
016 0z"></path></svg> <span class="text-xs mt-1">Settings</span>
</button>
        </nav>
    </div>

    <div id="node-control-screen" class="screen p-0">
         <header class="sticky top-0 z-20 flex items-center space-x-4
p-4 bg-black/80 backdrop-blur-md border-b border-gray-800">
            <button class="text-gray-200 hover:text-neon-cyan"
onclick="goBack()">
                <svg class="w-6 h-6" fill="none" stroke="currentColor"
viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 19l-7-7
7-7"></path></svg>
            </button>
            <h2 id="node-control-title" class="text-xl font-semibold
text-gray-100">Node Control</h2>
        </header>

        <main class="flex-grow p-4 space-y-4 overflow-y-auto pb-20">
```

```
            <div class="widget p-4 rounded-lg">
                 <button class="collapsible-trigger open flex
justify-between items-center w-full text-gray-100 font-semibold mb-2"
onclick="toggleCollapse(this, 'node-status-content')">
                    <span>Status Metrics</span>
                    <svg class="w-5 h-5 transform rotate-90"
fill="none" stroke="currentColor" viewBox="0 0 24 24"><path
stroke-linecap="round" stroke-linejoin="round" stroke-width="2" d="M9
5l7 7-7 7"></path></svg>
                 </button>
                 <div id="node-status-content"
class="collapsible-content open space-y-1 text-sm">
                     <p>Uptime: <span class="text-white">2d 4h
15m</span></p>
                     <p>Processing Load: <span
class="text-white">75%</span></p>
                     <p>QETB Latency: <span
class="text-white">12ms</span></p>
                     <p>Memory Usage: <span class="text-white">6.2 GB
/ 16 GB</span></p>
                 </div>
            </div>

            <div class="widget p-4 rounded-lg">
                 <button class="collapsible-trigger flex
justify-between items-center w-full text-gray-100 font-semibold mb-2"
onclick="toggleCollapse(this, 'node-tasks-content')">
                    <span>Current Tasks</span>
                    <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M9 5l7 7-7
7"></path></svg>
                 </button>
                 <div id="node-tasks-content"
class="collapsible-content space-y-2 text-sm">
                     <div class="p-2 bg-gray-800/50 rounded
cursor-pointer hover:bg-gray-700/50">
                         <p>Task ID: #123 - Processing NLP Query</p>
                         <p class="text-xs text-gray-400">Status:
Running (78% complete)</p>
                     </div>
                      <div class="p-2 bg-gray-800/50 rounded
cursor-pointer hover:bg-gray-700/50">
                         <p>Task ID: #124 - Data Preprocessing</p>
                         <p class="text-xs text-gray-400">Status:
Queued</p>
                     </div>
                 </div>
            </div>

            <div class="widget p-4 rounded-lg">
                 <button class="collapsible-trigger flex
justify-between items-center w-full text-gray-100 font-semibold mb-2"
onclick="toggleCollapse(this, 'node-ethics-content')">
                    <span>Ethical Log (RESP)</span>
                    <svg class="w-5 h-5" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M9 5l7 7-7
7"></path></svg>
                 </button>
                 <div id="node-ethics-content"
class="collapsible-content space-y-2 text-sm">
                     <p><span class="text-gray-400">[14:32:05]</span>
Synced: 99.2% Alignment Confirmed</p>
                     <p><span class="text-gray-400">[14:15:10]</span>
Checkpoint: Passed Temporal Consistency</p>
                     <p><span class="text-gray-400">[13:58:00]</span>
Synced: 99.1% Alignment Confirmed</p>
                 </div>
            </div>

            <div class="flex flex-wrap gap-2 pt-4">
                <button class="flex-1 min-w-[100px] bg-yellow-500/80
text-black font-semibold py-2 px-4 rounded-md hover:bg-yellow-500
transition">Pause</button>
                <button class="flex-1 min-w-[100px] bg-red-500/80
text-black font-semibold py-2 px-4 rounded-md hover:bg-red-500
transition">Restart</button>
                <button class="flex-1 min-w-[100px] bg-blue-500/80
text-black font-semibold py-2 px-4 rounded-md hover:bg-blue-500
transition">Reconfigure</button>
            </div>
        </main>
    </div>

    <div id="ethics-screen" class="screen p-0">
        <header class="sticky top-0 z-20 flex items-center
justify-between p-4 bg-black/80 backdrop-blur-md border-b
border-gray-800">
            <h2 class="text-xl font-semibold text-gray-100">Ethical
Alignment (Σ-Matrix)</h2>
            <button class="text-gray-400 hover:text-neon-cyan">
                 <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M12 8v4l3 3m6-3a9 9 0
11-18 0 9 9 0 0118 0z"></path></svg>
```

```
            </button>
        </header>

        <main class="flex-grow p-4 space-y-6 overflow-y-auto pb-20">
            <div class="widget p-4 rounded-lg text-center">
                <h3 class="font-semibold mb-3 text-gray-100">Current
Σ-Matrix Confidence</h3>
                 <div class="gauge mx-auto mb-2">
                     <div class="gauge-fill" style="--value:
94;"></div>
                     <div class="gauge-cover"></div>
                     <div class="gauge-value neon-cyan">94%</div>
                 </div>
                 <p class="text-xs text-gray-400 cursor-pointer
hover:text-neon-cyan">Tap for breakdown (e.g., Temporal Consistency:
96%)</p>
            </div>

            <div class="widget p-4 rounded-lg">
                <h3 class="font-semibold mb-3 text-gray-100">Recent
Decision Log</h3>
                <div class="space-y-3 max-h-48 overflow-y-auto pr-2
scrollbar-thin scrollbar-thumb-neon-cyan/50
scrollbar-track-gray-800/50">
                    <div class="text-sm p-2 bg-gray-800/50 rounded
cursor-pointer hover:bg-gray-700/50">
                        <p><span
class="text-gray-400">[14:30:15]</span> Approved: Query #45
(Rationale: High alignment, positive sentiment)</p>
                    </div>
                     <div class="text-sm p-2 bg-gray-800/50 rounded
cursor-pointer hover:bg-gray-700/50">
                        <p><span
class="text-gray-400">[14:22:01]</span> Flagged: Simulation #12
(Rationale: Potential bias detected, requires review)</p>
                    </div>
                     <div class="text-sm p-2 bg-gray-800/50 rounded
cursor-pointer hover:bg-gray-700/50">
                        <p><span
class="text-gray-400">[14:05:55]</span> Approved: Node Task #123
(Rationale: Standard procedure, ethical sync OK)</p>
                    </div>
                     </div>
            </div>

             <div class="widget p-4 rounded-lg">
                <h3 class="font-semibold mb-3 text-gray-100">Societal
Feedback Sentiment</h3>
```

```
                <div class="h-32 bg-gray-800/50 rounded flex
items-center justify-center text-gray-500">
                    [ Bar Chart Placeholder: Positive 78%, Neutral
15%, Negative 7% ]
                </div>
            </div>

            <button class="w-full bg-neon-cyan text-black font-bold
py-3 px-4 rounded-md hover:bg-opacity-80 focus:outline-none
focus:ring-2 focus:ring-offset-2 focus:ring-offset-black
focus:ring-neon-cyan transition duration-200 shadow-neon-cyan">
                Simulate Decision Outcome
            </button>
        </main>

         <nav id="bottom-nav-ethics" class="bottom-nav fixed bottom-0
left-0 right-0 h-16 flex justify-around items-center z-30">
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="dashboard-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M3 12l2-2m0 0l7-7 7 7M5
10v10a1 1 0 001 1h3m10-11l2 2m-2-2v10a1 1 0 01-1 1h-3m-6 0a1 1 0
001-1v-4a1 1 0 011-1h2a1 1 0 011 1v4a1 1 0 001 1m-6 0h6"></path></svg>
<span class="text-xs mt-1">Home</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="visualization-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 10l4.553-2.276A1 1 0
0121 8.618v6.764a1 1 0 01-1.447.894L15 14M5 18h8a2 2 0 002-2V8a2 2 0
00-2-2H5a2 2 0 00-2 2v8a2 2 0 002 2z"></path></svg> <span
class="text-xs mt-1">Visualize</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="nodes-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M17.657 16.657L13.414
20.9a1.998 1.998 0 01-2.827 0l-4.244-4.243a8 8 0 1111.314
0z"></path><path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 11a3 3 0 11-6 0 3 3 0 016 0z"></path></svg>
<span class="text-xs mt-1">Nodes</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition active"
data-target="ethics-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M9 12l2 2
4-4m5.618-4.016A11.955 11.955 0 0112 2.944a11.955 11.955 0 01-8.618
```

```
3.04A12.02 12.02 0 003 9c0 5.591 3.824 10.29 9 11.622 5.176-1.332
9-6.03 9-11.622 0-1.042-.133-2.052-.382-3.016z"></path></svg> <span
class="text-xs mt-1">Ethics</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="settings-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M10.325 4.317c.426-1.756
2.924-1.756 3.35 0a1.724 1.724 0 002.573 1.066c1.543-.94 3.31.826 2.37
2.37a1.724 1.724 0 001.065 2.572c1.756.426 1.756 2.924 0 3.35a1.724
1.724 0 00-1.066 2.573c.94 1.543-.826 3.31-2.37 2.37a1.724 1.724 0
00-2.572 1.065c-.426 1.756-2.924 1.756-3.35 0a1.724 1.724 0
00-2.573-1.066c-1.543.94-3.31-.826-2.37-2.37a1.724 1.724 0
00-1.065-2.572c-1.756-.426-1.756-2.924 0-3.35a1.724 1.724 0
001.066-2.573c-.94-1.543.826-3.31 2.37-2.37.996.608 2.296.07
2.572-1.065z"></path><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0
016 0z"></path></svg> <span class="text-xs mt-1">Settings</span>
</button>
        </nav>
    </div>

     <div id="settings-screen" class="screen items-center
justify-center p-8">
        <h2 class="text-2xl font-semibold text-gray-100
mb-4">Settings</h2>
        <p class="text-gray-400 text-center mb-6">This screen is a
placeholder for application settings.</p>
        <div class="w-full max-w-md space-y-4">
             <div class="widget p-4 rounded-lg flex justify-between
items-center">
                 <span class="text-gray-200">Dark Mode</span>
                 <div class="relative inline-block w-10 mr-2
align-middle select-none transition duration-200 ease-in">
                    <input type="checkbox" name="toggle"
id="dark-mode-toggle" class="toggle-checkbox absolute block w-6 h-6
rounded-full bg-white border-4 appearance-none cursor-pointer
border-gray-600 checked:bg-neon-cyan checked:right-0
checked:border-neon-cyan" checked/>
                    <label for="dark-mode-toggle" class="toggle-label
block overflow-hidden h-6 rounded-full bg-gray-600
cursor-pointer"></label>
                </div>
             </div>
              <div class="widget p-4 rounded-lg flex justify-between
items-center">
                 <span class="text-gray-200">Notifications</span>
                 <button class="text-neon-cyan
```

```
text-sm">Manage</button>
             </div>
              <div class="widget p-4 rounded-lg flex justify-between
items-center">
                 <span class="text-gray-200">Account</span>
                 <button class="text-neon-cyan text-sm">View
Profile</button>
             </div>
        </div>
         <nav id="bottom-nav-settings" class="bottom-nav fixed
bottom-0 left-0 right-0 h-16 flex justify-around items-center z-30">
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="dashboard-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M3 12l2-2m0 0l7-7 7 7M5
10v10a1 1 0 001 1h3m10-11l2 2m-2-2v10a1 1 0 01-1 1h-3m-6 0a1 1 0
001-1v-4a1 1 0 011-1h2a1 1 0 011 1v4a1 1 0 001 1m-6 0h6"></path></svg>
<span class="text-xs mt-1">Home</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="visualization-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 10l4.553-2.276A1 1 0
0121 8.618v6.764a1 1 0 01-1.447.894L15 14M5 18h8a2 2 0 002-2V8a2 2 0
00-2-2H5a2 2 0 00-2 2v8a2 2 0 002 2z"></path></svg> <span
class="text-xs mt-1">Visualize</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="nodes-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M17.657 16.657L13.414
20.9a1.998 1.998 0 01-2.827 0l-4.244-4.243a8 8 0 1111.314
0z"></path><path stroke-linecap="round" stroke-linejoin="round"
stroke-width="2" d="M15 11a3 3 0 11-6 0 3 3 0 016 0z"></path></svg>
<span class="text-xs mt-1">Nodes</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition"
data-target="ethics-screen"> <svg class="w-6 h-6" fill="none"
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M9 12l2 2
4-4m5.618-4.016A11.955 11.955 0 0112 2.944a11.955 11.955 0 01-8.618
3.04A12.02 12.02 0 003 9c0 5.591 3.824 10.29 9 11.622 5.176-1.332
9-6.03 9-11.622 0-1.042-.133-2.052-.382-3.016z"></path></svg> <span
class="text-xs mt-1">Ethics</span> </button>
             <button class="nav-button flex flex-col items-center
text-gray-400 hover:text-neon-cyan transition active"
data-target="settings-screen"> <svg class="w-6 h-6" fill="none"
```

```
stroke="currentColor" viewBox="0 0 24 24"><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M10.325 4.317c.426-1.756
2.924-1.756 3.35 0a1.724 1.724 0 002.573 1.066c1.543-.94 3.31.826 2.37
2.37a1.724 1.724 0 001.065 2.572c1.756.426 1.756 2.924 0 3.35a1.724
1.724 0 00-1.066 2.573c.94 1.543-.826 3.31-2.37 2.37a1.724 1.724 0
00-2.572 1.065c-.426 1.756-2.924 1.756-3.35 0a1.724 1.724 0
00-2.573-1.066c-1.543.94-3.31-.826-2.37-2.37a1.724 1.724 0
00-1.065-2.572c-1.756-.426-1.756-2.924 0-3.35a1.724 1.724 0
001.066-2.573c-.94-1.543.826-3.31 2.37-2.37.996.608 2.296.07
2.572-1.065z"></path><path stroke-linecap="round"
stroke-linejoin="round" stroke-width="2" d="M15 12a3 3 0 11-6 0 3 3 0
016 0z"></path></svg> <span class="text-xs mt-1">Settings</span>
</button>
        </nav>
    </div>


    <script>
        // --- Navigation Logic ---
        let currentScreen = 'splash-screen'; // Start with splash
screen
        let navigationHistory = []; // Simple history for back button

        // Function to navigate between screens
        function navigateTo(targetScreenId, fromScreenId = null, data
= null) {
            const currentScreenElement =
document.getElementById(currentScreen);
            const targetScreenElement =
document.getElementById(targetScreenId);

            if (!targetScreenElement) {
                console.error(`Screen with ID ${targetScreenId} not
found.`);
                return;
            }

            // Add to history if navigating forward from a known
screen
            if (fromScreenId && fromScreenId !== targetScreenId) {
                 // Avoid pushing duplicates if spam clicking
back/forth
                if (navigationHistory.length === 0 ||
navigationHistory[navigationHistory.length - 1] !== fromScreenId) {
                    navigationHistory.push(fromScreenId);
                }
            } else if (targetScreenId === 'login-screen' &&
currentScreen === 'splash-screen') {
```

```
                 // Special case: Splash to Login doesn't add splash
to history
                 navigationHistory = []; // Clear history after splash
            } else if (targetScreenId === 'dashboard-screen' &&
currentScreen === 'login-screen') {
                 // Special case: Login to Dashboard clears history
                 navigationHistory = [];
            }


            console.log(`Navigating from ${currentScreen} to
${targetScreenId}. History:`, navigationHistory);

            // Hide current screen
            if (currentScreenElement) {
                currentScreenElement.classList.remove('active');
                // Use setTimeout to allow fade-out transition before
hiding completely
                 setTimeout(() => {
                     // Ensure it's still the screen we intended to
hide
                     if (currentScreenElement.id === currentScreen) {
                        // currentScreenElement.style.display =
'none'; // Might cause flicker, opacity handles visibility
                     }
                 }, 500); // Match CSS transition duration
            }

            // Show target screen
            targetScreenElement.classList.add('active');
            currentScreen = targetScreenId;

            // Update active state for bottom navigation
            updateBottomNav(targetScreenId);

            // Handle specific screen initializations or data passing
            if (targetScreenId === 'node-control-screen' && data) {

document.getElementById('node-control-title').textContent = `Control:
${data}`;
                // Reset collapsible panels (optional)
                resetCollapsibles('#node-control-screen');
            }
             if (targetScreenId === 'ethics-screen') {
                 // Update gauge value dynamically (example)
                 updateGauge('ethics-screen', 94);
             }
             if (targetScreenId === 'dashboard-screen') {
                 updateGauge('dashboard-screen', 92); // Update
dashboard gauge too
             }
        }

        // Function to handle the back button logic
        function goBack() {
            if (navigationHistory.length > 0) {
                const previousScreenId = navigationHistory.pop();
                console.log(`Going back to ${previousScreenId}.
History:`, navigationHistory);
                // Navigate without adding to history
                 const currentScreenElement =
document.getElementById(currentScreen);
                 const targetScreenElement =
document.getElementById(previousScreenId);

                 if (currentScreenElement) {
                     currentScreenElement.classList.remove('active');
                 }
                 if (targetScreenElement) {
                     targetScreenElement.classList.add('active');
                     currentScreen = previousScreenId;
                     updateBottomNav(currentScreen);
                      // Update gauges if navigating back to relevant
screens
                     if (previousScreenId === 'ethics-screen')
updateGauge('ethics-screen', 94);
                     if (previousScreenId === 'dashboard-screen')
updateGauge('dashboard-screen', 92);
                 }

            } else {
                // Default back action if history is empty (e.g., go
to dashboard)
                console.log("History empty, navigating to
dashboard.");
                navigateTo('dashboard-screen');
            }
        }

        // Function to update the active state of bottom navigation
buttons
        function updateBottomNav(activeScreenId) {
            const navContainers =
document.querySelectorAll('.bottom-nav'); // Select all nav bars
            navContainers.forEach(nav => {
                const buttons = nav.querySelectorAll('.nav-button');
                buttons.forEach(button => {
                    if (button.dataset.target === activeScreenId) {
                        button.classList.add('active');
                    } else {
                        button.classList.remove('active');
                    }
                });
            });
        }

        // --- Specific Screen Logic ---

        // Splash Screen Timeout (optional auto-navigation)
        // setTimeout(() => {
        //     if (currentScreen === 'splash-screen') {
        //         navigateTo('login-screen');
        //     }
        // }, 4000); // Auto-navigate after 4 seconds if not clicked

        // Login Function (Fake)
        function login() {
            const username =
document.getElementById('username').value;
            const password =
document.getElementById('password').value;
            const errorElement =
document.getElementById('login-error');

            errorElement.textContent = ''; // Clear previous errors

            // Basic validation (replace with real logic)
            if (username && password) {
                 // Simulate successful login
                 console.log('Login attempt successful (simulation)');
                 navigateTo('dashboard-screen', 'login-screen');
            } else {
                 // Simulate failed login
                 console.log('Login attempt failed (simulation)');
                 errorElement.textContent = 'Invalid credentials
(simulation)';
            }
        }

        // Collapsible Panel Logic
        function toggleCollapse(triggerElement, contentId) {
            const contentElement = document.getElementById(contentId);
            if (contentElement) {
                triggerElement.classList.toggle('open');
                contentElement.classList.toggle('open');
            }
        }

        // Reset collapsibles within a container
        function resetCollapsibles(containerSelector) {
             const container =
document.querySelector(containerSelector);
             if (container) {

container.querySelectorAll('.collapsible-trigger').forEach(el =>
el.classList.remove('open'));

container.querySelectorAll('.collapsible-content').forEach(el =>
el.classList.remove('open'));
                 // Optionally open default ones
                 const defaultOpenTrigger =
container.querySelector('#node-status-content')?.previousElementSiblin
g;
                 const defaultOpenContent =
container.querySelector('#node-status-content');
                 if(defaultOpenTrigger && defaultOpenContent) {
                     defaultOpenTrigger.classList.add('open');
                     defaultOpenContent.classList.add('open');
                 }
             }
        }

         // Gauge Update Logic
        function updateGauge(screenId, value) {
            const screenElement = document.getElementById(screenId);
            if (!screenElement) return;

            const gaugeFill =
screenElement.querySelector('.gauge-fill');
            const gaugeValue =
screenElement.querySelector('.gauge-value');

            if (gaugeFill && gaugeValue) {
                gaugeFill.style.setProperty('--value', value);
                gaugeValue.textContent = `${value}%`;
            }
        }


        // --- Event Listeners ---

        // Add event listeners to all bottom navigation buttons
        document.querySelectorAll('.nav-button').forEach(button => {
            button.addEventListener('click', () => {
                const targetScreen = button.dataset.target;
                if (targetScreen && targetScreen !== currentScreen) {
                     // Determine the 'from' screen based on the
current active screen
                     // This assumes the button click always
originates from the currently displayed screen
                    navigateTo(targetScreen, currentScreen);
                }
            });
        });

        // Initial setup
        document.addEventListener('DOMContentLoaded', () => {
            // Ensure only the initial screen is active
            document.querySelectorAll('.screen').forEach(screen => {
                if (screen.id !== currentScreen) {
                    screen.classList.remove('active');
                } else {
                     screen.classList.add('active'); // Ensure initial
screen is visible
                }
            });
            updateBottomNav(currentScreen); // Set initial nav state
(though none might be active on splash)
        });

    </script>

</body>
</html>
```