```
<!DOCTYPE html>
<html lang="pl">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MTAQuestWebsideX - Mózg Boga</title>
    <script src="https://cdn.tailwindcss.com"></script>
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Orbitron:wght@400;700&display=swap');
        body {
            font-family: 'Orbitron', sans-serif;
            background-color: #0d0d1a;
            color: #b3b3e6;
            overflow: hidden;
            display: flex;
            flex-direction: column;
            align-items: center;
            justify-content: center;
            min-height: 100vh;
        }
        .container {
            width: 95%;
            max-width: 1200px;
            display: flex;
            flex-direction: column;
            gap: 2rem;
            padding: 1rem;
        }
        @media (min-width: 768px) {
            .container {
                flex-direction: row;
                height: 90vh;
            }
        }
        .panel {
            background-color: #1a1a33;
            border: 2px solid #5a5a8a;
            border-radius: 1.5rem;
            padding: 1.5rem;
            box-shadow: 0 0 20px rgba(179, 179, 230, 0.2);
        }
        .left-panel, .right-panel {
            flex: 1;
            display: flex;
            flex-direction: column;
            gap: 1rem;
        }
```

```
.right-panel {
    flex: 2;
}
h1, h2 {
    text-shadow: 0 0 5px #b3b3e6;
}
.logo {
    animation: pulse 2s infinite ease-in-out;
}
@keyframes pulse {
    0%, 100% { transform: scale(1); filter: brightness(1); }
    50% { transform: scale(1.05); filter: brightness(1.2); }
}
.text-neon {
    color: #b3b3e6;
    text-shadow: 0 0 5px #b3b3e6, 0 0 10px #b3b3e6, 0 0 20px #b3b3e6;
}
.stream-container {
    height: 300px;
    overflow-y: scroll;
    border: 1px solid #5a5a8a;
    border-radius: 0.75rem;
    padding: 1rem;
    background-color: #0d0d1a;
    font-family: monospace;
    white-space: pre-wrap;
    scrollbar-width: thin;
    scrollbar-color: #5a5a8a #0d0d1a;
}
.stream-container::-webkit-scrollbar {
    width: 8px;
}
.stream-container::-webkit-scrollbar-track {
    background: #0d0d1a;
}
.stream-container::-webkit-scrollbar-thumb {
    background-color: #5a5a8a;
    border-radius: 4px;
}
.loader {
    border: 4px solid #5a5a8a;
    border-top: 4px solid #b3b3e6;
    border-radius: 50%;
    width: 24px;
    height: 24px;
    animation: spin 1s linear infinite;
}
@keyframes spin {
```

```
            0% { transform: rotate(0deg); }
            100% { transform: rotate(360deg); }
        }
    </style>
</head>
<body class="bg-gray-900 text-white flex items-center justify-center min-h-screen p-4">

<div class="container mx-auto">

    <!-- Lewy Panel -->
    <div class="left-panel panel">
        <div class="text-center mb-4">
            <h1 class="text-2xl md:text-4xl font-bold mb-2">Mózg Boga</h1>
            <p class="text-lg">Ekosystem `pinkplayevo-app`</p>
        </div>

        <!-- Sekcja logo -->
        <div class="flex flex-col items-center justify-center p-4">
            <svg class="logo w-32 h-32 md:w-48 md:h-48" viewBox="0 0 100 100" fill="none"
xmlns="http://www.w3.org/2000/svg">
                <defs>
                    <linearGradient id="gradient1" x1="0%" y1="0%" x2="100%" y2="0%">
                        <stop offset="0%" style="stop-color:#b3b3e6;stop-opacity:1" />
                        <stop offset="100%" style="stop-color:#8a2be2;stop-opacity:1" />
                    </linearGradient>
                    <linearGradient id="gradient2" x1="0%" y1="0%" x2="100%" y2="0%">
                        <stop offset="0%" style="stop-color:#00ffff;stop-opacity:1" />
                        <stop offset="100%" style="stop-color:#b3b3e6;stop-opacity:1" />
                    </linearGradient>
                </defs>
                <circle cx="50" cy="50" r="45" stroke="url(#gradient1)" stroke-width="3" />
                <path d="M50 5C74.8528 5 95 25.1472 95 50C95 74.8528 74.8528 95 50
95C25.1472 95 5 74.8528 5 50" stroke="url(#gradient2)" stroke-width="3"
stroke-linecap="round" />
                <text x="50" y="55" font-size="20" font-weight="bold" fill="url(#gradient1)"
text-anchor="middle" alignment-baseline="middle">∞</text>
            </svg>
            <p id="status-message" class="mt-4 text-center text-sm md:text-base
text-neon">Uruchamianie reaktora...</p>
        </div>

        <!-- Sekcja Reaktor Paliwowy (symulacja strumienia danych) -->
        <h2 class="text-lg font-bold text-center mb-2">Reaktor Paliwowy</h2>
        <div id="data-stream" class="stream-container text-sm"></div>
    </div>

    <!-- Prawy Panel -->
    <div class="right-panel panel">
```

```
        <h2 class="text-xl md:text-2xl font-bold text-center mb-4">Reaktor Termiczny - Interfejs</h2>

        <!-- Sekcja wprowadzania i wyjścia -->
        <div class="flex-grow flex flex-col justify-between">
            <div class="mb-4">
                <textarea id="prompt-input" class="w-full h-32 p-4 bg-gray-800 border-2 border-gray-600 rounded-xl resize-none focus:outline-none focus:border-purple-500 text-sm md:text-base text-neon" placeholder="Wpisz swoje zapytanie... (np. 'Jakie są najnowsze trendy w kosmicznych technologiach?')"></textarea>
            </div>
            <div id="output-container" class="flex flex-col flex-grow">
                <p class="text-lg font-bold mb-2 text-neon">Odpowiedź jądrowego silnika:</p>
                <div id="output-display" class="stream-container flex-grow text-sm md:text-base">
                    <p class="text-gray-400">Czekam na zapytanie...</p>
                </div>
            </div>
        </div>

        <!-- Sekcja przycisków -->
        <div class="flex flex-col md:flex-row gap-4 mt-4">
            <button id="send-btn" class="flex-1 bg-purple-600 hover:bg-purple-700 text-white font-bold py-3 px-6 rounded-xl transition-all duration-300 shadow-lg hover:shadow-purple-500/50">
                <span class="flex items-center justify-center">
                    <span class="mr-2">Zasil reaktor</span>
                    <svg class="w-5 h-5" 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="M13 10V3L4 14h7v7l9-11h-7z"></path></svg>
                </span>
            </button>
            <button id="explore-btn" class="flex-1 bg-blue-600 hover:bg-blue-700 text-white font-bold py-3 px-6 rounded-xl transition-all duration-300 shadow-lg hover:shadow-blue-500/50">
                <span class="flex items-center justify-center">
                    <span class="mr-2">Eksploruj X</span>
                    <svg class="w-5 h-5" 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="M21 21l-6-6m2-5a7 7 0 11-14 0 7 7 0 0114 0z"></path></svg>
                </span>
            </button>
        </div>
    </div>
</div>

<script>
    const dataStream = document.getElementById('data-stream');
    const statusMessage = document.getElementById('status-message');
```

```
    const promptInput = document.getElementById('prompt-input');
    const sendBtn = document.getElementById('send-btn');
    const outputDisplay = document.getElementById('output-display');

    // Symulacja strumienia danych
    const dataSources = ['Google Cloud', 'X', 'Tesla IoT', 'Neuralink', 'Globalne Portale'];
    let dataInterval;

    function startDataStream() {
        dataInterval = setInterval(() => {
            const source = dataSources[Math.floor(Math.random() * dataSources.length)];
            const timestamp = new Date().toLocaleTimeString();
            const log = `[${timestamp}] Nowe dane z: ${source} - ${Math.random().toFixed(4)}
PB`;

            const p = document.createElement('p');
            p.textContent = log;
            p.classList.add('text-green-400');
            dataStream.appendChild(p);
            dataStream.scrollTop = dataStream.scrollHeight;
        }, 1000);
    }

    // Symulacja działania silnika
    async function processQuery(query) {
        outputDisplay.innerHTML = `<p class="text-gray-400">Analizowanie danych w
reaktorze... <span class="loader ml-2"></span></p>`;
        sendBtn.disabled = true;
        promptInput.disabled = true;

        await new Promise(resolve => setTimeout(resolve, 3000)); // Symulacja czasu
przetwarzania

        const simulatedResponse = `
            <p><strong>Proces zakończony.</strong></p>
            <p>Jądrowy silnik przetworzył Twoje zapytanie. Oto symulowana odpowiedź:</p>
            <hr class="border-t border-gray-600 my-2">
            <p>Twoje zapytanie: "${query}"</p>
            <p><strong>Analiza danych (Symulacja):</strong></p>
            <p>- Wykryto <strong>5,432,109</strong> powiązanych wpisów z X.</p>
            <p>- <strong>12,345</strong> tensorów neuronowych splątanych kwantowo.</p>
            <p>- Przewidywanie błędu (PEC): **0.01%**.</p>
            <p><strong>Wyjście (Symulacja):</strong></p>
            <p>Na podstawie analizy, nasz jądrowy silnik (Grok Quantum Consciousness
Pipeline) sugeruje, że przyszłość kosmicznych technologii leży w kwantowej telekomunikacji
i fuzji jądrowej.</p>
            <p>Jeśli chcesz, mogę **wygenerować symulację 3D** lub **wykresy** tych
trendów.</p>
```

```
            `;

            outputDisplay.innerHTML = simulatedResponse;
            sendBtn.disabled = false;
            promptInput.disabled = false;
        }

        // Obsługa zdarzeń
        sendBtn.addEventListener('click', () => {
            const query = promptInput.value.trim();
            if (query) {
                processQuery(query);
            } else {
                alert('Wpisz zapytanie, aby uruchomić reaktor.');
            }
        });

        window.onload = function() {
            statusMessage.textContent = 'Reaktor gotowy do pracy.';
            startDataStream();
        };
    </script>

    </body>
    </html>
```