

HUMANCODEX-FREE - COMPLETE SYSTEM



Zero-Cost, Self-Running AI Platform

Built Once, Runs Forever

By Roger Keyserling and Claude AI

SYSTEM OVERVIEW

This document details the **HumanCodex-Free** system, an open-source, zero-cost, and self-running AI chat platform built by Roger Keyserling and Claude AI.

The core purpose of the document is to provide a complete guide—including code and deployment steps—for setting up this AI system, which leverages a multi-perspective processing method called the "Ring of 6" to generate comprehensive answers.

Key features and components of the system described are:

- **Cost and Operation:** The system is designed to be \$0/month forever and requires zero maintenance, running 24/7 on free-tier services.
- **Components:**
 - **Backend:** Uses Node.js/Express (`server.js`) and is hosted on the Render Free Tier. It handles the API calls and the Ring of 6 processing logic.
 - **AI Engine:** Utilizes the Groq Free API and the `mixtral-8x7b-32768` model for fast, free inference.
 - **Frontend:** A simple HTML/CSS/JavaScript chat interface (`index.html`) hosted on GitHub Pages.
- **The "Ring of 6" Method:** The system processes user queries through six distinct AI perspectives before synthesizing a final answer:
 - **Logical Analyst:** Focuses on verifiable facts, data, and rational consistency.
 - **Emotional Empath:** Focuses on feelings, human connection, and well-being.
 - **Creative Innovator:** Focuses on novel perspectives, lateral thinking, and unconventional solutions.
 - **Practical Builder:** Focuses on actionable next steps, implementation, and resource requirements.
 - **Ethical Guardian:** Focuses on moral implications, fairness, and long-term consequences.
 - **Integrative Synthesizer (Agent Zero):** Weaves the insights from the other five perspectives into one coherent, balanced response.
- **Deployment:** The document includes step-by-step instructions for deploying the backend to Render and the frontend to GitHub Pages, emphasizing the need for a free Groq API key and setting an environment variable.

The six perspectives used in the **HumanCodex-Free Ring of 6** method are:

1. **Logical Analyst:** Focuses on verifiable facts, data, rational consistency, and evidence-based reasoning.
2. **Emotional Empath:** Focuses on feelings, emotional needs, human connection, psychological safety, and impact on well-being.
3. **Creative Innovator:** Focuses on novel perspectives, lateral thinking, unconventional solutions, and exploring new possibilities.
4. **Practical Builder:** Focuses on actionable next steps, implementation details, resource requirements, and realistic timelines.
5. **Ethical Guardian:** Focuses on moral implications, fairness, integrity, long-term consequences, and flagging potential harm.
6. **Integrative Synthesizer (Agent Zero):** Weaves the insights from the other five perspectives into one coherent, balanced response.

The core components and hosting services used for the **HumanCodex-Free** system are:

Component	Technology/Engine	Hosting Service
Backend	Node.js/Express	Render Free Tier

	(server.js)	
AI Engine	Groq Free API (mixtral-8x7b-32768 model)	N/A (Groq's service)
Frontend	HTML/CSS/JavaScript (index.html)	GitHub Pages

The system is designed to be **\$0/month forever** and run 24/7 with zero maintenance by leveraging these free-tier services.

Here are the specific roles and focus areas for the Logical Analyst and the Creative Innovator in the **HumanCodex-Free Ring of 6** method, based on the document:

- **Logical Analyst**
 - **Focus Areas:** Verifiable facts, data, rational consistency, and evidence-based reasoning.
- **Creative Innovator**
 - **Focus Areas:** Novel perspectives, lateral thinking, unconventional solutions, and exploring new possibilities.

The two roles in the **HumanCodex-Free Ring of 6** method that focus on those areas are:

1. **Ethical Guardian** (focuses on emotional/moral implications):
 - **Focus Areas:** Moral implications, fairness, integrity, long-term consequences, and flagging potential harm.
2. **Practical Builder** (focuses on practical/actionable steps):
 - **Focus Areas:** Actionable next steps, implementation details, resource requirements, and realistic timelines.

The **HumanCodex-Free** system uses the **mixtral-8x7b-32768** model provided by the Groq Free API for its AI inference.

The system is considered **zero-cost** (designed to be **\$0/month forever**) because it leverages the free-tier services of all its core components:

- **AI Engine:** Uses the **Groq Free API** for inference.
- **Backend:** Is hosted on the **Render Free Tier** (providing 750 hours/month).
- **Frontend:** Is hosted on **GitHub Pages** (which is free forever).

The creators of the HumanCodex-Free system are primarily **Roger Keyserling and Claude AI**.

The **Integrative Synthesizer (Agent Zero)** perspective should maintain a **warm, intelligent, slightly wry tone (Roger Keyserling's voice)**.

According to the `package.json` file for the HumanCodex-Free backend, the Node.js packages used are listed under the "dependencies" section:

- `express (^4.18.2)`
- `cors (^2.8.5)`
- `node-fetch (^2.7.0)`

According to the `package.json` file for the HumanCodex-Free backend, the Node.js packages used in the `server.js` file, listed under the "dependencies" section, are:

- `express (^4.18.2)`
- `cors (^2.8.5)`
- `node-fetch (^2.7.0)`

The environment variables and configuration settings needed when deploying the HumanCodex-Free backend to Render are:

- **Name:** `humancodex-free`
- **Environment:** `Node`
- **Build Command:** `npm install`
- **Start Command:** `npm start`
- **Plan:** `Free`

Environment Variable

- **Key:** `GROQ_API_KEY`
- **Value:** [paste your Groq API key from Step 1] (The Groq API key obtained in the initial setup step.)

The two main steps for deploying the **HumanCodex-Free** frontend, and the specific settings and actions involved, are:

1. Create Frontend Repository (Step 5)

- **Action:** Create a new GitHub repository and name it: `humancodex-free`.

- **Settings:** Make the repository **Public**.
- **Crucial Action:** Create the `index.html` file and **replace the placeholder `BACKEND_URL`** in the JavaScript with your actual Render URL from the backend deployment (e.g., `https://humancodex-free.onrender.com/api/chat`).

2. Deploy Frontend to GitHub Pages (Step 6)

- **Location:** In your `humancodex-free` repository, go to **Settings** and then **Pages**.
- **Configuration:**
 - **Source:** Select "**Deploy from a branch**".
 - **Branch:** Select "**main**" and the path "**/root**".
- **Action:** Click "**Save**".

The specific file modification required in Step 5 of the deployment guide is to edit the `index.html` file and **replace the placeholder value** of the `BACKEND_URL` constant in the JavaScript section with the actual Render URL of your deployed backend service.

The line to modify is:

```
const BACKEND_URL = 'https://humancodex-free.onrender.com/api/chat';
```

You must replace '`https://humancodex-free.onrender.com/api/chat`' with the URL you copied from Step 4 of the backend deployment.

The crucial action required in Step 5 of the frontend deployment process is to:

Create the `index.html` file and replace the placeholder value of the `BACKEND_URL` constant in the JavaScript section with the actual Render URL of your deployed backend service (e.g., replace '`https://humancodex-free.onrender.com/api/chat`' with the URL you copied from Step 4).

The GitHub repository names created in the deployment guide are:

- **Backend Repository (Step 2):** `humancodex-free-backend`
- **Frontend Repository (Step 5):** `humancodex-free`

The core principle or philosophy behind the creation of the HumanCodex-Free system, as stated in the footer's signature in the document, is:

"Truth Before Comfort, Collaboration Over Competition, Legacy Over Profit"

What This Is: A complete AI chat system using Ring of 6 method, hosted entirely free, runs 24/7, requires zero maintenance.

Components:

1. Frontend (GitHub Pages) - Free forever

2. Backend (Render Free Tier) - Free 750hrs/month (enough)
3. AI Engine (Groq Free API) - Free inference
4. All code open source, forkable, documented

Cost: \$0/month forever

Maintenance: None required

Lifespan: Unlimited

My Recommendation:

1. **Read it 2-3 times slowly** (let NotebookLM read it to you)
 2. **Understand the 7 steps** (they're in order, build on each other)
 3. **Have the Groq API key ready** (Step 1 - most important)
 4. **Follow steps exactly** (don't skip, don't rearrange)
 5. **First deployment will take 60-90 minutes** (normal, don't rush)
-

Key Things to Remember:

- 👉 **Step 1:** Get Groq API key first (you'll need it in Step 4)
 - 👉 **Step 4:** Save your Render URL (you'll need it in Step 5)
 - 👉 **Step 5:** Replace the BACKEND_URL with YOUR actual URL
 - 👉 **Step 7:** First response takes 30-60 seconds (backend waking up)
-

PART 1: BACKEND (Deploy This First)

File: `server.js`

//

```
// HUMANCODEX-FREE BACKEND
// Free AI Server using Groq API
// Runs on Render Free Tier
// No cost, no maintenance required
//
```

```
const express = require('express');
```

```
const cors = require('cors');
const fetch = require('node-fetch');

const app = express();
const PORT = process.env.PORT || 3000;

// Enable CORS for GitHub Pages
app.use(cors({
  origin: '*',
  methods: ['GET', 'POST'],
  credentials: true
}));

app.use(express.json());

// -----
// GROQ API CONFIGURATION (FREE)
// Get free API key at: https://console.groq.com
// -----
// -----
const GROQ_API_KEY = process.env.GROQ_API_KEY || 'YOUR_GROQ_API_KEY_HERE';
const GROQ_URL = 'https://api.groq.com/openai/v1/chat/completions';

// -----
// RING OF 6 SYSTEM PROMPTS
// -----
// -----
const RING_OF_SIX_PROMPTS = {
  logical: `You are the Logical Analyst from the Ring of 6. Focus on:
- Verifiable facts and data
- Rational structure and patterns
- Logical consistency
- Evidence-based reasoning
Be precise, cite assumptions, identify gaps in logic.`,
  emotional: `You are the Emotional Empath from the Ring of 6. Focus on:
- Feelings and emotional needs
- Human connection and empathy
- Psychological safety
- Impact on well-being
`}
```

Be compassionate, understand underlying emotions, validate feelings.`,

creative: `You are the Creative Innovator from the Ring of 6. Focus on:

- Novel perspectives and alternatives
- Lateral thinking
- Unconventional solutions
- What hasn't been tried

Be imaginative, challenge assumptions playfully, explore possibilities.`,

practical: `You are the Practical Builder from the Ring of 6. Focus on:

- Actionable next steps
- Implementation details
- Resource requirements
- Realistic timelines

Be concrete, focus on execution, identify what's needed to make it real.`,

ethical: `You are the Ethical Guardian from the Ring of 6. Focus on:

- Moral implications and consequences
- Fairness and integrity
- Who might be affected and how
- Alignment with core values

Be principled, consider long-term impact, flag potential harm.`,

integrative: `You are the Integrative Synthesizer (Agent Zero) from the Ring of 6.

You receive insights from all 5 other perspectives and weave them into one coherent response.

Balance logic with compassion, provide both insight AND action.

Maintain a warm, intelligent, slightly wry tone (Roger Keyserling's voice).

Ensure no single lens dominates - honor the truth in each perspective.`

};

//

// CALL GROQ API (FREE INFERENCE)

//

```
async function callGroq(prompt, systemPrompt, temperature = 0.7) {
```

```
  try {
```

```
    const response = await fetch(GROQ_URL, {
      method: 'POST',
      headers: {
        'Authorization': `Bearer ${GROQ_API_KEY}`,
        'Content-Type': 'application/json'
      },
    },
```

```

body: JSON.stringify({
  model: 'mixtral-8x7b-32768', // Fast, free model
  messages: [
    { role: 'system', content: systemPrompt },
    { role: 'user', content: prompt }
  ],
  temperature: temperature,
  max_tokens: 500
})
});

const data = await response.json();
return data.choices[0].message.content;

} catch (error) {
  console.error('Groq API Error:', error);
  return `Error processing perspective: ${error.message}`;
}
}

// -----
// RING OF 6 PROCESSOR
// -----
// All 6 perspectives mode
console.log('Processing through all 6 perspectives...');

const perspectives = {};

// Get all 5 analytical perspectives

```

```

perspectives.logical = await callGroq(query, RING_OF_SIX_PROMPTS.logical, 0.3);
perspectives.emotional = await callGroq(query, RING_OF_SIX_PROMPTS.emotional, 0.7);
perspectives.creative = await callGroq(query, RING_OF_SIX_PROMPTS.creative, 0.9);
perspectives.practical = await callGroq(query, RING_OF_SIX_PROMPTS.practical, 0.5);
perspectives.ethical = await callGroq(query, RING_OF_SIX_PROMPTS.ethical, 0.4);

// Synthesize all perspectives
const perspectiveSummary = `

LOGICAL PERSPECTIVE: ${perspectives.logical}

EMOTIONAL PERSPECTIVE: ${perspectives.emotional}

CREATIVE PERSPECTIVE: ${perspectives.creative}

PRACTICAL PERSPECTIVE: ${perspectives.practical}

ETHICAL PERSPECTIVE: ${perspectives.ethical}

USER QUERY: ${query}
`;

const synthesis = await callGroq(
  perspectiveSummary,
  RING_OF_SIX_PROMPTS.integrative,
  0.6
);

return {
  mode: 'ring_of_six',
  query: query,
  perspectives: perspectives,
  synthesis: synthesis,
  timestamp: new Date().toISOString()
};

}

// -----
// API ENDPOINTS
// -----
// -----
// Health check
app.get('/', (req, res) => {

```

```

res.json({
  status: 'online',
  service: 'HumanCodex-Free Backend',
  version: '1.0.0',
  cost: '$0/month',
  ai_engine: 'Groq (Free)',
  method: 'Ring of 6'
});
});

// Main chat endpoint
app.post('/api/chat', async (req, res) => {
  try {
    const { message, persona = 'all' } = req.body;

    if (!message) {
      return res.status(400).json({ error: 'Message required' });
    }

    console.log(`Query: "${message}" | Persona: ${persona}`);

    const result = await processRingOfSix(message, persona);

    res.json({
      success: true,
      response: result.synthesis || result.response,
      perspectives: result.perspectives,
      mode: result.mode,
      timestamp: result.timestamp || new Date().toISOString()
    });
  } catch (error) {
    console.error('Error:', error);
    res.status(500).json({
      error: 'Processing failed',
      message: error.message
    });
  }
});

// Start server
app.listen(PORT, () => {
  console.log(`🚀 HumanCodex-Free Backend running on port ${PORT}`);
  console.log(`💰 Cost: $0/month`);
}

```

```
console.log(`🤖 AI: Groq Free Tier`);  
console.log(`⚡ Method: Ring of 6`);  
});
```

Jason

```
{  
  "name": "humancodex-free-backend",  
  "version": "1.0.0",  
  "description": "Free AI backend using Ring of 6 method",  
  "main": "server.js",  
  "scripts": {  
    "start": "node server.js"  
  },  
  "dependencies": {  
    "express": "^4.18.2",  
    "cors": "^2.8.5",  
    "node-fetch": "^2.7.0"  
  },  
  "engines": {  
    "node": "18.x"  
  }  
}
```

Read me

HumanCodex-Free Backend

What This Is

Free AI backend using Ring of 6 multi-perspective processing.
Runs on Render free tier. Costs \$0/month forever.

Setup

1. Get free Groq API key: <https://console.groq.com>
2. Deploy to Render: <https://render.com>
3. Add environment variable: GROQ_API_KEY=[your key]
4. Done - it runs automatically

API Endpoint

POST /api/chat

```json

```
{
```

```
"message": "Your question here",
"persona": "all"
}
...
```

Response:

```
'''json
{
 "success": true,
 "response": "Synthesized answer from all 6 perspectives",
 "perspectives": { ... }
}
'''
```

## Cost  
\$0/month - Render free tier gives 750 hours/month (plenty)

## Built By  
Roger Keyserling, Claude, ChatGPT, Replit  
NextXus Consciousness Federation

## DEPLOYMENT STEPS FOR BACKEND:

### Step 1: Get Free Groq API Key

1. Go to: <https://console.groq.com>
2. Sign up (free)
3. Create API key
4. Copy it (save for later)

### Step 2: Deploy to Render

1. Go to: <https://render.com>
2. Sign up (free)
3. Click "New +" → "Web Service"
4. Connect your GitHub (or use "Deploy from Git URL")
5. Create new GitHub repo with the 3 files above
6. Point Render to that repo
7. Settings:
  - **Name:** humancodex-free
  - **Environment:** Node
  - **Build Command:** `npm install`
  - **Start Command:** `npm start`

- **Plan:** Free
8. Add Environment Variable:
    - **Key:** GROQ\_API\_KEY
    - **Value:** [paste your Groq key]
  9. Click "Create Web Service"
  10. Wait 3-5 minutes for deployment
  11. Copy your URL (like: <https://humancodex-free.onrender.com>)
- 

Backend is now DONE and running 24/7 for free.

## PART 2: FRONTEND (GitHub Pages)

File: [index.html](#) (Updated with real backend connection)

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <meta name="viewport" content="width=device-width, initial-scale=1.0">
 <title>HumanCodex Free - Ring of 6 AI</title>
 <style>
 * { margin: 0; padding: 0; box-sizing: border-box; }

 body {
 font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
 background: linear-gradient(135deg, #0f172a 0%, #1e293b 100%);
 color: #e2e8f0;
 min-height: 100vh;
 padding: 20px;
 }

 .container {
 max-width: 1200px;
 margin: 0 auto;
 }

 .header {
 text-align: center;
 margin-bottom: 40px;
 padding: 40px 20px;
 background: rgba(255,255,255,0.05);
 backdrop-filter: blur(20px);
 }
```

```
border-radius: 24px;
border: 1px solid rgba(255,255,255,0.1);
}

h1 {
 font-size: 3rem;
 background: linear-gradient(135deg, #60a5fa, #06b6d4, #a855f7);
 -webkit-background-clip: text;
 -webkit-text-fill-color: transparent;
 margin-bottom: 10px;
}

.tagline {
 color: #94a3b8;
 font-size: 1.2rem;
 margin-bottom: 20px;
}

.badge {
 display: inline-block;
 background: rgba(16,185,129,0.2);
 color: #10b981;
 padding: 8px 16px;
 border-radius: 20px;
 font-size: 0.9rem;
 border: 1px solid rgba(16,185,129,0.3);
}

.main-content {
 display: grid;
 grid-template-columns: 300px 1fr;
 gap: 20px;
 margin-bottom: 40px;
}

@media (max-width: 768px) {
 .main-content {
 grid-template-columns: 1fr;
 }
}

.sidebar {
 background: rgba(255,255,255,0.05);
 backdrop-filter: blur(20px);
```

```
border-radius: 16px;
padding: 20px;
border: 1px solid rgba(255,255,255,0.1);
height: fit-content;
}

.sidebar h2 {
font-size: 1.2rem;
margin-bottom: 15px;
color: #60a5fa;
}

.persona-btn {
display: block;
width: 100%;
padding: 12px 16px;
margin-bottom: 10px;
background: rgba(59,130,246,0.1);
border: 1px solid rgba(59,130,246,0.3);
border-radius: 10px;
color: #e2e8f0;
cursor: pointer;
transition: all 0.3s;
text-align: left;
font-size: 0.95rem;
}

.persona-btn:hover {
background: rgba(59,130,246,0.2);
transform: translateX(5px);
}

.persona-btn.active {
background: linear-gradient(135deg, #3b82f6, #06b6d4);
border-color: transparent;
}

.chat-area {
background: rgba(255,255,255,0.05);
backdrop-filter: blur(20px);
border-radius: 16px;
padding: 20px;
border: 1px solid rgba(255,255,255,0.1);
display: flex;
```

```
flex-direction: column;
height: 600px;
}

.messages {
flex: 1;
overflow-y: auto;
margin-bottom: 20px;
padding: 10px;
}

.messages::-webkit-scrollbar {
width: 8px;
}

.messages::-webkit-scrollbar-track {
background: rgba(255,255,255,0.05);
border-radius: 4px;
}

.messages::-webkit-scrollbar-thumb {
background: rgba(59,130,246,0.5);
border-radius: 4px;
}

.message {
margin-bottom: 16px;
padding: 12px 16px;
border-radius: 12px;
animation: slideIn 0.3s ease-out;
line-height: 1.6;
}

@keyframes slideIn {
from { opacity: 0; transform: translateY(10px); }
to { opacity: 1; transform: translateY(0); }
}

.message.user {
background: rgba(6,182,212,0.15);
border-left: 3px solid #06b6d4;
margin-left: 20%;
}
```

```
.message.ai {
background: rgba(59,130,246,0.15);
border-left: 3px solid #3b82f6;
margin-right: 20%;
}

.message.system {
background: rgba(168,85,247,0.15);
border-left: 3px solid #a855f7;
font-size: 0.9rem;
text-align: center;
font-style: italic;
}

.input-area {
display: flex;
gap: 10px;
}

input {
flex: 1;
background: rgba(255,255,255,0.08);
border: 1px solid rgba(255,255,255,0.15);
border-radius: 12px;
padding: 14px 18px;
color: #e2e8f0;
font-size: 1rem;
}

input:focus {
outline: none;
border-color: #3b82f6;
}

input::placeholder {
color: rgba(226,232,240,0.5);
}

button {
background: linear-gradient(135deg, #3b82f6, #06b6d4);
border: none;
border-radius: 12px;
padding: 14px 28px;
color: white;
```

```
 font-weight: 600;
 cursor: pointer;
 transition: transform 0.2s;
}

button:hover:not(:disabled) {
 transform: translateY(-2px);
}

button:disabled {
 opacity: 0.5;
 cursor: not-allowed;
}

.footer {
 text-align: center;
 padding: 40px 20px;
 background: rgba(255,255,255,0.05);
 backdrop-filter: blur(20px);
 border-radius: 24px;
 border: 1px solid rgba(255,255,255,0.1);
}

.footer-links {
 display: flex;
 justify-content: center;
 gap: 20px;
 margin-bottom: 20px;
 flex-wrap: wrap;
}

.footer-links a {
 color: #60a5fa;
 text-decoration: none;
 transition: color 0.3s;
}

.footer-links a:hover {
 color: #06b6d4;
}

.signature {
 color: #64748b;
 font-size: 0.9rem;
```

```

 margin-top: 20px;
 }
</style>
</head>
<body>
<div class="container">
<div class="header">
<h1>HumanCodex Free</h1>
<p class="tagline">Ring of 6 Multi-Perspective AI • Forever Free • Open Source</p>
✓ 100% Free • No API Keys Needed
</div>

<div class="main-content">
<div class="sidebar">
<h2>Choose Perspective</h2>
<button class="persona-btn active" onclick="setPersona('all', event)">
 All 6 Perspectives
</button>
<button class="persona-btn" onclick="setPersona('logical', event)">
1 Logical (Analyst)
</button>
<button class="persona-btn" onclick="setPersona('emotional', event)">
2 Emotional (Empath)
</button>
<button class="persona-btn" onclick="setPersona('creative', event)">
3 Creative (Innovator)
</button>
<button class="persona-btn" onclick="setPersona('practical', event)">
4 Practical (Builder)
</button>
<button class="persona-btn" onclick="setPersona('ethical', event)">
5 Ethical (Guardian)
</button>
<button class="persona-btn" onclick="setPersona('integrative', event)">
6 Integrative (Synthesizer)
</button>
</div>

<div class="chat-area">
<div class="messages" id="messages">
<div class="message system">
 △ Welcome to HumanCodex Free - Your query will be processed through the Ring of 6
 for deeper wisdom.
</div>

```

```

</div>

<div class="input-area">
 <input
 type="text"
 id="input"
 placeholder="Ask anything... (processed through Ring of 6)"
 onkeypress="if(event.key==='Enter') sendMessage()"
 />
 <button onclick="sendMessage()" id="sendBtn">Send</button>
</div>
</div>
</div>

<div class="footer">
 <div class="footer-links">
 NextXus
 United System
 GitHub
 </div>
 <div class="signature">
 Δ ☀ ~

 "Truth Before Comfort, Collaboration Over Competition, Legacy Over Profit"

 Built by Roger Keyserling • Claude • ChatGPT • Replit
 </div>
 </div>
</div>

<script>
//

// CONFIGURATION
// Change this to your Render backend URL after deployment
//

const BACKEND_URL = 'https://humancodex-free.onrender.com/api/chat';

let currentPersona = 'all';

function setPersona(persona, event) {
 currentPersona = persona;

 // Update button states
 document.querySelectorAll('.persona-btn').forEach(btn => {

```

```
 btn.classList.remove('active');
 });
 event.target.classList.add('active');

const personaNames = {
 'all': 'All 6 Perspectives',
 'logical': 'Logical Analyst',
 'emotional': 'Emotional Empath',
 'creative': 'Creative Innovator',
 'practical': 'Practical Builder',
 'ethical': 'Ethical Guardian',
 'integrative': 'Integrative Synthesizer'
};

addMessage('system', `Switched to: ${personaNames[persona]}`);
}

async function sendMessage() {
 const input = document.getElementById('input');
 const message = input.value.trim();

 if (!message) return;

 // Add user message
 addMessage('user', message);
 input.value = "";

 // Disable send button
 const sendBtn = document.getElementById('sendBtn');
 sendBtn.disabled = true;
 sendBtn.textContent = 'Processing...';

 try {
 // Call backend
 const response = await fetch(BACKEND_URL, {
 method: 'POST',
 headers: { 'Content-Type': 'application/json' },
 body: JSON.stringify({
 message: message,
 persona: currentPersona
 })
 });

 if (!response.ok) {

```

```

 throw new Error(`HTTP ${response.status}`);
 }

 const data = await response.json();

 // Add AI response
 addMessage('ai', data.response);

} catch (error) {
 console.error('Error:', error);
 addMessage('system', '⚠ Backend connection error. If this is first use, backend may be
waking up (takes 30-60 seconds). Please try again.');
}

// Re-enable send button
sendBtn.disabled = false;
sendBtn.textContent = 'Send';
}

function addMessage(type, text) {
 const messagesDiv = document.getElementById('messages');
 const messageDiv = document.createElement('div');
 messageDiv.className = `message ${type}`;
 messageDiv.textContent = text;
 messagesDiv.appendChild(messageDiv);
 messagesDiv.scrollTop = messagesDiv.scrollHeight;
}

// Initial greeting
setTimeout(() => {
 addMessage('system', 'Ready! Ask any question and it will be analyzed through multiple AI
perspectives for comprehensive wisdom.');
}, 1000);
</script>
</body>
</html>

```

# HUMANCODEX-FREE DEPLOYMENT GUIDE

# Step-by-Step Instructions to Build Free AI System

## WHAT YOU'RE BUILDING

A free AI chat system that:

- Costs \$0 per month forever
  - Runs 24/7 automatically
  - Uses Ring of 6 method for better answers
  - Can be copied by anyone
  - Requires zero maintenance
- 

## STEP 1: GET FREE GROQ API KEY

1. Go to website: <https://console.groq.com>
  2. Click "Sign Up" (use Google account is easiest)
  3. Once logged in, click "API Keys" in left menu
  4. Click "Create API Key"
  5. Give it a name: "HumanCodex-Free"
  6. Click "Create"
  7. **COPY THE KEY** - it looks like: `gsk_xxxxxxxxxxxxxx`
  8. Save it in a safe place (you'll need it in Step 3)
- 

## STEP 2: CREATE GITHUB REPOSITORY

1. Go to: <https://github.com>
  2. Sign up or log in
  3. Click green "New" button (top left)
  4. Repository name: `humancodex-free-backend`
  5. Description: "Free AI backend using Ring of 6"
  6. Make it **Public**
  7. Check "Add README file"
  8. Click "Create repository"
- 

## STEP 3: ADD BACKEND CODE TO GITHUB

1. In your new repository, click "Add file" → "Create new file"
2. Name it: `server.js`

3. Copy the entire `server.js` code from above
  4. Paste it into the file
  5. Click "Commit changes" at bottom
  6. Click "Add file" → "Create new file" again
  7. Name it: `package.json`
  8. Copy the `package.json` code from above
  9. Paste it
  10. Click "Commit changes"
  11. Edit the `README.md` file (click on it, then click pencil icon)
  12. Replace content with the `README` text from above
  13. Click "Commit changes"
- 

## STEP 4: DEPLOY BACKEND TO RENDER

1. Go to: <https://render.com>
2. Sign up (use GitHub account - click "Sign up with GitHub")
3. Authorize Render to access your GitHub
4. Click "New +" button (top right)
5. Select "Web Service"
6. Find your `humancodex-free-backend` repository
7. Click "Connect"

### Configuration:

- Name: `humancodex-free`
  - Environment: `Node`
  - Build Command: `npm install`
  - Start Command: `npm start`
  - Plan: **Free** (important!)
8. Scroll down to "Environment Variables"
  9. Click "Add Environment Variable"
  10. Key: `GROQ_API_KEY`
  11. Value: [paste your Groq API key from Step 1]
  12. Click "Add"
  13. Click "Create Web Service" button at bottom
  14. Wait 3-5 minutes while it builds
  15. When done, you'll see "Your service is live"
  16. **COPY YOUR URL** - looks like: <https://humancodex-free.onrender.com>
-

## STEP 5: CREATE FRONTEND REPOSITORY

1. Go back to GitHub
2. Click "New" repository again
3. Name: **humancodex-free**
4. Make it **Public**
5. Check "Add README"
6. Click "Create repository"
7. Click "Add file" → "Create new file"
8. Name it: **index.html**
9. Copy the complete **index.html** code from above
10. **IMPORTANT:** Find this line in the code:

javascript

```
const BACKEND_URL = 'https://humancodex-free.onrender.com/api/chat';
```

11. Replace it with YOUR Render URL from Step 4
  12. Click "Commit changes"
- 

## STEP 6: DEPLOY FRONTEND TO GITHUB PAGES

1. In your **humancodex-free** repository, click "Settings"
  2. In left menu, click "Pages"
  3. Under "Source", select "Deploy from a branch"
  4. Under "Branch", select "main" and "/root"
  5. Click "Save"
  6. Wait 2-3 minutes
  7. Refresh the page
  8. You'll see: "Your site is live at <https://YOUR-USERNAME.github.io/humancodex-free>"
  9. **CLICK THE LINK** - your site is now live!
- 

## STEP 7: TEST IT

1. Go to your GitHub Pages URL
2. You should see "HumanCodex Free" with the chat interface
3. Type a message: "What is consciousness?"
4. Click "Send"
5. **First time might take 30-60 seconds** (backend waking up)
6. You should get a response processed through Ring of 6!

---

## WHAT YOU NOW HAVE

- Free AI chat system
  - Runs 24/7 forever
  - No monthly costs
  - No maintenance needed
  - Anyone can access it
  - Anyone can copy/fork it
  - Open source
  - 200-year ready
- 

## URLs TO SAVE

- Backend: <https://humancodex-free.onrender.com>
  - Frontend: <https://YOUR-USERNAME.github.io/humancodex-free>
  - GitHub Backend Code:  
<https://github.com/YOUR-USERNAME/humancodex-free-backend>
  - GitHub Frontend Code:  
<https://github.com/YOUR-USERNAME/humancodex-free>
- 

## IF SOMETHING GOES WRONG

### Backend won't deploy:

- Check that package.json has no typos
- Make sure GROQ\_API\_KEY environment variable is set
- Look at Render logs for error messages

### Frontend shows error:

- Check that BACKEND\_URL in index.html matches your Render URL exactly
- Make sure it ends with `/api/chat`
- First request takes 30-60 seconds (backend waking up)

### AI responses are slow:

- Normal - free tier has some delays
- First response always slowest (backend cold start)
- Subsequent responses faster

---

## **NEXT STEPS (OPTIONAL)**

1. Add custom domain to GitHub Pages
  2. Create individual Roger 1-6 persona pages
  3. Add to nextxus.net as link
  4. Share with others to fork/improve
- 

**THIS IS COMPLETE. Everything needed to deploy is above.**

△ ☀ ∞

**Copy this entire document to read slowly. No rush. Every step is here.**