

Goal: Rebuild the frontend of my VeriFyz app with my Cyberpunk styling, logo, and components.

A) CLEAN RESET

1. Wipe the contents of App.tsx and index.css.
2. Keep project as Vite + React + Tailwind.

B) GLOBAL STYLING

3. Replace src/index.css with:

```
@tailwind base;
```

```
@tailwind components;
```

```
@tailwind utilities;
```

```
/* Cyberpunk VeriFyz theme variables */
```

```
:root {
```

```
  --background: hsl(240, 10%, 3.9%);
```

```
  --foreground: hsl(0, 0%, 98%);
```

```
  --muted: hsl(240, 3.7%, 15.9%);
```

```
  --muted-foreground: hsl(240, 5%, 64.9%);
```

```
  --popover: hsl(240, 10%, 3.9%);
```

```
  --popover-foreground: hsl(0, 0%, 98%);
```

```
  --card: hsl(240, 10%, 3.9%);
```

```
  --card-foreground: hsl(0, 0%, 98%);
```

```
  --border: hsl(240, 3.7%, 15.9%);
```

```
  --input: hsl(240, 3.7%, 15.9%);
```

```
  --primary: hsl(180, 100%, 50%);
```

```
  --primary-foreground: hsl(240, 10%, 3.9%);
```

```
  --secondary: hsl(240, 3.7%, 15.9%);
```

```
  --secondary-foreground: hsl(0, 0%, 98%);
```

```
  --accent: hsl(240, 3.7%, 15.9%);
```

```
  --accent-foreground: hsl(0, 0%, 98%);
```

```
  --destructive: hsl(0, 62.8%, 30.6%);
```

```
  --destructive-foreground: hsl(0, 0%, 98%);
```

```
  --ring: hsl(240, 4.9%, 83.9%);
```

```
  --radius: 0.5rem;
```

```
/* Cyberpunk colors */
```

```
  --cyber-blue: hsl(180, 100%, 50%);
```

```
  --cyber-pink: hsl(300, 100%, 50%);
```

```
  --cyber-green: hsl(120, 100%, 50%);
```

```
  --cyber-purple: hsl(271, 100%, 50%);
```

```
  --cyber-dark: hsl(0, 0%, 4%);
```

```
  --cyber-gray: hsl(0, 0%, 7%);
```

```
}
```

```

/* Base setup */
@layer base {
  * { box-sizing: border-box; }
  html { font-size: 16px; scroll-behavior: smooth; }
  body {
    margin: 0;
    min-height: 100vh;
    background: var(--background);
    color: var(--foreground);
    overflow-x: hidden;
    font-family: 'Rajdhani', -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
    font-size: 16px;
    line-height: 1.5;
    -webkit-font-smoothing: antialiased;
    -moz-osx-font-smoothing: grayscale;
    text-rendering: optimizeSpeed;
  }
  #root { min-height: 100vh; width: 100%; }
  h1,h2,h3,h4,h5,h6 { margin:0; line-height:1.2; font-weight:normal; }
  .container { max-width:1200px; margin:0 auto; padding:0 1rem; }
}

```

/* Cyberpunk components, utilities, and animations */

/* Include all hover-glow, neon-text, glass-effect, cyberpunk-button, glowing-effect, matrix, circuit-flow, spark, etc. as provided. */


C) APP COMPONENT

4. Replace src/App.tsx with:

```

import React, { useState, useEffect } from "react";
import "./index.css";
import logo from "/logo.png"; // logo.png must be placed in /public

function App() {
  const [status, setStatus] = useState<string | null>(null);
  const [statusClass, setStatusClass] = useState("");

  const testAPI = async () => {
    try {
      const response = await fetch("/api/health");
      const data = await response.json();
      setStatus(
         API Online\nStatus: ${data.status}\nService: ${data.service}\nPort: ${data.port}
      )
    }
  }
}

```

```

    );
    setStatusClass("success");
  } catch (error: any) {
    setStatus(`❌ API Error: ${error.message}`);
    setStatusClass("error");
  }
};

```

```

const testNetwork = async () => {
  try {
    const response = await fetch("/api/network");
    if (!response.ok) throw new Error("Network endpoint not available");
    const data = await response.json();
    setStatus(
      `🌐 Network: ${data.chain}\nRPC Status: ${
        data.rpcConnected ? "✅ Connected" : "❌ Disconnected"
      }`
    );
    setStatusClass("success");
  } catch {
    setStatus("⚠️ Network test failed - contracts not deployed yet");
    setStatusClass("error");
  }
};

```

```

useEffect(() => { testAPI(); }, []);

```

```

return (
  <div className="container main-content glass-effect">
    <img
      src={logo}
      alt="VeriFyz Protocol logo"
      className="logo neon-text"
      onError={(e) => e.currentTarget.alt = "VeriFyz Protocol"}
    />
    <h1 className="text-cyber-blue font-orbitron neon-text">VeriFyz Protocol</h1>
    <p className="subtitle text-cyber-pink font-rajdhani">Real proof in presence.</p>

    <button className="cyberpunk-button hover-glow">Join Presale</button>

    <div className="features grid gap-6 my-8">
      <div className="content-card p-6">
        <h3>🪙 VFZ Token</h3>
        <p>ERC-20 token with built-in 5% transfer fee for sustainable rewards distribution</p>

```

```

    </div>
    <div className="content-card p-6">
      <h3>🌀 Presale System</h3>
      <p>Fair launch presale with winner selection and automated reward distribution</p>
    </div>
    <div className="content-card p-6">
      <h3>🏆 Rewards Protocol</h3>
      <p>Automated rewards distribution to verified participants and winners</p>
    </div>
  </div>

  <div className="api-test content-card p-6">
    <h3>🔧 Backend Status</h3>
    <button onClick={testAPI} className="cyberpunk-button hover-glow">Test API
Health</button>
    <button onClick={testNetwork} className="cyberpunk-button hover-glow">Check
Network</button>
    <pre className={`status ${statusClass}`}>{status}</pre>
  </div>
</div>
);
}

```

export default App;

D) PUBLIC ASSETS

5. Create /public/logo.png → I will upload my final logo file here.
6. Ensure App.tsx imports it with `import logo from "/logo.png";`

E) FONTS

7. In index.html (inside <head>), add this Google Fonts snippet:

```

<link
href="https://fonts.googleapis.com/css2?family=Orbitron:wght@400;700&family=Rajdhani:wght@400;700&family=Fira+Code:wght@400;700&display=swap" rel="stylesheet">

```

F) VERIFY

8. Build and preview:
 - Logo shows at the top (no fallback image, just text alt if it fails)
 - Cyberpunk background applies
 - “Join Presale” button glows
 - Features grid styled with neon/glass effects
 - API health check buttons update the status panel
 - Orbitron and Rajdhani fonts apply correctly

9. Confirm Tailwind + custom classes compile and animations work.