Cognitive Mirror Tone Logic Fix Summary

Final Fix Summary: "Frank Gate" (Tone Rendering Bug)

Even when the user selected "Frank Friend" from the dropdown, responses were displayed as if they were from "Stoic Mentor." The backend logs confirmed correct tone logic, but frontend rendering displayed the wrong persona visually.

Root Cause

1. Frontend was comparing exact string values like "Frank Friend", while the dropdown stored value="frank" and Supabase logged tone_mode inconsistently.

2. No normalization function was used to resolve different formats before rendering.

What We Fixed

Added displayTone() utility in App.js just before return():

```
const displayTone = (mode) => {
  const t = mode?.trim().toLowerCase();
  return t === 'frank' ? ' Frank Friend' : ' Stoic Mentor';
};
```

Replaced all JSX comparisons with displayTone(item.tone_mode).

Supporting Logic Confirmed

- Supabase tone_mode is stored as-is (frank, Frank Friend, stoic, etc.) - fine if normalized on render.

Cognitive Mirror Tone Logic Fix Summary

- Backend (CJS) is correctly logging forcedTone, normalizing it, and using it in the OpenAI prompt.

What We Learned

1. Normalize all tone modes before comparing.

2. Don't use display strings like 'Frank Friend' as logic tokens.

3. React function blocks must be declared before return(), not nested.

Dev Safeguards for Future Engineers

1. Always normalize tone with .trim().toLowerCase().

2. Never hardcode tone strings in JSX.

3. Use dropdown values 'frank' and 'stoic'.

4. displayTone() is the only safe way to render voice indicators.

Workflow Improvements

- Keep resurface-dev as staging and master locked.

- Keep backend logs verbose during feature merges.

- Migrate tone_mode values to lowercase if possible.

- Add code comments where normalization matters.

- Verify tone string + emoji alignment before PR close.