Marisa Peer

Jr. AI Software Engineer

Technical Challenge: The "Live" Mood Architect

Recruitment Task

Goal: Build and deploy a full-stack AI web app that generates personalized therapeutic affirmations.

Time Expectation: 4–6 Hours (includes deployment troubleshooting)
Seniority Target: New Grad / Junior

1) Context & What We’re Evaluating

This challenge is designed to assess a “builder” mindset: the ability to ship a small but complete feature end-to-end. You may use AI tools to help you code faster—what matters is that your final solution works, is deployed, and is secure.

We’re evaluating: (1) Full-stack wiring (frontend ↔ backend), (2) correct OpenAI API usage, (3) deployment competence, (4) environment variable hygiene, and (5) user-friendly error handling.

2) Stack Requirements

- Frontend: A modern JavaScript framework (React, Next.js, Vite, or Vue).
- Backend: Python using FastAPI, Flask, or Django.
- AI Integration: Must call the OpenAI API (or similar) directly (no hardcoded/mocked responses).
- Repository: All code must be hosted on a public GitHub repository.
- Deployment: The application must be live and accessible via a public URL.

You may use free tiers: Vercel/Netlify for frontend and Render/Railway/PythonAnywhere (or equivalent) for backend.

3) Product Requirements

3.1 The UI

- Clean, responsive interface.
- Inputs: user name and how they are feeling (free text or a small preset list + optional “details” field).

- Button to generate the affirmation.
- Result area that displays the generated affirmation and a loading state.

3.2 The Backend Logic

- Expose a single endpoint (suggested: POST /api/affirmation) that accepts JSON { name: string, feeling: string }.
- Construct a prompt that produces a short, empathetic, non-clinical “therapeutic affirmation”.
- Call the OpenAI API using your API key stored ONLY as an environment variable.
- Return JSON { affirmation: string } to the frontend.

3.3 Error Handling (Required)

- If the AI call fails (timeouts, network issues, API errors), the user must see a friendly message in the UI.
- No raw stack traces in the browser. No blank screens.
- Backend should return meaningful HTTP status codes (e.g., 400 for validation, 502/504 for upstream failures).
- Add basic input validation (non-empty name and feeling).

4) Security & Environment Variables (Automatic Fail Conditions)

Security Test: You must use environment variables for your OpenAI API key and any other secrets. Never commit secrets to GitHub.

Include a screenshot in your submission showing your hosting provider’s environment variable configuration (blur the secret value).

5) Deployment Requirements

- Frontend must be deployed to a public URL (e.g., Vercel or Netlify).
- Backend must be deployed to a public URL (e.g., Render/Railway/PythonAnywhere).
- Frontend must call the deployed backend (not localhost).
- CORS must be configured properly for your deployed frontend domain.
- The deployed app must be testable immediately by opening the live URL and generating an affirmation.

6) Suggested Architecture (You May Choose Your Own)

- Separate frontend + backend deployments.

7) Prompt & Safety Guidance (Required)

Your prompt should be empathetic and supportive, but must avoid medical claims or instructions. It should not diagnose conditions or provide crisis counseling.

- Include a system message that enforces:
    - No medical or legal advice; no diagnosis.
    - No self-harm guidance; if user expresses self-harm intent, respond with a safe, supportive message and encourage seeking professional help.
    - Keep responses short (2–4 sentences), warm, and specific to the user’s input.

8) README Requirements (Required)

- How to run locally (frontend and backend).
- Required environment variables (example names, no real secrets).
- How to deploy (brief steps or links to the provider setup).
- Your live URLs (frontend and backend).

9) Submission Checklist

- Public GitHub repository link.
- Live URL (the app).
- Backend URL.
- Screenshot of environment variable configuration in hosting provider (secret value blurred).
- Short note (2–5 bullets): what you would improve with more time.

Thank you-we’re excited to see what you build!