Full Stack Developer - Round One

Congratulations on your selection for this round.

This document describes the first round assignment for candidates applying to the Full Stack Developer position at BrainGnosis. The goal is to give you enough context to plan, build, and submit a small but meaningful module that uses OmniBAR.

Your challenge

Build a small web application that uses OmniBAR to evaluate at least one clear component in an agentic workflow. You decide the use case, the stack, and the interface. Keep the scope tight and ship an end to end path that a reviewer can run locally. We encourage you to be creative on how you design and present this application.

OmniBAR Repository - https://github.com/BrainGnosis/OmniBAR

Examples of suitable modules

- Prompt trace scoring. Collect prompts and variables, capture the model output, call OmniBAR to score, store the result, show a history table with filters.
- Tool call reliability. Simulate or integrate a tool call, record success or failure and latency, compute a reliability score, chart recent runs.
- Memory aggregation quality. Submit memory snippets, score for duplication and staleness, display issues and let the user mark items for cleanup.
- RAG answer check. Upload a tiny corpus, ask a question, score answer faithfulness and source coverage, store run cards. You may also design your own module. You can refer to the OmniBAR repository for ideas and available metrics.
- You may also design your own examples. Refer to the OmniBAR repository for ideas and available metrics.

Expected effort and timeline

Please submit within ten days of receiving this brief. Finishing earlier is fine. The time window is a deadline, not the time we expect you to spend. You can choose whichever example you want, pick the one that lets you show your strengths within the time box.

Required deliverables

Ship a working module that runs locally. It should call OmniBAR, store results, and display them in a simple interface.

- Code repository link, public or private invite
- README with setup steps, assumptions, and tradeoffs
- Demo video, short one under 5 mins, screen recording with voiceover

Nice to have

- A few targeted tests
- A short write up of what you cut and why you cut it

Starter kit

We will share an API key for LLM usage so you can execute benchmarking calls on OmniBAR. OmniBAR is open source and can be found at the GitHub link listed above.

Evaluation rubric

Some of the evaluation criteria used for this task is:

- Problem framing. Is the selected module clear and reasonable
- Architecture and data model. Is the flow from input to score to storage to view sensible
- Code quality and correctness. Structure, readability, error handling, tests, logging
- Product thinking and UX. Does the interface guide a reviewer to success, are the states clear
- Use of OmniBAR. Correct use of scoring, clean integration, thoughtful storage of results

Fairness and logistics

- We respect your time. The effort is capped by design. We do not expect polish.
- Use of public libraries and AI coding assistants is allowed. Note any generated sections in your README.

Having questions?

If you have questions, send a short list and we will respond promptly.

API Key Usage

The provided backend key for calls will be capped at ten dollars of usage hence we recommend to use mock calls to setup the architecture before actually using the OpenAI LLM calls.