Viven Engineering take-home: Build a Digital Twin MVP

For this take-home, you'll build a "digital twin" of yourself. You should design your digital twin to be able to answer questions *as you*—based on data you provide or connect.

This document outlines a core set of minimal product requirements to start with and then offers a few options for "deep-dive" areas for you to focus on.

Phase 1: Core requirements

1. **Digital twin functionality:** Your twin should be able to actually function as a twin.
2. **A front-end experience:** We want to query your digital twin using *some* interactive front-end, not just a command line call.
3. **Data integration:** Your twin should give responses grounded in some data about the user.

Note: These requirements are intended to form a *minimal* starting point for your submission, so you do not need to get too bogged down in them. Feel empowered to make simplifying design decisions for this first phase – choose certain use cases to design your twin for, take a not-so-beautiful first pass at the UI, use synthetic data to start with. You will focus on one extension area for the next phase, where depth will be more important to us.

Constraints:

- Your final submission should be standalone and easily reproducible (e.g. a shared GitHub repo/gist).
- You may use the following OpenAI API key for this project:

```
None
sk-proj-CZqarOPYZ8guycDoThz3bYO5fjH3T8xDtXsdxq0dP7nlPhsRBoPcvydIgWAL8VGNqWl50iBxOe
T3BlbkFJKKBlx1al25eXfv10aVpjnclR2fVtKgXHUS701_5okfyx-_u_CsO3VsENYDJCqCsbkbsVZrLicA
```

  Please be mindful of your usage and use the mini/nano models and design around their limitations if necessary.
- Do not include secrets (API keys, passwords, sensitive personal information) in your submission.

Phase 2: Deep-dive options

We are also interested in seeing which part of this problem excites you the most. After completing the minimum requirements above, *please choose exactly one of the following areas* to design and implement a meaningful extension. Please share which of the following you chose to focus on in advance of the interview, so that we can plan for our most relevant team members to join the conversation.

- **Data Integrations or Authentication:** How can you manage real-life integrations (e.g. Google Suite, Notion, Slack, etc.)? How can you enable safe logins to your digital twin product?
- **Twin Sharing:** What would you change about your product to allow others to talk to your twin? How would you support sharing different 'parts' of yourself?
- **Response Quality:** How can you expand the frontier of what your digital twin can do? How do you assess its successes and failures, and how can you improve it?

How we'll evaluate

We're primarily looking for:

- **System design:** Data models, retrieval & agentic flow
- **Engineering quality:** Clear code, reasonable structure, reproducibility
- **Depth of understanding:** How well do you understand your codebase? How will you design extensions?

We expect you to spend 4-6 hours on this project; please do not spend more than 10 (even if you're really getting into it). We value good scope choices and deep understanding over a sprawling submission—build something solid end-to-end, and document tradeoffs and next steps clearly.

IMPORTANT: Make sure to share your GitHub link and choice of deep-dive area with us as soon as you can!

If you have any questions, feel free to reach out Jonathan Delauri