

"New Year, New You" Portfolio Challenge

Jan. 12, 2026

<https://dev.to/challenges/new-year-new-you-google-ai-2025-12-31>

What this is

This submission is a **live, interactive developer portfolio** — not a PDF, not a résumé page, and not a GitHub profile.

It is a **deployed website** that presents my work, explains how I think as an engineer, and allows visitors to explore my projects with optional AI assistance.

The portfolio is intentionally simple, fast, and readable. Nothing is hidden behind logins or gimmicks.

Where you find it

You access the portfolio in **two places**:

1. **Embedded directly inside the DEV contest submission post**

- The portfolio is live and embedded using Google Cloud Run's embed feature
- You can interact with it *without leaving the DEV page*

2. **Via a direct public URL**

- The same portfolio is available at its Cloud Run deployment URL
- No account, signup, or configuration required

Both access points show the *same live application*.

How you use it (from a visitor's point of view)

1. Landing on the site (Home)

When you arrive, you see a short introduction that explains:

- who I am as a developer
- the kind of systems I build
- what this portfolio is meant to show

Optionally, you can trigger a short AI-generated introduction that summarizes my work in plain language.

Nothing auto-plays. Nothing demands attention.

2. Browsing the portfolio

The **Portfolio** section presents a small, curated set of projects.

Each project card shows:

- the problem it addresses
- the main technologies used
- a link to the GitHub repository
- a link to a live demo (when applicable)

This is intentionally selective — it highlights representative work rather than everything I've ever built.

3. Exploring a project in depth

Clicking a project opens a **dedicated project page**.

Here you'll find:

- a clear description of the problem
- key design decisions
- tradeoffs and limitations
- links to the underlying code
- notes on what I would improve next

In some cases, you'll also see a short, clearly labeled AI-generated critique that reflects on the design choices.

This section is about *engineering judgment*, not marketing.

4. Understanding how I work

The **How I Work** section explains my approach to:

- structuring projects
- managing data and versions
- ensuring correctness and explainability
- using AI as an assistant rather than a replacement

This gives context that a résumé or repo list usually can't.

5. Asking questions (optional)

The **Ask the Portfolio** page lets you ask questions such as:

- “Which project best demonstrates data integrity?”
- “What kind of problems does this developer enjoy?”

- “Which work shows the most technical depth?”

The responses are generated using the portfolio’s own content as context.
There is no memory, no tracking, and no open-ended chatbot behavior.

You ask a question → you get an answer → that’s it.

6. About the person

The **About** section provides brief personal context:

- background
- interests
- what motivates my work

It’s intentionally short and human.

What makes this different from a normal portfolio

- It is **live software**, not static content
- It treats projects as **engineering artifacts**, not thumbnails
- It uses AI **sparingly and transparently**
- It prioritizes clarity, restraint, and explainability

The goal is not to impress with volume or novelty, but to make it easy for someone to understand:

what I build, how I think, and why my work is trustworthy.

Technical Overview (Submission-Level)

Submission: A minimalist, AI-assisted developer portfolio deployed on **Google Cloud Run**, built with a small, transparent web stack and a single, well-scoped integration with **Google AI** (Gemini).

The portfolio is a static-first site with one optional dynamic capability: an AI layer that can introduce the developer and answer questions about the portfolio content using curated, project-level context.

The system prioritizes:

- clarity over novelty
- explainability over automation
- content quality over UI effects

All project code lives on GitHub; the portfolio acts as a curated, deployable interface over that work.

Stack (Intentionally Minimal)

- **Frontend:** Static site (Astro or Next.js in static export mode)
- **Backend:** Lightweight API (FastAPI or Node) for Gemini calls
- **AI:** Gemini via Google AI Studio
- **Deployment:** Single container on Cloud Run
- **Data:** Hand-authored JSON/YAML project metadata (no database)

No auth. No state. No user tracking.

Module-Level Technical Description

/home — AI-assisted introduction

Function:

Provides a short, dynamic introduction generated by Gemini.

Implementation:

- Static page
- Optional “Generate intro” button
- One API call to Gemini using a fixed system prompt
- No conversation state retained

Purpose:

Demonstrates responsible, minimal AI usage without obscuring content.

/portfolio — project index

Function:

Lists selected projects as structured cards.

Implementation:

- Static render from a single metadata file (e.g., `projects.json`)
- Each entry includes:
 - title
 - one-line description
 - technologies
 - GitHub link
 - optional live demo link

Purpose:

Fast scanning and deliberate curation; no infinite lists.

/project/{slug} — project detail pages

Function:

Presents a technical deep dive for each project.

Implementation:

- Static routes generated from metadata
- Markdown-based content for:
 - problem definition
 - design decisions
 - tradeoffs
 - limitations
- Optional Gemini-generated critique (pre-generated or on-demand)

Purpose:

Shows reasoning and engineering judgment rather than surface polish.

/how-i-work — engineering principles

Function:

Explains the developer's working style and technical values.

Implementation:

- Single static page
- Structured sections:

- tooling
- data hygiene
- reproducibility
- use of AI as an assistant

Purpose:

Communicates senior-level thinking and consistency across projects.

/ask-the-portfolio — constrained AI Q&A

Function:

Allows visitors to ask questions about the portfolio content.

Implementation:

- Simple chat UI
- Each request:
 - injects a short portfolio summary as context
 - uses a bounded prompt
 - returns a single Gemini response
- No memory, no follow-ups

Purpose:

Demonstrates AI augmentation without turning the site into a chatbot product.

/about — personal context

Function:

Provides human context without duplicating a résumé.

Implementation:

- Static content
- No AI involvement

Purpose:

Grounding and credibility.

Contest Compliance Summary

- ✓ Deployed on Google Cloud Run
 - ✓ Uses Google AI (Gemini) in a meaningful but limited way
 - ✓ Portfolio is embedded directly in the DEV submission post
 - ✓ Simple, inspectable architecture
 - ✓ Original synthesis of prior work
-

Bottom line

This submission reads as:

“A developer who understands when *not* to use AI — and does so deliberately.”

That restraint is the elegance.

If you want, next we can:

- lock the exact stack (Astro vs Next vs Svelte)
- draft the Gemini system prompts
- or write the one-paragraph **DEV submission description** in judging language