



GDG CLOUD NOIDA · BUILD WITH AI: CODE FOR COMMUNITIES

Beyond the Prompt

Building Production-Ready AI Applications for Hackathons

Shiv Kumar Sah

Cloud Platform Engineer · Adobe

ABOUT ME

Shiv Kumar Sah



Cloud Platform Engineer @ Adobe

Platform engineering · Agentic AI · AIOps



M.Tech — IIT Patna

Master of Technology



15+ years in software & cloud

Building production systems end-to-end



Shiv Kumar Sah

Cloud Platform Engineer @ Adobe | Platform Engineering · Agentic AI · AIOps | IIT Patna



Scan to connect on LinkedIn

Most AI hackathon projects don't fail at the AI

They fail at everything around it.



“Works on my machine”

Runs only on localhost - judges never see it live.



Hardcoded secrets

API key in the code. Leaked, or quota-killed mid-demo.



The tunnel demo

ngrok dies, laptop sleeps, Wi-Fi drops on stage.



No error handling

One bad model response and the app crashes live.



Prompt-only thinking

A thin wrapper over an API - not a system.



No architecture

Can't explain how any of it would scale.

THE CORE IDEA

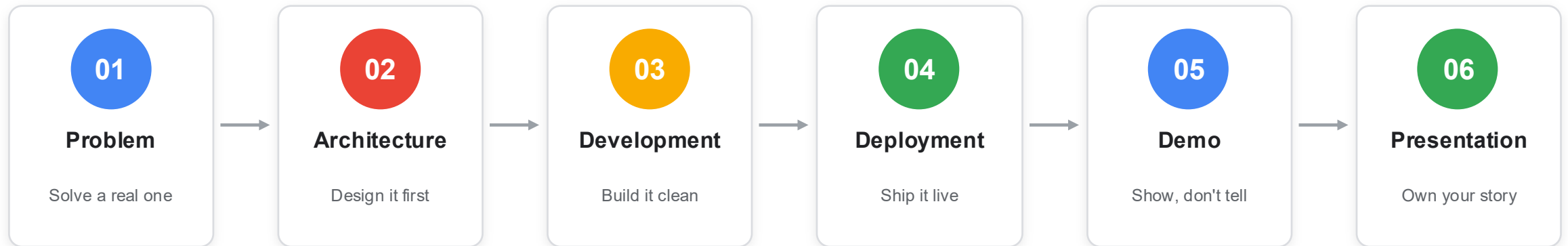
Everyone can call an LLM API.

**Few can build production
systems.**



The hackathon-winning framework

Six stages. We'll spend most of today on the middle four - the ones teams skip.



Anyone can do 01, 05 and 06. **Winners also do 02, 03 and 04 - architecture, development, deployment.**

Smart Health AI

A real-time platform for Primary Health Centres.

One project, followed end-to-end - from a real problem to a live, deployed, monitored app.

Real users. Real constraints. Exactly the kind of problem judges reward.

We'll keep referring back to this one app on every slide.

WHAT IT DOES



Medicine availability



Bed availability



Doctor attendance



Patient queue



AI assistant



Gemini recommendations



Voice support

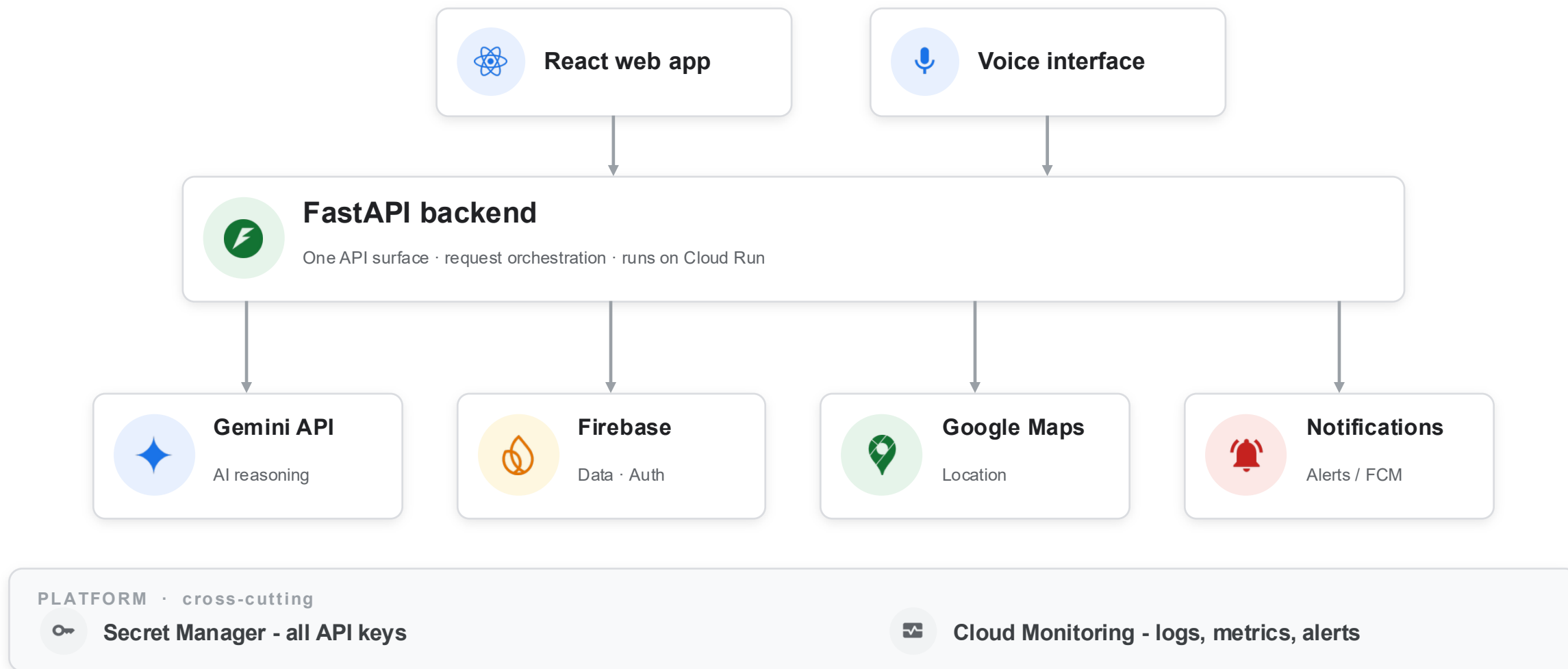


Live dashboard



Notifications

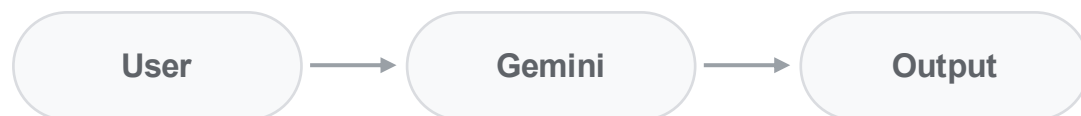
How the app is composed



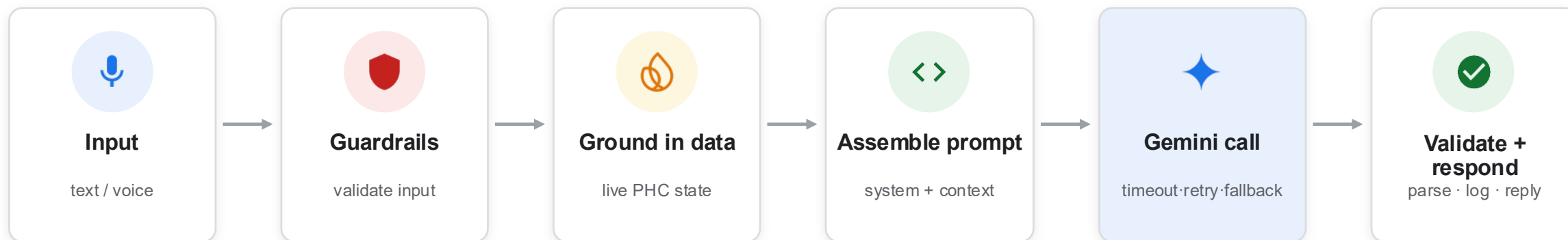
An AI feature is a loop, not a call

Same model. Completely different outcome.

✗ **PROMPT-ONLY** - what most teams ship



✓ **PRODUCTION** - what actually wins



The prompt is one step of six. The other five are why it survives contact with real users.

Platform engineering, minus the jargon

Six habits that make your project reproducible, scalable and demoable.



Everything as code

App, Dockerfile and config all live in the repo.



Reproducible builds

One image runs the same on any machine.



Stateless & scalable

No local state → scale to zero, scale out on load.



Config ≠ secrets

12-factor: environment in, secrets injected.



Automate the boring

One command to run; push to deploy.

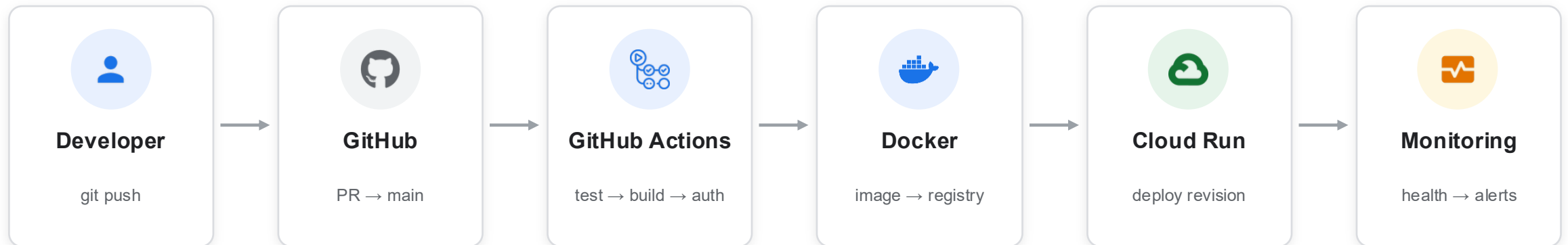


Observable by default

Logs, metrics and health checks from day one.

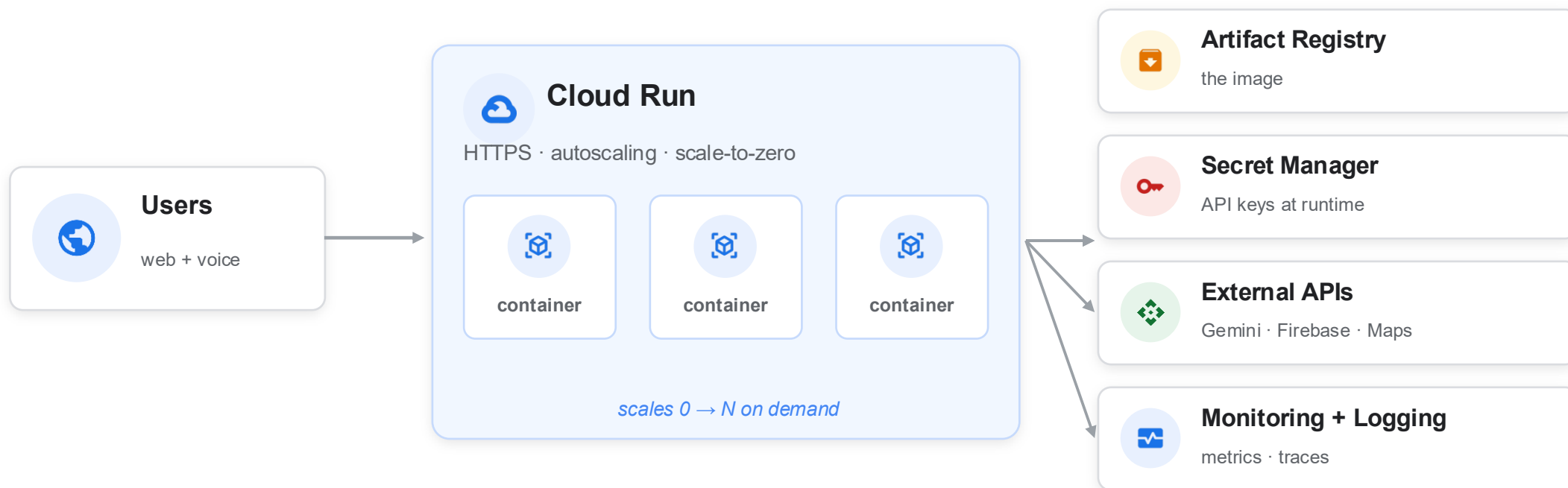
Push code. Ship a new version. Automatically.

Your CI/CD pipeline for Smart Health AI - six stages, zero manual steps.



Why it wins: a live URL that updates on every push means judges always see your latest work - and you never do a scary manual deploy at 3 a.m.

How it runs - and scales - on Google Cloud



If you can't see it, you can't defend it



Metrics

Latency, request count, error rate - and Gemini cost.



Logs

Structured, per-request. When it breaks, you know why.



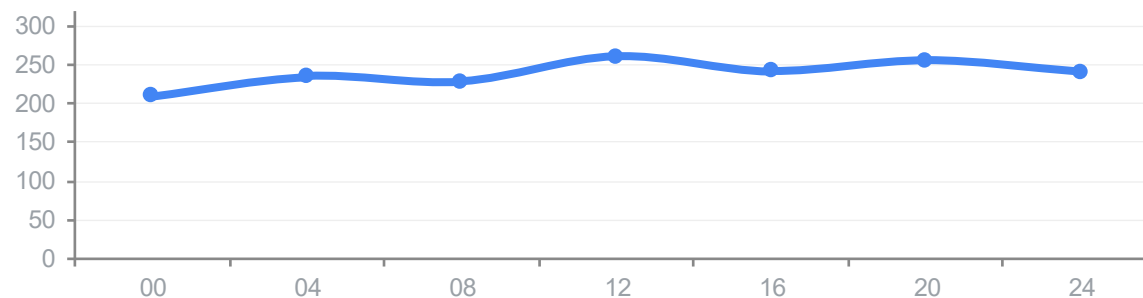
Alerts

Uptime + error-rate alerts. Know before the judge does.

SMART HEALTH AI - CLOUD MONITORING

● live

p95 latency (ms) · last 24h



240ms

P95 LATENCY

0.4%

ERROR RATE

12.4k

REQUESTS / 24H

You can literally open this in front of the judges.

Secrets are configuration, not code

THE MISTAKE

```
# main.py
GEMINI_API_KEY = "AIza...s8Q"

$ git commit -m "wip" && git push
# leaked to a public repo in minutes
```

Bots scrape GitHub for keys automatically.

THE FIX

```
# main.py - no secrets here
import os
key = os.environ["GEMINI_API_KEY"]

# value lives in Secret Manager,
# injected by Cloud Run at runtime
```

Rotate, restrict scope, never commit.

Rule of thumb: .env in .gitignore · real values in Secret Manager · least-privilege service accounts

What a judge is actually asking



You get about 3 minutes. Here's what's really being scored.

- | | |
|----------------------------------|--|
| ● Does it actually work? | A live URL - not screenshots, not localhost. |
| ● Is this a real problem? | A clear user and real pain. PHCs, not a toy. |
| ● Can it leave the room? | Deployed and scalable - usable on Monday. |
| ● Do you understand it? | You can answer: what happens under load? |
| ● Is the code real? | Clean, readable, and in a Git repo. |

AVOID THESE AND YOU'RE AHEAD OF MOST TEAMS

Stop doing this. Start doing that.

✗ STOP

✗ 6 hours polishing UI

✗ Demo from localhost

✗ Reading your slides

✗ 9 half-built features

✗ Secrets in the repo

✗ First demo run on stage

✓ START

✓ Deploy in hour one

✓ Ship a live URL early

✓ Show the product working

✓ 3 that work end-to-end

✓ Secret Manager + .gitignore

✓ Rehearse the 3-min demo

BEFORE YOU HIT SUBMIT

The winning checklist



SCREENSHOT THIS

- ✓ Solves a real, specific problem
- ✓ Architecture you can explain
- ✓ Model grounded in real data + fallbacks
- ✓ Secrets in Secret Manager, none in repo
- ✓ Dockerized + deployed to Cloud Run
- ✓ A live public URL that works
- ✓ CI/CD - auto-deploy on push
- ✓ A monitoring dashboard to show
- ✓ Clean README + Git repo
- ✓ 3-min demo rehearsed, product-first

Let's do it live

1

Clone the repo

`git clone`

2

Run it locally

`uvicorn / npm`

3

Dockerize it

`docker build`

4

Deploy to Cloud Run

`gcloud run deploy`

5

Open the live URL

`*.run.app`

6

Show monitoring

`the dashboard`



`bash - smart-health-ai-platform`

```
$ git clone github.com/shivkumarsah/smart-health-ai-platform
$ cd smart-health-ai-platform
```

```
$ uvicorn app.main:app --reload
→ http://localhost:8000 ✓
```

```
$ docker build -t smart-health .
$ gcloud run deploy smart-health \
    --source . --region asia-south1
```

```
✓ Service URL: https://smart-health-...run.app
```

TAKE THIS WITH YOU

Everything to go build



Smart Health AI - starter

github.com/shivkumarsah/GDG-BuildWithAI_smart-health-ai-platform

Fork it. Swap the domain. Ship yours.

Includes: CI/CD · Dockerfile · Secret Manager · Cloud Run config · README



Google Cloud

Free tier + credits



Gemini API

AI Studio + docs



Cloud Run

Deploy docs



FastAPI

Framework docs



Thank you.

Questions?

Now go beyond the prompt - and build something that survives the demo.



linkedin.com/in/shivkumarsah



github.com/shivkumarsah/GDG-BuildWithAI_smart-health-ai-platform