# Prompt Engineering Portfolio (Gemini API)

Hands-on projects demonstrating practical prompt engineering with Python + Google Gemini.

##  Projects

- \*\*Prompt Studio (Email Generator)\*\* – Structured, constraint-driven email creation with logs

- \*\*Gemini Chatbot\*\* – Lightweight multi-turn chat with short-term memory

- \*\*Resume Writer\*\* – ATS-friendly, role-targeted resume generator

##  Concepts Used

- Persona & tone control

- Few-shot prompting

- Chain-of-thought style guidance

- Prompt templates with variables

- Logging & reproducibility

##  Structure

prompt-engineering-portfolio/  
├─ README.md  
├─ requirements.txt  
├─ .gitignore  
├─ .env.example  
├─ src/  
│ ├─ common.py  
│ ├─ prompt\_studio.py  
│ ├─ chatbot.py  
│ └─ resume\_writer.py  
└─ logs/  
└─ .keep

bash

CopyEdit

##  Setup

```bash

# 1) Clone repo

git clone <your-repo-url>.git

cd prompt-engineering-portfolio

# 2) Python env (optional but recommended)

# python -m venv .venv && source .venv/bin/activate # macOS/Linux

# .venv\Scripts\activate # Windows

# 3) Install deps

pip install -r requirements.txt

# 4) Add API key

cp .env.example .env # or Copy-Item .env.example .env (Windows PowerShell)

# paste your GEMINI\_API\_KEY into .env

# 5) Run a project

python src/prompt\_studio.py

# or

python src/chatbot.py

# or

python src/resume\_writer.py

** Notes**

* Logs are written to logs/prompt\_log.txt by Prompt Studio.
* Keep .env private. Never commit API keys.
* Feel free to tweak the prompts for tone, length, and structure.