**Github Cheat Sheet**

**GIT CHEAT SHEET**

| **Task** | **Command** |
| --- | --- |
| Create dev branch | git checkout -b dev |
| Push dev branch to GitHub | git push -u origin dev |
| Switch between branches | git checkout main or git checkout dev |
| Merge dev → main | git checkout main → git merge dev → git push |
| Add all changes | git add . |
| Commit with a message | git commit -m "Your message" |
| Push to current branch | git push |
| See branches | git branch |
| See what branch you're on | git status |

**🧠 Regarding the System Prompt Getting Large**

Yes, it’s **totally normal** to feel like it's getting bulky. System prompts **can** be long — especially when you're guiding a structured, branded, interactive experience like you're doing.

Here’s the breakdown:

**✅ What’s Fine (for Now)**

* You’re using **OpenAI's GPT-3.5/4**, which can handle large context windows (up to 16K or 128K tokens depending on the model).
* The **system prompt is just one part** of that total context. You're safe keeping it detailed — especially for control over output and behavior.
* A longer prompt often **prevents hallucinations**, so in your case, it's helpful.

**⚠️ But Eventually…**

If it gets much longer, it can:

* Eat into your **available tokens** for user input + retrieval context.
* Become harder to **maintain or scale** across multiple personas or page types.

**🧩 Recommended Strategy: Split It Into Structured Prompt Parts**

As you scale:

1. **Split into 2–4 parts** in code:
   * Global instructions (HTML formatting, brand tone, structure rules)
   * Persona-specific voice or behavior
   * Course-specific overrides (like how to define HOME or handle WFA visuals)
   * Optional dynamic injections (user device type, recent interactions, etc.)
2. Combine them **in the backend dynamically** before sending the full prompt.

python

CopyEdit

final\_prompt = f"{global\_instructions}\n\n{persona\_prompt}\n\n{course\_rules}"

1. Store each in the **Supabase DB** under a ai\_prompt\_fragments table or similar for dynamic loading.

**Bonus Option:**

You could even give each **page** or **topic** a mini "micro-prompt" (like for WFA, AWFA, WFR...) and append that only when relevant — like a plugin-style system.

Would you like a schema idea for storing and managing that in Supabase? Or an updated FastAPI snippet for dynamically merging prompts?