Prompt & Context Engineering

__/__/____

→ Have proper document Structure do faste efficient use of content window size

→ for a project the doc should be

Docs/
- Bug_tracking.md
- implementation.md
- project_structure.md
- UI_UX_doc.md

.cursor/rules
- generate.mdc
- workflow.mdc ⟶ short and clean methods

PRD.md → Product requirement Document

→ decide the techstack by yourself.

⟶ 3-1-0

__/__/__

Tips for new code:

Tip 1: Choose the technology by your own research and ask the agent to do the project on the same

Tip 2: Always choose popular technology. coz AI is much more familiar & Large community to help.

Tip 3: Design a solution and ask the AI to work it. Dont give open-ended questions to AI.

Tip 4: Break big task to sub-task & list and give the AI as list of tasks.

Tip 5: find the right balance of taskscope (not to big or too small). Comes with experience).

Tip 6: Examples in prompts with edge cases so the AI can understand better.

Tip 7: give some sample code to learn to AI.

__/__/____

Tip8: Use @doc & @web context utilities to built a working and test pratica coding.

Tip9: Make sure add prompts to ensure security and make the code perform well.

Tip10: Ask for any additional considerations for building any feature.

Tip11: Use either of two approach

AI to write test after development

Test-driven development
→ write test first & develop a code to satisfy all tests.

Tip12: Have AI create your documentation for precise understa...

Tip13: give clear names to fn's. with specific names.

__/__/____

Modifying Existing Code:

Tip14: Keep code organized by asking AI to refactor & simplify the code.

→ helps in better understanding

- Eg: phase splitting data parsers to csv parsers, xml parsers etc...

Tip15: properly manage the context to AI to prevent hallucinations.

Tip16: Tag specific files to context to get the AI to understand in depth

Tip17: Keep files at 300 lines of code or less to make AI understand easier & efficient.

Tip18: Full conversation is a context so use new chat for new thing.

Tip19: New feature → New chat.

Tip20: Tell AI what works well (need no changes) & what doesn't

at the end of every feature development.

__/__/____

Tip21: Ask AI to find edge cases & all possible bugs.

Tip22: Ask AI to review the code.

Tip23: What's & How's of a current code before editing.

TROUBLESHOOTING:

Tip24: Be specific about what is going wrong & What is correct.

Tip25: Share screenshots of what is wrong.

Tip26: Share the entire errors.

Tip27: Beaver method: ↓
⇒ add logs to everything
⇒ give that to AI to find the error quickly.

Tip28: Ask AI to explain the code & figure out your way to handle bug.

__/__/____

Tip29: Ask AI to use "RADICALLY different approach".

forces to think differently.

Tip30: Know when to stop Asking AI & look by your own.

LEARNING TO CODE:

Tip31: Keep it simple + you're new dev to AI.

Tip32: Explain, code line by line with comments.

→ Implement (or) Ask AI if you've seen this pattern before to enhance in learning.

Tip33: "Ask AI to explain specific tech or concept.

Tip34: Ask how to build.

__/__/____

Tip 35: Ask AI to show you good examples for understanding.

Tip 36: Tell AI what you understand & what you don't.

Tip 37: focus on learning & fundamental core concepts.