

Reflection – My Experience Using AI Tools for Full-Stack Development

What Went Well

- **Code generation was solid** – Auth (JWT + bcrypt), React components, and Dockerfiles worked out of the box.
 - **Modular approach** – Doing Auth → Tasks → Uploads → WebSockets → CI kept results clean.
 - **Debugging help** – AI quickly spotted errors and suggested fixes.
 - **Learning** – Picked up async FastAPI, React hooks, cleaner Docker builds, and GitHub Actions.
-

What Annoyed Me

- **Config files overwritten** – e.g., my GitHub Actions workflow got replaced.
 - **Risky Docker advice** – Suggested `docker system prune -f` which wiped other containers.
 - **Needed very specific instructions** – Vague requests gave poor results, detailed ones worked.
 - **Context loss in long chats** – Sometimes forgot architecture.
 - **Over-engineering** – Added unnecessary complexity for simple tasks.
-

Lessons Learned

- Be clear and specific.
 - Work one module at a time.
 - Review commands before running.
 - Commit often for rollback.
 - Let AI handle boilerplate, but keep configs, business logic, and security in my hands.
-

My Take

- **Time saved:** 6–10 days → 8–10 hours.
 - **Code quality:** good base, needed polish.
 - **Reliability:** strong for standard tasks, fine for bigger parts with review.
-

Final Thoughts

AI felt like a **super-fast junior dev** — great for speed, but still needs supervision. Would I use it again? **Definitely**. Next time: stricter boundaries, module-by-module, and no blind trust on system commands.
