



**Let's make a new startup
with AI.
Finally, I don't need
programmers. Fuck yeah**



ARE YOU CRAZY OR INSANE?



**Tools like v0, Cursor, Windsurf can
generate apps rapidly and amazingly, but...**

- ◆ Have you thought about **security**?
- ◆ How do you want to **version control**?
- ◆ What about **optimization, deployment, or architecture**?
- ◆ Are you considering **cost management** and **licenses**?

⚠ These software
engineering essentials can't be ignored!



*It's quite a
responsibility.
I just want to have fun.*



HERE ARE SOME TIPS





Security

- **Keep API keys secret:** Never hardcode keys in your app. Store secrets in environment variables or a free secrets manager to keep them safe. 1Password has some interesting features that are worth looking at!
- **Scan for vulnerabilities:** Run a free Snyk scan (or enable GitHub Dependabot) to catch known security flaws in your code/dependencies early.
- **HTTPS everywhere:** Always serve your app over HTTPS. Tools like Let's Encrypt give free SSL certificates!
- **Lock your accounts:** Enable 2-factor authentication on critical services (GitHub, cloud platforms, AI tools etc.). It's free and prevents easy hacks on your accounts.
- **Set hard budget limits** in all tools that charge for usage, like Chat-GPT. It might save you a lot of money!





Version Control

- **Stop emailing zip files:** Use Git for version control. It tracks every change so you can undo mistakes and never lose code again.
- **Free cloud backup:** Push your code to GitHub or GitLab (free private repos) for safekeeping. If your laptop crashes, your code stays safe in the cloud.
- **Branch for safety:** Experiment on a new branch instead of directly on main. Merge when it's working, or delete it if not – no harm done to your main codebase.
- **GitHub Desktop helps:** If the command line scares you, use the free GitHub Desktop app. It's a point-and-click way to commit, push, and pull code – no terminal needed. Most coding editors have already build-in graphic interface for GIT.
- **Automate testing & deploys:** Use GitHub Actions (free) to run checks or deploy your app on every push. Automation catches issues early and saves manual effort.





Deployment

- **One-click hosting:** Deploy your app on platforms like Railway Vercel. You just push code and chill.
- **No server?** No problem: Firestore offers hosting, database, and auth with a free plan. Great way to get backend features without managing any servers.
- **Static site hero:** Use Netlify, Vercel, or GitHub Pages for a static website or frontend app. They're free – just connect your repo or drop your files to go live.
- **Continuous deploy:** Link your code repo to your host (Railway, Vercel, etc.). Every time you push changes, the service auto-deploys your app. No more manual uploads!
- **Serverless functions:** Need a bit of backend logic? Use cloud functions on a free tier (Firestore Functions, Vercel, Netlify, AWS, GCP, Azure). Your code runs on demand.





Rules

- **Split project into smaller tasks** – Easier to manage, debug, and parallelize.
- **Write detailed documentation first** – Force model/agent to follow it. Include goals, edge cases, and examples.
- **Monitor token and compute costs** – Large files or loops can explode usage. If your app is using AI, then integrate your app with LangSmith or a similar observability tool.
- **Limit workspace scope** – Restrict to relevant directories or files only for each task.
- **Define clear input/output format** – Avoid ambiguity. Keep structure strict and predictable. Define all details and rules in each prompt.
- **Use isolated test cases** – Help the model validate logic step-by-step. Ask the model to create tests according to BDD (Behavior Driven Development).
- **Fail fast, retry smart** – Catch issues early, and design fallback or retry logic.



Michał Mazur

@michmzr

```
// Got questions? Ideas? Bugs to report?  
if (you.needHelp || you.haveThoughts) {  
  commentBelow();  
  sendDM("I'm friendly, promise 😊");  
}
```

