**What I Learned from Building FocusBuddy**

**Introduction**

When I started building FocusBuddy, I just wanted a script that could help me get through a few 25-minute focus sessions without getting overwhelmed. I didn’t expect the process to teach me so much—not just about scripting and terminal tools, but about productivity, accessibility, and myself.

In this final blog post, I want to reflect on the bigger lessons I’ve learned through this project. FocusBuddy became more than just a timer—it became a case study in designing for neurodivergence, automating motivation, and building systems that support mental health.

**1. Simplicity Wins**

The most important lesson? **Simple tools can have a big impact.**

FocusBuddy is under 150 lines of Bash. It uses only built-in tools like sleep, read, and date. There’s no fancy backend, no complex logic, no dependencies.

Yet despite its simplicity, it changed how I work. It made it easier to start. It removed the “what do I do now?” question. It gave me structure.

This taught me that a tool doesn’t have to be complex to be powerful. It just has to solve the right problem—and solve it clearly.

**2. Design Is Empathy**

FocusBuddy was never meant to be a product for everyone. It was built specifically for brains like mine—easily distracted, frequently overwhelmed, always juggling too much.

That specificity was its strength.

Every choice I made, from the visual break messages to the one-command startup, came from personal frustration. I wasn’t designing from theory—I was designing from lived experience.

And that taught me this: **Design isn’t about adding features—it’s about removing friction.**

When you understand your user (especially when that user is yourself), you build differently. You don’t just ask “what can this do?” You ask, “what will this feel like at my worst?”

**3. Automation Is Empowerment**

There’s a common narrative in tech that automation is about efficiency—doing more, faster, with fewer people.

But for me, automation wasn’t about speed. It was about **reducing decision fatigue**.

Every day, I’d face dozens of micro-decisions: When do I start? What should I do first? Is this task important? When should I take a break? Should I log what I did?

FocusBuddy removed all of that. One command, and it took over. It guided me from task to task, session to break, without me needing to think.

This reframed automation in my mind: it’s not just about scaling—it’s about self-care. It’s about building systems that support you when your brain can’t.

**4. Logging Creates Momentum**

At first, the log file was just for fun—a CSV of session times. But it quickly became something more.

Seeing my tasks stack up in sessions.csv and done.txt changed how I felt at the end of each day. no longer had to *remember* what I did. I could *see* it.

This gave me a sense of accomplishment—something people with ADHD often struggle with.

It also gave me data. I started noticing patterns:

* I was most focused between 10am and 1pm.
* My mood impacted my session length.
* Breaks helped, but only if I stood up and moved.

These weren’t things I tracked consciously. But by automating the logs, I accidentally created a feedback loop.

**5. Self-Compassion Is a Feature**

One of the unexpected benefits of FocusBuddy was how kind it felt. The messages were gentle. The tone was encouraging. It didn’t punish me for quitting early or forgetting a session.

This wasn’t just good UX—it was **intentional**.

Most productivity tools are rigid. They treat missed goals as failures. But FocusBuddy was built for a brain that doesn’t always work predictably. That meant building a tool that didn’t expect perfection.

Instead of alarms, it uses friendly cowsay messages. Instead of asking “why didn’t you finish?”, it asks “want to try again?”

This made me more likely to come back. More importantly, it reminded me that I’m not lazy—I just need better systems.

**6. Technical Confidence Grows With Purpose**

Before this project, I’d written small Bash scripts, but nothing this interactive. I didn’t know how to trap signals or log to CSV. I didn’t know how to show a countdown timer.

But with FocusBuddy, every new feature came from a real need. I wasn’t learning syntax for its own sake—I was solving a problem I deeply cared about.

That made learning easier. More engaging. More fun.

By the end of the project, I felt more confident with Bash scripting, system tools, and even scheduling jobs with cron.

It reminded me that **purpose drives learning**. When you care about the outcome, the tech becomes secondary.

**7. Community Matters**

When I started sharing FocusBuddy, I didn’t expect anyone else to use it. But soon, a few classmates and developers tried it and said: “Hey, this actually helps.”

Some suggested features. Others forked the script and made their own versions.

This experience reminded me that even small tools can create shared understanding. There are a lot of people out there—especially developers—struggling with similar attention issues. We just don’t always talk about it.

FocusBuddy became a conversation starter. A way to talk about mental health, neurodiversity, and software in the same breath.

**8. There’s Beauty in the Terminal**

I’ve always loved the command line—it’s fast, distraction-free, and honest. But FocusBuddy made me appreciate it even more.

There’s something deeply satisfying about typing one command and having everything flow from there. No popups, no syncing issues, no lag.

FocusBuddy reminded me that you don’t need a GUI to build something beautiful. You just need intention—and a little bit of color from lolcat.

**Final Reflection**

FocusBuddy started as a personal project. It became a tool, then a habit, then a mindset.

It taught me that I don’t need to fix myself—I just need to build systems that fit me. Systems that assume forgetfulness, support focus, and celebrate small wins.

I also learned that automation isn’t just technical—it’s emotional. It’s about removing friction, reinforcing structure, and giving yourself the space to just *do the thing*.

For anyone considering building their own automation project, I offer this advice:

* Start with a pain point you actually feel.
* Keep the first version simple.
* Let your tool grow with you.
* And make it kind—because life is already hard enough.

Stay focused, and be kind to your brain.