**How FocusBuddy Works: A Terminal-Based Productivity Assistant**

**Introduction**

In the first blog post, I explained why I built FocusBuddy—a lightweight, terminal-based productivity tool designed to help people (especially those with ADHD) manage their focus cycles with less friction. This time, I’ll walk you through **how it works** under the hood.

FocusBuddy may be a small Bash script, but it does a lot with a little. It combines task selection, a Pomodoro timer, session logging, and break reminders—all inside a clean command-line interface. This post breaks down the key features, tools used, and how I built it step by step.

**Project Goals Recap**

The idea behind FocusBuddy was simple: automate the entire focus session workflow.

* Prompt user to choose a task
* Start a 25-minute focus timer
* Log session start/end and task to a CSV file
* Mark task as completed in a task list
* Trigger a 5-minute break reminder
* Optionally repeat the cycle or exit

Everything happens in one shell session—no tabs, no apps, no context switching.

**Technologies and Tools Used**

FocusBuddy is written in **Bash**, using standard Linux tools that are pre-installed on most systems:

* read for user input
* date for timestamp logging
* sleep for timers
* figlet, lolcat, cowsay for fun terminal visuals (optional)
* awk, sed for text processing
* echo, printf, tee for display and logging
* cron (optional) for scheduling sessions

No external dependencies are required to use the core functionality.

**Focus Session Flow**

Here’s what happens step-by-step when you run the script.

**Step 1: Welcome Screen**

The script begins with a warm welcome:

figlet FocusBuddy | lolcat

cowsay "Ready to focus?" | lolcat

This may seem unnecessary, but a friendly greeting helps reduce startup anxiety. It’s also part of making the tool ADHD-friendly—engaging, but not overwhelming.

**Step 2: Task Prompt**

You’re asked to either choose a task from a saved list or enter a new one:

read -p "What will you focus on today? " task

The task string is stored in a variable for later logging. You don’t need to navigate a GUI or open a separate task manager. Everything is handled in one place.

**Step 3: Start the Timer**

Once the task is confirmed, the script logs the start time:

start\_time=$(date +"%H:%M")

start\_epoch=$(date +%s)

It then starts a 25-minute countdown:

echo "Starting 25-minute focus session..."

sleep $((25 \* 60))

You can replace this with a visible countdown later, but for now, it runs silently.

**Step 4: End Timer and Log Session**

When the timer ends, the script records the end time and duration:

end\_time=$(date +"%H:%M")

end\_epoch=$(date +%s)

duration=$(( (end\_epoch - start\_epoch) / 60 ))

This data is then appended to a CSV log:

echo "$task,$start\_time,$end\_time,${duration}m" >> sessions.csv

This file becomes a growing record of everything you’ve accomplished. It’s timestamped, readable, and can be imported into Excel, Notion, or any planner.

**Step 5: Mark as Completed**

To reinforce a sense of accomplishment, FocusBuddy also appends the task to a done.txt file:

echo "[$(date +%F)] $task" >> done.txt

This simple text file becomes a daily archive of finished work. I often check it at the end of the week for a dopamine boost.

**Step 6: Break Reminder**

After logging, the tool gently encourages a 5-minute break:

echo "Take a 5-minute break! Stretch, breathe, drink water."

sleep $((5 \* 60))

You can customize this part with music, visuals, or guided meditation suggestions.

**Step 7: Repeat or Exit**

Finally, you’re asked if you want to start another session:

read -p "Start another session? (y/n) " choice

If you type ‘y’, the process restarts. If not, the script exits with a kind farewell:

figlet "Good job!" | lolcat

**Bonus Features**

**Optional: Task List Integration**

You can preload tasks from a tasks.txt file and display them with numbered choices.

select task in $(cat tasks.txt); do

echo "You chose: $task"

break

done

This makes it easier to reuse tasks and build routines.

**Optional: Auto-Launch with Cron**

To make FocusBuddy part of my daily schedule, I used cron to run it automatically:

0 10 \* \* \* /home/salman/focusbuddy.sh

This makes it pop up every morning at 10 AM like a personal assistant.

**Optional: Config File**

Advanced users can use a .focusbuddyrc config file to define session length, break time, paths, and themes.

**How Long It Took to Build**

It took me 2–3 days to go from concept to functional script. I started by mocking up the flow in pseudocode, then wrote a basic version. I iterated quickly—testing timers, refining prompts, and improving the log format.

Most of the complexity came from polishing the user experience and making it ADHD-friendly—not from the code itself.

**Challenges Faced**

* Handling user input cleanly (especially spaces and quotes)
* Preventing log duplication
* Keeping the script short but readable
* Deciding how much “fun” to include (too much color = distracting)
* Making it fail-safe (e.g., what if the CSV file is missing?)

I added small failsafes like touch sessions.csv and graceful exits if something goes wrong.

**What I’m Proud Of**

* It works on every machine I’ve tried
* It’s under 150 lines of code
* It genuinely helps me start and finish tasks
* It’s open source and shareable
* It respects neurodivergent workflows

**Example Output**

Here’s what a user might see:

\_\_\_\_\_ \_ \_ \_

| \_\_\_| \_\_ \_\_\_ \_\_\_| | \_\_\_\_\_ \_ \_\_ | |\_\_ (\_)\_ \_\_

| |\_ | '\_\_/ \_ \/ \_\_| |/ / \_ \ '\_ \| '\_ \| | '\_ \

| \_|| | | \_\_/ (\_\_| < \_\_/ | | | | | | | |\_) |

|\_| |\_| \\_\_\_|\\_\_\_|\_|\\_\\_\_\_|\_| |\_|\_| |\_|\_| .\_\_/

|\_|

What will you focus on today? Write blog post

Starting 25-minute focus session...

...

Session complete!

Task: Write blog post

Duration: 25 minutes

Take a 5-minute break.

**Final Thoughts**

FocusBuddy may be written in Bash, but it functions like a true productivity assistant. Its power lies in its simplicity. You don’t need to install an app, sign up for a service, or manage a UI. Just open your terminal, type one command, and you’re on your way.

In the next blog, I’ll talk about the improvements I want to make—like dynamic task tracking, mood logs, even a lightweight GUI version. FocusBuddy is just getting started.

Thanks for reading.