

## # ☒ \*\*Developer Habit & Task Checklist\*\*

### ## \*\*1. Daily / Consistent Coding Habits\*\*

- [ ] Write code every day (even 30 minutes)
- [ ] Practice coding challenges regularly
- [ ] Type out and run code from tutorials
- [ ] Experiment with code and break it to learn
- [ ] Review your own code
- [ ] Refactor old code
- [ ] Keep a consistent learning schedule
- [ ] Take notes on what you learn

### ## \*\*2. Projects & Practical Application\*\*

- [ ] Build small projects regularly
- [ ] Create real-world applications from scratch
- [ ] Finish projects you start
- [ ] Build a CRUD app
- [ ] Build a portfolio website
- [ ] Build a simple API
- [ ] Build an authentication system
- [ ] Build a CLI tool
- [ ] Create automation scripts
- [ ] Build one real-world clone (Twitter, Trello, etc.)
- [ ] Showcase all projects in a portfolio
- [ ] Work on real-world projects (freelance, internships, personal apps)

### ## \*\*3. Core Knowledge & Technical Skills\*\*

- [ ] Learn one programming language deeply
- [ ] Learn and practice new programming languages
- [ ] Understand programming fundamentals (variables, loops, functions)
- [ ] Understand data structures and algorithms
- [ ] Understand object-oriented programming (OOP)
- [ ] Understand basic software architecture (client-server)
- [ ] Learn how to search effectively (Google, Stack Overflow, docs)
- [ ] Learn how to use documentation
- [ ] Read official documentation
- [ ] Learn testing basics
- [ ] Write tests for your code
- [ ] Learn the software development lifecycle
- [ ] Learn to read error messages
- [ ] Learn debugging tools/techniques
- [ ] Learn command line basics
- [ ] Learn basics of networking & APIs
- [ ] Learn basics of databases
- [ ] Learn common design patterns

### ## \*\*4. Debugging & Code Quality\*\*

- [ ] Debug systematically
- [ ] Trace problems to their root
- [ ] Don't ignore issues
- [ ] Use debugger tools and logs
- [ ] Write clean, maintainable code
- [ ] Document your code and projects thoroughly

☐ Regularly review and improve old code

### ## \*\*5. Reading & Analyzing Code\*\*

- ☐ Read other people's code
- ☐ Analyze open-source codebases
- ☐ Participate in code reviews
- ☐ Study different design patterns and approaches

### ## \*\*6. Version Control & Collaboration\*\*

- ☐ Learn and use Git
- ☐ Commit your code regularly
- ☐ Use GitHub
- ☐ Learn branching and merging
- ☐ Collaborate with other developers
- ☐ Pair program
- ☐ Contribute to open-source (optional but helpful)

### ## \*\*7. Learning & Continuous Improvement\*\*

- ☐ Learn through online courses
- ☐ Learn through books
- ☐ Learn through tutorials
- ☐ Stay updated via tech blogs
- ☐ Follow newsletters or podcasts
- ☐ Improve problem-solving and logical thinking
- ☐ Break problems into smaller steps
- ☐ Keep improving past projects

### ## \*\*8. Community, Networking & Asking for Help\*\*

- ☐ Ask for help when stuck (after trying first)
- ☐ Clearly explain what you've already tried when asking for help
- ☐ Join developer communities and forums
- ☐ Participate in meetups or workshops
- ☐ Share your knowledge with others
- ☐ Teach or explain concepts you learned
- ☐ Network online or in-person

### ## \*\*9. Professional Development\*\*

- ☐ Create a GitHub portfolio
- ☐ Write clear README files for projects
- ☐ Build a LinkedIn profile
- ☐ Apply for internships or junior roles

### ## \*\*10. Health, Mindset & Productivity\*\*

- ☐ Take breaks regularly
- ☐ Prioritize sleep
- ☐ Exercise consistently
- ☐ Maintain a good diet
- ☐ Stay patient when facing errors
- ☐ Focus on consistency, not perfection
- ☐ Maintain persistence and a growth mindset
- ☐ Stay organized with your development environment
- ☐ Set clear short-term learning goals
- ☐ Set clear long-term learning goals