

# THE AI TRAP

-ANANYA.B.M,III SEM CSE(AI&ML)

"I shipped code I don't understand and I bet you have too"

-Jake Nations

The AI Trap: Are We Coding Faster, But Understanding Less? Let's be honest: we have all done it. You have a deadline, a complex assignment, or a bug you just can't fix. You paste the prompt into ChatGPT or Gemini, copy the code it spits out, run it, and... it works! You submit it and move on.

But if someone asked you to explain exactly how that code works line-by-line, could you?

Backlog items that used to take days are now taking hours and large refactors that have been on the books for years are finally being done. There's a saying that Large Production systems always fail in unexpected ways. When you think and solve you understand the code better than you are debugging. And the problem is now we're generating code at such speed and such volume where our understanding is having a hard time keeping up.

The hard part was never about the mechanics of coding or the syntax, the problem is that we are not actually understanding the problem and designing the solution and no tool can eliminate that fundamental difficulty and this is because most of us get confused between the phrase simple and easy where they are not same in more sense simple made easy.

The problem isn't that the code is bad—it's that we don't understand it. And when a system you don't understand breaks in production you are powerless to fix it. Don't Outsource Your Thinking. So, should we stop using AI? Absolutely not. But we need to change how we use it. The solution is to become "Context Engineers." Instead of asking AI to "just write the code," we should do the hard work first:

- \* Research & Plan: Understand the problem deeply yourself.
- \* Write a Spec: Create a detailed plan of what the code should do.
- \* Generate: Only then, use AI to write the syntax based on your plan.

## The Takeaway for Us Students

As future engineers, our value isn't in how fast we can type or how well we can prompt a chatbot. Our value lies in understanding systems.

Use AI to speed up your hands, not to replace your brain. The best developers won't be the ones who generate the most code—they will be the ones who actually understand what they built.