



The Earning-Before-Learning Trap: How Short-Term Income Pressure Weakens Long-Term Engineering Excellence

Students are rushing to earn before they learn, using AI to deliver things fast, skipping fundamentals and building learning debt that break their ability to think, adapt, and engineer real systems.

By **Haidar Ali** | University of Agriculture | Published: October 2026

✉ mughalhaider08@gmail.com | Position: Software Engineering Student

Introduction

In today's digital world, earning has become more urgent pursuit than learning. A growing number of students and self-taught beginners are falling into this pattern, chasing how many dollars they can earn without building the depth that real engineering demands. *According to the Stack Overflow 2024 Developer Survey, over **40%** of professional developers are self-taught, and many enter paid work within their first two years of coding, often before their fundamentals are solid [2].* They generally focus on the payment, not the learning process, and completely ignore the consistent practice that actually builds long-term growth in future. The pattern is like: a student watches a success story of someone who earned thousands of dollars in a month, then student suddenly triggers, enrolls in a three-months bootcamp or course, posts a gig on a freelance platform, and luckily gets a client. The project gets delivered – but under deadline pressure, the underlying concepts behind the work fade away. The rush to earn early, driven by freelance platforms, social media success stories, and parental and economic pressure, is silently weakening the engineering depth of an entire generation. I call this “**Earning-Before-Learning Trap**”.

Technology Overview

The Human nature is always to find short cuts, the same pattern exists in software industry where workloads and pressure push them to do shortcuts. Beginners always in a hurry to complete the project by cutting the understanding edge and deliver the project on time. This approach implies the cost of rework and dirty code **Ward Cunningham coined it in 1992 called Technical debt**—a cost which interest paid later.

Technical debt lives in code. Learning debt lives in the engineer's mind. **Learning debt** is the knowledge gaps created when a student or a developer skips foundational understanding in order to gain faster and surface-level acquisition and interest is paid later when career grows and inability to adapt to changings systems in future.

Technical debt can be repaid but learning debt is harder to repay once skipped.

Now Artificial Intelligence comes in, before that technology, if a student did not understand something, he went to google, stack Overflow, documentation, trial and error. Although it was frustrating, but learning happened. Now students and beginners just copy the error and fixed by AI very quickly and project gets done before deadline. A student who is in learning age, gets the freelance project, the deadline and greed of earning money, having no idea what the code does, when a client comes back with change request, there is no understanding to solve it—only panic. This is technical depth takes time to absorb. But when students in his early career do this, it will be very frustrating when their career grows.

It is the story of my friend who was working as a freelancer and got a job. He says that he did not write even a single line of code as AI writes the code and finding the errors quickly. Whenever a bug introduces I just go to AI



assistant and it would be done. But in this process no learning takes a place. He could operate the tools, but not engineer the solution.

Key Findings & Impact

The Earning-before-learning trap is not just an opinion, it is visible in global data, hiring trends, AI adoption and family pressure in Pakistani homes.

Pakistan ranks **4th globally** in freelancing nearly **\$400** million in IT exports during FY 2025-26 reports [1]

Stack Overflow 2024 developer survey reports showed over 40% of professional developers and self-taught with his early in career [2] The industry is expanding rapidly in width but not in depth.

The hidden pressure: In many Pakistan's homes, there is a trend follows badly. Engineers worth is measured in dollars, not problems solved. The mindset is very narrow – families pressure student who is in early stage of career they want to shift their minds towards earning first and ignoring the process of learning. Parents and siblings are unfamiliar with engineering core builds. They measured in monthly income rather than concept mastered. A student studying system design but earning nothing is ignored as it is wasting of time. While anyone in relatives earns \$500 dollars on fiverr is encouraged. This immediate pressure pushes thousands of high-potential students into freelance platform before they are technically ready.

Social media success stories have a key role making the students attention segregated from their core engineering principles. Ones find a post on a developer showing thumbnail "he earned \$1000 from in a single day" It reshapes his priorities and he just follows the money instead of fundamentals and deep learning.

Artificial Intelligence (AI) although fast and rapid. Artificial Intelligence has reshaped developer productivity. GitHub research shows developers complete tasks **up to 55% faster** using Copilot [3], and **76%** of developers now use or plan to use AI coding tools [2]. Speed is here but they can't explain, debug or extend without AI. Productivity increases understanding does not.

Recommendations & Actions

Solving this trap requires more than industry awareness – it demands through the coordinated shift of universities, families, and students themselves.

Academic institutions must look beyond traditional lectures and encourage the engineering depth as a measurable outcome. They should integrate problem-solving, system design, and AI- literacy modules that teach students when to use AI and when to thought without it. Professors should actively inject the awareness of latest learning sources to the students, and rewarding deep understanding over traditional assignments. Government should encourage the learning by giving students stipends while studying along with tuition fee so student can focus without any financial pressure by family and friends.

There should be an awareness-sessions for the **parents** whose mindset is only to get paid after degree or during degree. They should realize that engineering excellence requires years of disciplined practices. Parent-teacher session, awareness campaigns, and media coverage should highlight that real expertise rarely comes in three months.

For students themselves should carry the responsibilities. Students facing financial issues should consider part-time work outside their actual field—tutoring, content writing, teaching the learned concepts to junior students rather than jumping into freelancing when they are just starting in the field which stops their learning. Students first should evaluate their interest to particular specialized field. Choose the right interest then earn with it. **Study to thrive not to earn once strong core has been established then earning automatically follows.**

Every engineer in Pakistan should ask himself the question. Am I building a career, or just collecting the dollars in pressure? Real engineers are shaped in slow, frustrating, unintentional hours of genuine learning. The money will flow—but only to those – but only to those who first grinds the fundamentals to become worthy in future.



About the Author

Haidar Ali is a software engineering student with keen interest in software engineering culture, research, and Pakistan's growing tech industry. He can be reached at mughalhaider08@gmail.com

Keywords: Software Engineering Education; Freelancing; Artificial Intelligence; Learning Debt; Pakistan Tech Industry; Software Engineering; Technical Debt; Pakistan Tech

References

- [1] Government of Pakistan, *Pakistan Economic Survey — Information Technology Chapter*, FY 2024–25. Available: https://www.finance.gov.pk/survey/chapter_25/15_Information_Technology.pdf
- [2] Stack Overflow, *2024 Developer Survey*. Available: <https://survey.stackoverflow.co/2024/>
- [3] GitHub, *Quantifying GitHub Copilot's Impact on Developer Productivity and Happiness*. Available: <https://github.blog/news-insights/research/research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>