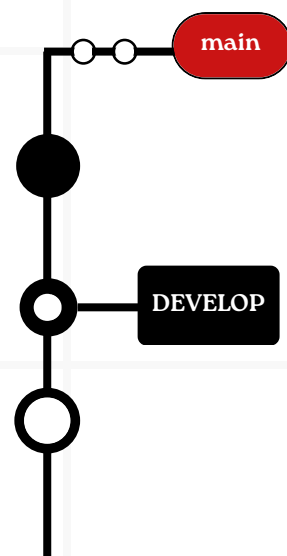


GIT

informed



UPDATES YOU'LL ACTUALLY WANT TO PULL

What's inside?

Section 01
\$ Beyond the tap

Section 02
\$ Code cracker

Section 03
\$ Creative 03

Section 04
\$ From the principal

```
$ git add
$ git commit -m "volume 5
out :)"
$ git push
```



Section 1

Beyond the Tap

By Rohan Amudhala



I wouldn't have written about NFC if it wasn't for three things that went wrong.

The first was sophomore year. I was building a contactless payment system for a college hackathon. Everyone was using Stripe, obvious choice, but I wanted to understand the layer underneath. I spent two weeks learning NFC. Not how to use an API. How it actually works. Built a proof-of-concept that could read MIFARE cards. It didn't win, but a judge told me the technical depth was impressive for someone who is trying to hack something together in 36 hours. That stuck with me.

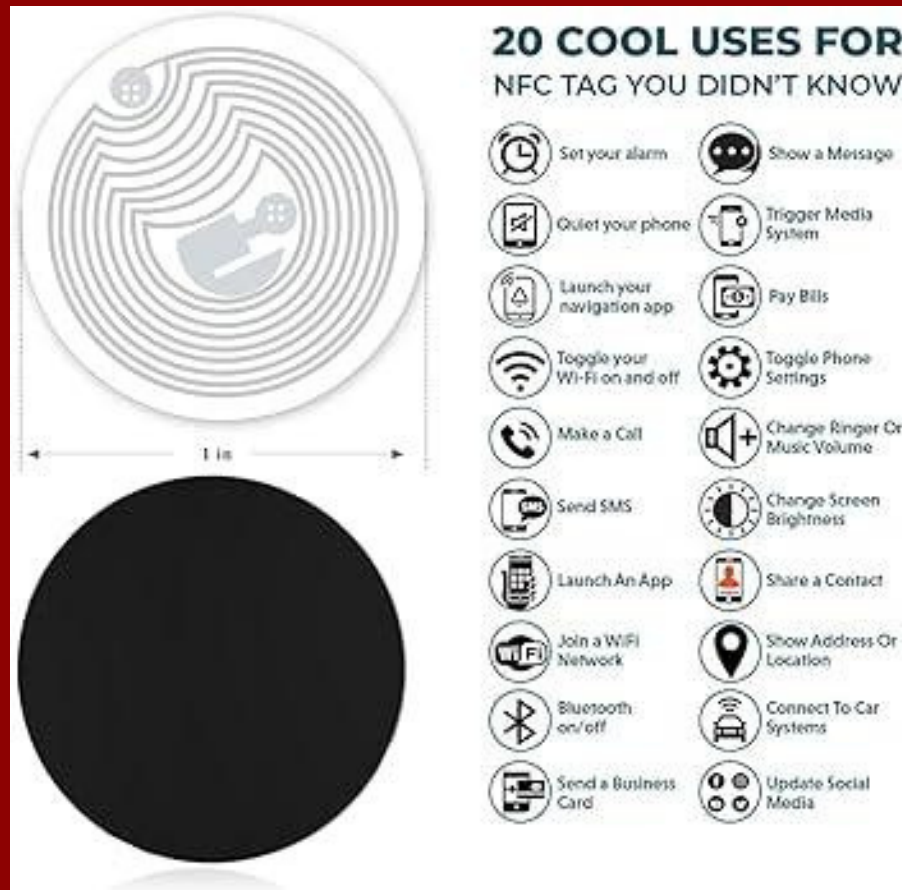
The second was when I realized I'd been using NFC every single day for two years without understanding it.

I tap my metro card. I tap to pay. I bump my phone to transfer files. It works. So I never questioned it. That bothered me more than it should have. Here I was, someone who claims to care about technology, and I was just consuming it, not understanding it.

The third is this article. Because I realized the only way I could actually write about NFC was to admit I'd been approaching it wrong the entire time.

NFC is Near Field Communication. Simply put, its two devices in close proximity, usually within 4 centimeters, exchange data wirelessly.

The technical specifications are straightforward. It operates at 13.56 MHz. The ISO/IEC standards are public. But knowing how it works and understanding why it matters are two very different things.



I spent the last couple of days digging into this. Reading research papers I didn't need to read. Looking at security vulnerabilities in NFC implementations. Trying to understand not just the mechanics, but the design philosophy behind a technology that chooses invisibility over user awareness.

Here's what changed for me. NFC works best when you don't think about it. That's intentional. Unlike QR codes, which demand your attention and require active participation, NFC is designed to be frictionless. You don't scan, you just tap. You don't read, you just bump. The technology disappears into the action. And in the cases where NFC is most prevalent, that invisibility is the entire point.



But invisibility has a cost. Most people have no idea how vulnerable their contactless cards are. There are tools available right now that can skim an NFC payment card from 10 meters away. The scary part is that it doesn't exist in theory, but it does in practice. And most users have no clue because the technology was designed for convenience, not transparency.

So I did what I usually do when something doesn't sit right with me. I went too deep. I spent hours understanding threat models. I learned about relay attacks. I found academic papers on NFC security that nobody outside the field reads. And I realized something uncomfortable. I knew more about NFC than most people who use it daily, and I still didn't know enough to actually protect myself.

That gap bothers me. It should bother everyone.

I could have written a surface-level explanation. Explaining the most important bulletins. But that felt dishonest. That's the kind of understanding I was running from. So instead I'm writing this. An admission that I went looking for clarity and found complexity instead.

The person reading this probably taps their phone every day without thinking about it. That's the technology working as intended. But maybe if you're like me and you start asking questions, you'll realize there's a lot more to know. Whether that's worth your time is up to you.

This is **Rohan**, signing off. I really hope this was more useful than just me explaining the obvious.

SECTION 02

✿ CRACK THE CODE ✿

By A.Ashritha

Answer this encrypted question to get to the final answer!

Find every word from the list below hidden in the grid — across or down. Every letter also carries its own cipher code from the key at the bottom. The same code answers appear in the ACROSS / DOWN clues, and the cipher key unlocks the final question at the end!

2 O	23 R	22 Y	2 O	2 O	13 Q	11 S	3 G	18 F	24 Z	13 Q	20 P	26 U	21 D	12 T
24 Z	18 F	21 D	2 O	7 J	21 D	5 E	14 C	23 R	24 Z	10 W	26 U	6 B	5 E	12 T
4 M	11 S	17 I	3 G	19 N	9 A	25 L	2 O	26 U	15 X	12 T	26 U	18 F	14 C	12 T
9 A	13 Q	14 C	6 B	6 B	12 T	3 G	18 F	8 H	12 T	9 A	22 Y	2 O	2 O	1 K
4 M	2 O	21 D	26 U	25 L	9 A	12 T	17 I	2 O	19 N	2 O	11 S	3 G	21 D	13 Q
8 H	26 U	7 J	20 P	9 A	6 B	16 V	23 R	14 C	2 O	26 U	17 I	19 N	17 I	23 R
14 C	10 W	17 I	1 K	22 Y	9 A	8 H	5 E	19 N	14 C	2 O	21 D	17 I	19 N	3 G
13 Q	7 J	9 A	14 C	11 S	11 S	22 Y	10 W	21 D	4 M	21 D	7 J	4 M	3 G	14 C
9 A	14 C	9 A	14 C	8 H	5 E	16 V	9 A	9 A	19 N	3 G	19 N	3 G	6 B	20 P
4 M	10 W	4 M	8 H	19 N	14 C	11 S	25 L	26 U	2 O	3 G	5 E	22 Y	16 V	17 I
1 K	14 C	7 J	9 A	1 K	9 A	20 P	25 L	9 A	17 I	19 N	12 T	5 E	15 X	12 T
19 N	22 Y	21 D	19 N	5 E	8 H	10 W	21 D	9 A	11 S	6 B	10 W	2 O	24 Z	20 P
1 K	5 E	23 R	19 N	5 E	25 L	18 F	16 V	23 R	5 E	3 G	2 O	2 O	13 Q	3 G
15 X	22 Y	5 E	5 E	19 N	26 U	4 M	21 D	4 M	19 N	3 G	23 R	9 A	17 I	24 Z
11 S	7 J	9 A	25 L	3 G	18 F	4 M	12 T	20 P	9 A	14 C	1 K	5 E	12 T	26 U

Clues to the crack the code



Down

1. Converting encoded data back to its original form (8 letters) — **1st-letter code: 21**
2. Organized, structured collection of data (8 letters) — **1st-letter code: 21**
4. Security system that filters incoming/outgoing network traffic (8 letters) — **1st-letter code: 18**
8. Medium over which a signal or data travels (7 letters) — **1st-letter code: 14**
9. Unwanted random disturbance that corrupts a signal (5 letters) — **1st-letter code: 19**
10. Interconnected system of computers or devices (7 letters) — **1st-letter code: 19**

Across

3. A time-varying quantity that conveys information (6 letters) — **1st-letter code: 11**
5. Varying a carrier wave's properties to encode information (10 letters) — **1st-letter code: 4**
6. Converting data into a specific format for transmission or storage (8 letters) — **1st-letter code: 5**
7. Fast memory layer that stores frequently accessed data (5 letters) — **1st-letter code: 14**
11. Readable message before it is encrypted (9 letters) — **1st-letter code: 20**
12. Core part of an operating system that manages hardware (6 letters) — **1st-letter code: 1**
13. A formatted unit of data sent over a network (6 letters) — **1st-letter code: 20**

✿ CRACK THE CODE ✿

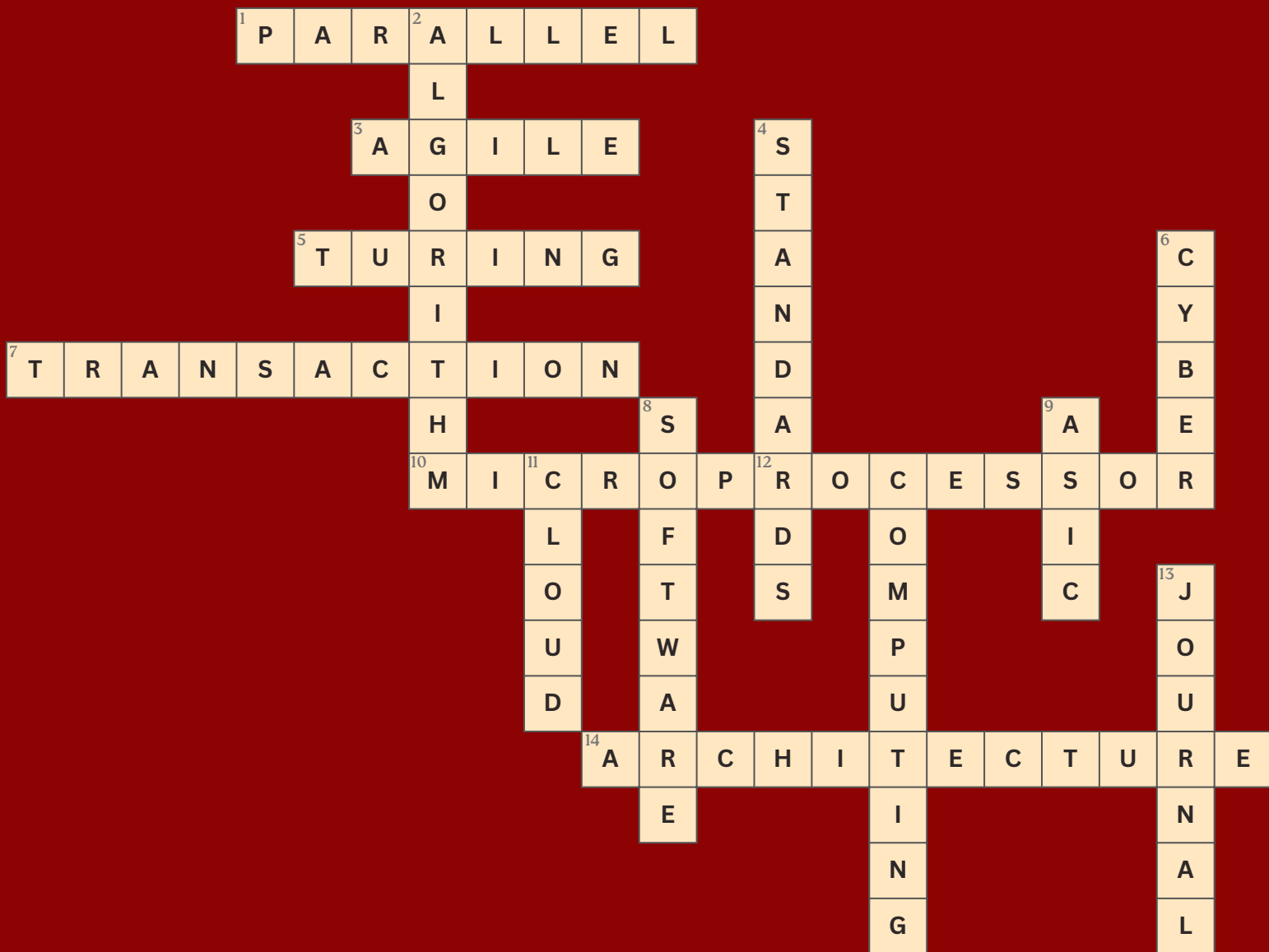
Answer this encrypted question to get to the final answer!

F 9 12 8 5 23 2 18 17 19 18 2 23 4 9 12 17 2 19 12 8 5 2 23 Y ?

¹ K	²	³	⁴	⁵	⁶	⁷	⁸	⁹	¹⁰	¹¹	¹²	¹³
¹⁴	¹⁵	¹⁶	¹⁷	¹⁸	¹⁹ N	²⁰	²¹	²²	²³ R	²⁴	²⁵ L	²⁶

Edition 4

CROSSWORD SOLUTION



Section 3

When Your Code Writes Itself, Who Holds the Architecture?

By Shivappasad



Software development is no longer confined to developers sitting behind screens and manually writing every line of code. Today, everything from smart devices to physical infrastructure is connected through APIs, while modern Integrated Development Environments (IDEs) constantly suggest automated improvements and refactoring decisions. We have entered the era of Agentic AI, where software tools have evolved beyond passive assistants into autonomous systems capable of reasoning, planning, and executing tasks independently. What was once a simple text editor has transformed into a platform capable of hosting multiple AI agents that can analyze entire codebases, launch isolated testing environments, and deploy small fixes before developers even notice a problem. The traditional debugging session is no longer a conversation between a programmer and a compiler; it now includes an intelligent digital collaborator that observes, analyzes, and commits changes directly into shared repositories.

The fundamental idea behind agentic engineering is a shift in responsibility—from writing code manually to defining intent and overseeing execution. The promise is compelling: applications that assemble themselves, infrastructure that scales automatically in response to demand, and systems that can identify and repair their own failures. In a strange twist of technological progress, developers are increasingly finding themselves managing software that writes software, a career path that nobody's computer science professor really prepared them for. Yet every layer of automation introduces new risks. An AI agent generating an API controller or recommending a database optimization may seem highly productive, but it often operates with limited transparency. These systems can modify architectural foundations, influence design decisions, and potentially expose proprietary business logic to external cloud-based platforms used for model improvement. Beneath the appeal of frictionless development lies a collection of some of computer science's most difficult challenges, including zero-trust security, the economics of large-scale inference, and maintaining predictable system behavior.

Among these challenges, security remains the most significant obstacle. Autonomous agents frequently generate and execute code dynamically, often operating beyond traditional repository trust boundaries.

This creates opportunities for emerging threats such as prompt-injection attacks and malicious dependency manipulation. A single compromised package or hallucinated code suggestion could propagate through a continuous integration pipeline and eventually reach production systems. Scalability introduces another concern. Sustaining real-time reasoning across multiple AI agents requires substantial computational resources, pushing organizations to build efficient hybrid-cloud infrastructures that balance performance, latency, and operational costs. It turns out that replacing developers with AI does not eliminate expenses; it simply changes the monthly budget from coffee and snacks to GPUs and inference clusters. Equally problematic is the issue of determinism. Unlike traditional software, AI systems rely on probabilistic outputs rather than fixed logic. As a result, enterprise environments can begin to behave like opaque systems where interacting agents may enter endless execution cycles, generate conflicting decisions, or fail to align on governance and compliance requirements.

When viewed from a broader perspective, these same concepts are already shaping the future of enterprise technology.

Continuous delivery platforms, cloud orchestration systems, large-scale data pipelines, and automated security frameworks increasingly depend on agentic architectures to manage complex and rapidly changing environments. Successfully operating these ecosystems requires strong software knowledge graphs, rigorous provenance validation, and deterministic safeguards capable of keeping pace with machine-speed decision-making. However, every autonomous action introduces another potential point of failure. The challenge is no longer simply creating AI systems that work; it is designing agentic infrastructures that remain predictable, secure, and interoperable within human-led environments. In a world where one flawed automated decision can ripple through an entire organization, protecting system integrity becomes just as important as achieving automation itself.

SECTION 04

From the Principal

Dr. S. Radha
Principal, SSN College of Engineering

About Dr. S. Radha

Dr. S. Radha is the Principal of SSN College of Engineering and a Senior Professor in the Department of Electronics and Communication Engineering. With over 33 years of teaching experience and 24 years in research, her work focuses on Wireless Sensor Networks, IoT, and sensor technologies. She was a visiting researcher at Carnegie Mellon University. Her contributions include 183 publications, nine granted patents, and 12 externally funded projects. As an active senior member of IEEE, she currently serves as Treasurer of the IEEE Madras Section and as Secretary of the IEEE ComSoc Madras Chapter.



1) Many students join IEEE for the certificate or since it adds to their resume, but don't really engage after that. Having been actively involved as a senior member with IEEE for decades and holding an incredibly high position, what's something IEEE gave you that a certificate never could?

To me, IEEE has always been much more than an organization that simply provides certificates to students. I've seen students from several colleges excel and go on to hold leadership positions in IEEE as young professionals.

Being in IEEE, my membership is now 26 years old. I was also a student member once. I renewed my IEEE membership when I went on to become a PhD scholar. That's when I realized that I wanted to reconnect with IEEE. What IEEE gave me was a lot of opportunities to collaborate and to network with people working in similar domains and learn from them. From the early 2000s, I was able to attend a lot of conferences because of that membership privilege, learn from people, connect, and interact with them.

Although resources were scarce at the time, my IEEE membership granted me access to their documents upon request. But now, as a leader associated with the Madras Section, and globally in the IEEE Computer Society, I'm learning a lot. There's scope for learning everywhere, and it gives me a new dimension.

I may be holding an administrative position here, but when I interact with senior professionals who hold positions much higher than mine, I gain a different perspective. While I'm working with a relatively small community here, they are leading and collaborating with people across the globe. As a result, their outlook and way of thinking are often much broader and different. That gives me new information to take in as a leader, from a leader. So this is how I view IEEE. I can say I've been inspired by seeing many people.

Currently, the IEEE President – Mary Ellen Randall, who has also been working at IBM for a long time, travels around the globe frequently. Watching her successfully juggle her corporate role, a startup, and her family proves that a healthy work-life integration is achievable. Observing her drive inspires me to push my own boundaries and reach higher goals within IEEE.

2) You've guided numerous PhD scholars and published over 180 research papers, achievements that may seem daunting to many students. For someone just beginning their research journey, where does it all start?

Research doesn't just come to you. It begins with an inherent curiosity and a genuine passion for exploring, discovering, and creating something new. You need to learn something new, understand what's already available in the literature, identify the gaps, and then think about how those gaps can be addressed.

Only when you have some passion towards that, will you truly get it. I had this passion because when I did my BE, which at that time I believed was the highest degree, it was enough for me to get a job. But when I started working, I saw that coworkers with master's degrees had a much deeper understanding of the field than I did. That changed my view. I thought getting a higher degree would help me learn better and know more. But it didn't really click for me until my PhD. After completing it, my perception in learning and grasping information was entirely different.

So the learning and understanding capability I had with a BE degree was significantly enhanced after my PhD. I never thought that doing a PhD would mean I'd go on to guide so many PhD scholars or publish so many papers. It just came along the way. I started guiding people who wanted to associate with me and my work. Also, our college gave me a good platform to excel in research, administration, and academics, thanks to the management. Otherwise, balancing all this in a private institution wouldn't be possible.

I truly want to thank our management, because without their support it wouldn't be possible. The resources needed for a PhD has been given by our SSN Management, which I don't think is something many colleges give.

Our management has also supported PhD scholars with fellowships and faculty members who aspired to pursue higher studies. PhDs were sponsored by them as well. All of this enabled me to guide more PhD scholars. I have a passion for learning, and whoever comes to me, I never force them to choose a topic. If one is interested in doing research in a particular domain, I can give a problem statement. If you want to do something of your own interest, you can also do it under me, and I'll guide you and show you the right direction.

So the scholars who have been associated with me so far were confident that I would see them through to completion. Nobody has discontinued under me so far. I've seen many students discontinue their PhDs for various reasons. Even when someone wants to discontinue, I try to advise them to take a longer break instead, even six months or a year if needed, but to come back and complete it. That's how many people have completed their PhDs under me.

The first thing is passion towards learning, if that passion for learning is there, everything else follows. I'm still learning. Wherever I go, I ask my students, "What is the takeaway?" When we listen to an expert, what's your takeaway? That's the thumb rule I try to instill, whether it's a conference, a workshop, or listening to eminent speakers. That's how I try to guide my students, they listen, analyze, and eventually arrive at a solid problem statement to work on.

3) Students often aspire to pursue higher studies, research, or careers abroad. Based on your experience at Carnegie Mellon, what qualities or experiences do you believe help students stand out on a global stage?

One thing I observed when I got the opportunity to go to Carnegie Mellon in 2010, after having joined SSN in 2003, was the work culture among students. The first thing I noticed was that students in foreign universities were working for more than 18 hours a day during the weekdays. Weekends they'd relax and enjoy, but Monday to Friday they worked hard. I think that's the culture that's missing here. We tend to work only from 8 to 5 here, but there, I saw students working like that, and I found myself pushed to work that way too, because once you're there, you want to be part of it, like everyone else.

I felt that if we could create that kind of environment here, more people would choose higher education over going straight into a job. A job is needed for some people, not everyone can go directly into higher education. But for those who want to pursue higher education, there are a lot of opportunities, with industry professionals coming in to participate in the curriculum and setting up lab facilities. A lot of real problem statements are available to work on, and you get to work in the actual core domain of whatever field you choose for your higher education.

Those are the kinds of opportunities you get there compared to here. I think a lot of it comes down to your peer group. If everyone around you is moving towards a job, you'll move that way too. If everyone is moving towards learning, you'll move that way. So whatever you like, choose that and move forward.

When it comes to higher education, that choice should be yours, nobody should force you. A basic degree is something our parents guide us towards based on our interests, but beyond that, the individual has to choose and move forward. Only then can you succeed.

4) Many students participate in various student led clubs from organisations like IEEE, ACM and many more. Beyond technical skills and academics, what do you think students gain from taking on active roles in these communities?

The first thing is being Master of Ceremonies. Wherever IEEE events are organized, students come forward to make it happen, and that itself gives you confidence. You're seeing a big, diverse crowd bringing together students as well as professionals, fellows, and celebrated IEEE members.

As a youngster, when you're given the confidence and responsibility to organize something, such as serving as the Master of Ceremonies, it becomes a valuable learning experience. Even when an event is organized by a section or an Organizational Unit, students are often brought in to run it and support various aspects of the event through voluntary contributions. You end up connecting with people of different age groups, standing in front of them, talking to them, and that builds your confidence.

When students are allowed to organize and coordinate a team, they learn how to be part of a team and how to run a flawless event. There might be minor hiccups initially, but they learn from them and correct them the next time.

For example, there might be a delay in starting an event. Abroad, when an event is scheduled for 8:00, it starts sharply at 8:00, not a minute later. Here, when you're just starting out, an event might start at 8:05, but from the next event onward, you start correcting it, because senior people and the audience have all arrived on time. That's a learning experience in itself, and team spirit is something you simply can't learn from a classroom.

In the classroom, you're organizing with the same group of people. But when you step outside into IEEE, you see people from different backgrounds, groups, and cultures, and that gives you a different kind of learning. It improves how you organize and connect with people, and also helps in understanding their needs and what needs to be done. That's a different experience altogether.

I've seen many people who have come from other colleges and are on the Board of Governors of various IEEE societies. How did they get there? By organizing small events, then medium ones, then big ones, and eventually reaching a position where they decide where and how events can be organized.

They possess the sanctioning authority now. These are things you learn once you step outside the classroom, and this forum gives you that.

Like ACM, IEEE, and IET, there are so many professional societies out there. Whenever you get an opportunity, grab it and move forward. These professional bodies give you a different flavor, and a different kind of learning altogether.

5) With so many opportunities and career paths available today, it can be very overwhelming, and difficult for students to know where to focus their time and energy. What guidance would you offer to those who are still exploring their interests and future goals?

The first thing is to ask your heart what you want to do. You'll always be influenced by your surroundings, your parents or your peer group, but you also have to ask yourself what you truly want. If you think about what you want to do and act accordingly, you'll end up satisfied. So decide for yourself, because UG is generally a broad spectrum of learning. Nobody can say there's one correct path to choose at this stage. Maybe some people succeed in choosing correctly right away, but for the majority, it's a broad base learning platform.

In a classroom, a student might not be good at programming but chooses computer science because their parents pushed them towards it based on today's job market, telling them, "Choose computer science, and after four years, you'll get a job."

Whether that student finds it interesting or not is a question mark. Only after you learn will you come to know your real interest.

That clarity often comes only once you reach higher semesters, or get into the industry, or pursue an internship, where you learn what industries are actually looking for, and whether that kind of work suits you.

Even as a teacher, I may not get to teach the course I wish to initially. Whatever the department offers, I have to teach.

But once I reach a certain level, I can choose according to my interest and teach accordingly. It's like that everywhere. At the entry level, you may not find satisfaction, but as you move up, you find your interest. So keep searching in that direction, and you'll continue on that path. My suggestion to students is to ask your heart, find your passion and interest, and move accordingly, instead of searching through others' opinions. You can listen to those who've succeeded in life, not necessarily to follow their words exactly, but to analyze it yourself and decide accordingly.

6) The technologies students learn today may look very different a decade from now. What skills or qualities do you believe will help students remain adaptable and succeed in the long run?

If you look at our program outcomes, lifelong learning is one of them. There are about 11 graduate attributes, the first being core domain learning, problem solving skills, and so on, but ultimately, lifelong learning is one of the outcomes. As an engineer, or in any profession, lifelong learning is essential. Only then can we keep up with rapidly evolving technology and face whatever challenges come our way. Thirty five years ago, when we learned computer science, we didn't even know what a computer really was, we just knew a CPU was a box.

Today, computer science is something else entirely. We learned all of this along the way, as we grew in our profession and as technology grew with us. So we have to keep learning.

For example, we were talking about 5G, and now people have already started R&D activity on 6G. So now we need to focus on what 6G is, where the difference lies, and why it's called what it's called. There's a lot of material available, a lot of online lectures, and we need to find the time to upgrade ourselves. Otherwise, it becomes very difficult to keep up with how things are moving. Being an engineer, being any kind of professional, means always being a learner. Otherwise we become outdated and we won't fit the market's demands. That's why, right from when you enter the course, the program outcomes clearly state that lifelong learning is one of them.

Even the faculty currently in your department, I don't think many of them formally learned ML or deep learning, but they're still able to teach you and give you problem statements to work on. How? By learning from others, going through papers, because we have that capability to learn easily. The only requirement is that we keep learning new things; it's essential for everyone who wants to succeed in this market.

I never want to reach a point where I say I've learned everything and I'm done learning. Because then we become outdated from society, or our community, I'd say.

COMING SOON

Interview by Anni Priscilla

Former IEEE SB Vice Chairperson

GLIMPSE OF THE INTERVIEW

In our next issue, we sit down with **SSN alumna Anni Priscilla, former IEEE SB Vice Chair at SSN** and now a Data Analyst at Bill Trust, for a candid conversation on breaking into analytics, using AI to work smarter, and navigating imposter syndrome early in your career.

A sneak peek at what's ahead:

"That's what companies expect too, that's why they want employees using AI, so progress happens at a faster rate."

"If you don't get the job you wanted, that doesn't define you. Always remember you're capable of much more, and if the best hasn't arrived yet, it's still to come. So relax, take time for yourself, and enjoy the moment. I know that's easier said than done during this season, but as your senior and alumni, that's my advice to you.."

Coming soon! Stay Tuned..

★ MEET THE TEAM ★

EDITORIAL



Isha V



Shivapprasad



Ashritha



Akkshay



Rohan

DESIGN



Harshita B