



git informed

UPDATES YOU'LL ACTUALLY WANT TO PULL



What's inside
this issue?

SECTION 1:

Love clicking tractors
@2 a.m.

SECTION 2:

You had me at "Hello
World" - An Ode to
Code

SECTION 3:

Code Confessions

SECTION 4:

From the Alumni



Section 1

Love Clicking Tractors at 2 AM

-By Sanjana Y

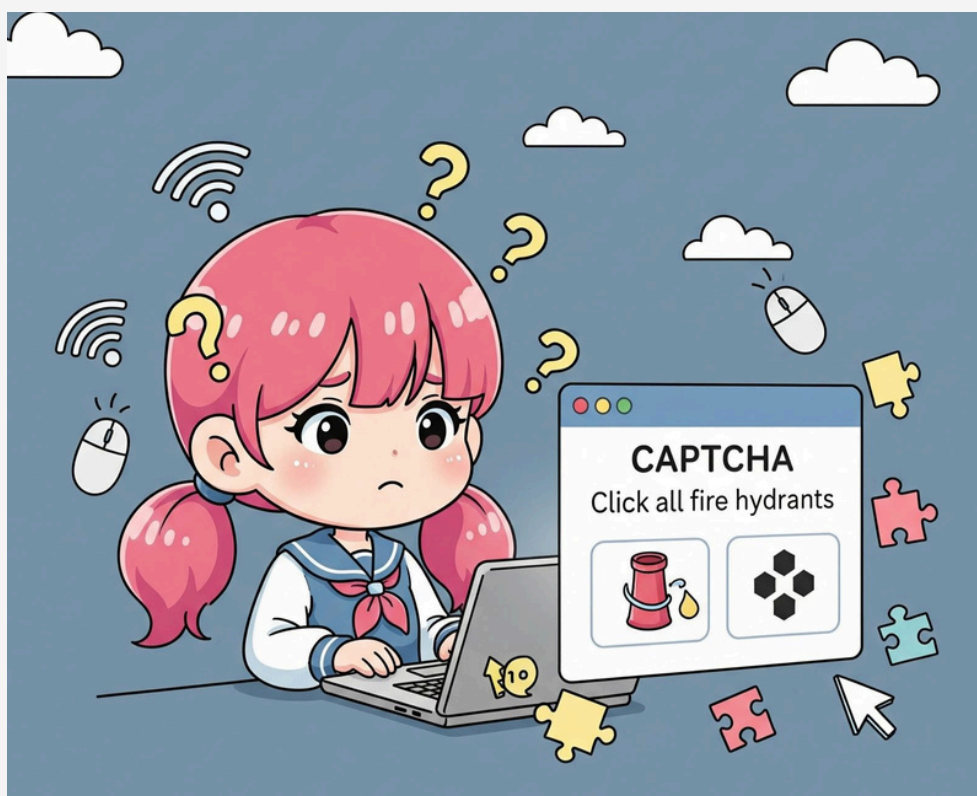


Today, I cried. Not because my father told me he was proud of me. But because a website called me, 'HUMAN'. CAPTCHA's funny alright, if only also a nuisance with its ideology: 'One does not simply access the website'. How many times have we found ourselves squinting at blurry crosswalks, second-guessing traffic lights, and clicking tiny boxes like our very existence depends on it? All for a CAPTCHA test to whisper, "Not a bot."

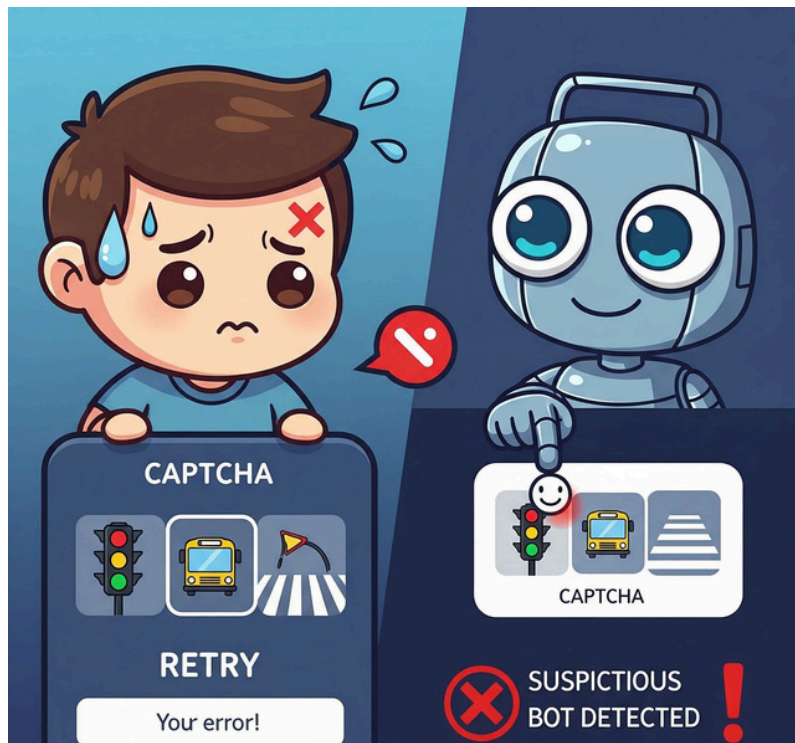
It's humbling, really and sometimes infuriating to have our humanity validated not by society, not by art, not by love, not even by that one judgemental paternal relative. But by an algorithm trained to filter out bots who are, honestly, sometimes just faster than us.

In other words, CAPTCHA which stands for Completely Automated Public Turing test to tell Computers and Humans Apart is just a fancy way of saying, are you slow? Are you imperfect? If yes, off you go human. Else, prepare to be locked out.

So, how does a bot detect bots? Surely not out of friendship. CAPTCHA isn't just a gatekeeper, it's a full-blown behavioral analyst. They silently monitor your mouse movements, typing speed, click rhythm, and even the time you take to make decisions. The same way your dad watched you assemble IKEA furniture without helping, waiting for you to give up. But hey, at least CAPTCHA benevolently offers you a second chance. Same cannot be said for the former, I fear.



Bots, on the one hand, move with cold precision.
No hesitation. No fumbles. In contrast, humans are messy.
We overshoot buttons. We hover too long.
We miss a corner. We're... flawed.
And that's exactly what CAPTCHA celebrates.
For once in your life, your sheer existence as a human being is
just enough. CAPTCHA does NOT expect you to crack JEE.



If your behavior seems “human-ish” and you get the answer wrong, you usually get another chance. But if your movements feel “bot-like” and you get it wrong? You’re likely locked out. Even if you do get it right, moving like a bot can lower your “human score” on Google’s invisible CAPTCHA scale — which means you might get served another, even harder challenge, just to prove you weren’t cheating. "There are no winners in this game, only survivors", takes a whole new meaning with CAPTCHA.

For humans, the 'survival' rates generally hover around 85–95% depending on the CAPTCHA type. But bots are catching up. Some now have success rates between 50–85%, using a variety of questionable tactics. Machine learning image classifiers let them identify fire hydrants better than we can. Headless browsers mimic human behaviors like scrolling and clicking. And if all else fails, they outsource their humanity - literally. Entire services exist that pay real humans pennies to solve CAPTCHAs in real time. Yes, bots hire us now. That’s where we are.

It's oddly poetic, isn't it? I spent 19 years waiting for fatherly approval, only for CAPTCHA to give it to me in 3 seconds every time I surf the web. Sure, I still had to redo the test because I missed a sliver of the traffic light to the square on the left, but believe me when I say: "Love clicking tractors @2 a.m.

Section 2

You had me at "Hello World" *- An Ode to Code*

By Dona Merlin Rony



It all began with the prompt on the screen,
Those were the times during the quite quarantine,
With staggered typing of
`print ("Hello World")`
A timid tap - and then the code unfurled

And there it began with `ifs` and `else's`,
Nested thoughts and logic pulses.
Infinite loops and missed semicolons,
Errors that echoed like quiet violins.

Stacks of worries, queues of doubt,
Graphs and trees that branched their way out.

Heaps of pressure I tried to sort,
Merge, split or search - my only resort

From Python to C, React to Node,
Each traversal, I sailed through
nothing but code.

Hackathons fueled by caffeine and flair,
Pull requests flying through midnight air.

Projects piled up like unsaved tabs
Deadlines, dashboards, and sleepless labs.
I solved their tests in silent bounds,
My worth condensed to $O(n)$ sounds.

But then it ran - no bugs to diagnose,
Just clean outputs in structured rows.
The tests all passed, no edge case whined,
And for a moment, all felt well-defined.

The syntax changed, the IDEs too,
But the spark remained in something true.

Through stacks and strings and lines I hurled-
'You had me at Hello World.'

The poem ends where many journeys begin with a flicker of curiosity, a simple program, and the unexpected joy of watching something come to life on screen. Yet beyond that first moment of success, things quickly change. The simplicity of `print("Hello World")` gives way to a vast landscape of tools, frameworks, expectations, and uncertainty about what to learn next.

As learning deepens, the pace accelerates. Conversations shift to web development, machine learning, app development, DevOps, cloud computing, and artificial intelligence. Full-stack projects have become common. Contributions to open-source repositories, competitive programming, and the rush for internships begin to define the academic rhythm. And in the midst of all this, it's easy to feel pressure, as though mastering everything is the only way forward.

But technical growth doesn't require expertise in every domain. It begins by exploring areas that genuinely resonate with the learner. For some, that means building user interfaces with tools like React or Figma. Others are drawn to backend development, where APIs, database design, and performance tuning come into focus. For those curious about data, Python libraries such as NumPy, Pandas, and scikit-learn often lead to a deeper interest in data science or machine learning.

One of the most significant shifts in recent times is the rapid integration of AI and automation into mainstream development. Tools like GitHub Copilot, ChatGPT, and Replit AI are no longer just educational aids; they are actively transforming how professionals write, test, and architect software. In this environment, success is becoming less about memorising syntax and more about applying, adapting, and collaborating with evolving technologies.

As projects grow in complexity, so does familiarity with the development process. Concepts like RESTful APIs, cloud deployment, Git version control, and debugging workflows become essential. Mistakes are part of the journey, and with each challenge comes a deeper understanding of how systems function in the real world.

While AI and large language models dominate headlines, chasing trends alone rarely leads to meaningful growth. The key lies in going beyond surface-level tutorials by building tangible projects, solving real problems, and engaging with the underlying principles. A solid foundation in one domain often proves more valuable than superficial familiarity with many, as the saying goes – a jack of all trades is a master of none.

Eventually, most learners discover the area of tech that just feels right, the kind of work that draws you in, even when it's hard. And once that happens, lean into it. Because, no matter how much the industry evolves, that initial spark—the one behind your very first successful line of code is what keeps the journey worthwhile.



Section 3

Code Confessions

-By Shivani



Hi there! Welcome to the pilot run of our brand-new column, Code Confessions! Here, you can expect to hear about people's mishaps in their learning journey and how they overcame them. Don't worry—we'll give you all the context you need to follow along. Stories like these are a great way to become less intimidated by unfamiliar tech (and unfamiliar people) !

Today, we dive into an incident that happened to one of our Tech team heads, Pranav (he says hi!).

Linux is a great operating system to start tinkering on. It allows for greater control and learning opportunities, has a strong online community you can rely on for support (technical or emotional), and—most importantly—is not stuck behind a paywall! Depending on the distro (short for “distribution”—a pre-packaged version of the Linux OS), the learning curve isn't too steep either.

One day, Pranav, our resident Linux fan, was playing around with his OS. He was trying to understand `usermod` , a command that helps you control privileges and properties of the users on your system. As you might have guessed, this requires admin/root privileges, which means he had to be part of the `sudo` group. Being in `sudo` gives you privileges on par with root, but in a much safer and more controlled way.

While trying to use the `append-group` command:

```
sudo usermod -aG new_group user
```

—which would add “user” to the group “new_group” in addition to all their existing groups—he typed this instead:

```
sudo usermod -G new_group user
```


This command added “user” to “new_group” but removed them from all their previous groups in the process.

As you can imagine, this was a problem. He was now locked out of his system. There was no one else in the sudo group, meaning “user” couldn’t be added back directly. He couldn’t access root either, because access is disabled in most distros for security reasons, as it was in this case as well.

Normally, cue panic. But not for Pranav—he’d been here before and knew what to do. He used a live USB, where he tried to change root from the USB to his hardware, and give himself admin access again. Once rebooted, everything should be fine. This had worked a thousand times before.

He rebooted—and the system crashed. Literally.

Somehow, the hack that worked before, that should have worked now as well, chose this opportune moment to fail. The issue was the distro of Linux he was using. He was working on a Fedora setup, instead of his usual Arch setup. Fedora is more polished and user-friendly right off the bat. Arch, on the other hand, gives you full control and expects you to know what you’re doing. That difference, especially in how system recovery tools are structured, ended up being crucial.

He searched high and low for a solution. Finally, he ended up having to reinstall the OS.

His words of wisdom? Always read the documentation carefully and thoroughly, especially before running commands that affect user access.

TL;DR: I locked myself out of admin permissions once in a Linux server, and I had to reinstall the entire operating system. All of this could have been prevented by reading documentation.

FROM THE *Alumni*



*"Don't wait to be
'ready' before
participating—jump
into contests, learn
from the
experience, and
improve with every
attempt"*

Here's what we asked Naren Sivakumar:

1. Could you briefly introduce yourself and share what you're currently working on in AI?

I'm Naren Sivakumar, a PhD student at the University of Maryland. I'm working on benchmarking AI in all their unique shapes and forms, across various tasks. This includes LLMs, VLMs, and multi-modal LLMs for now.

2. What excites you most about your current work or research, and what inspired you to focus on large language models?

What excites me the most is to correctly quantify what these models can do. There's a lot of fluff and hype surrounding these models, and their capabilities mean you can test a LOT of things out. These tests also do not rely on the traditional framework of metrics, so it requires a lot of out of the box thinking!

3. Looking back, was there a specific moment or project that confirmed your decision to pursue a PhD in this field?

Ah, I've always been super into research. Even at SSN, when COVID hit, a few friends and I got together and enrolled in a competition to determine the best model for detecting plant species, mentored by Dr. Prabavathy and Dr Lekshmi. They pushed us to do more than just the ask, and we stood fifth in the competition worldwide! Since then, I've participated in quite a few hackathons, competitions and research endeavors. My advisor during my master's thesis (Dr. Lara Martin) encouraged me to pursue a PhD throughout my master's as well, so I must thank her for her guidance and ceaseless support.

4. For those new to LLMs and RAG, what foundational skills would you recommend, and what advice do you have for choosing between research and industry roles?

Full disclosure, I'm not a person who obeys the rules very well. If I see something I like, I just dive right in, and learn relevant topics through googling, LLMs and other tools on the way. My only advice is that there isn't really a bar for entry, just jump in with some project you think you'd like to work on and learn on the go! With respect to research versus industry, I can only speak for the research side of things. Research is something that serves you a lot of ups and downs, rejections, and failures. Despite this, the thrill of working on something cutting edge that could influence the way the world looks at things is often enough (at least for me) to overcome these setbacks. It's a long road, but is full of discoveries and thrills!

Editorial



Dona Merlin Rony



Sanjana Y



Shivani Garimella



Sowkya Bellana



Akshayalakshmi

Design



Shriya Alladi



Hemavathy T E