NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Finite Loop
Inspire the Next

INFINITE INSIDER

10 YEAR ANNIVERSARY EDITION 2.0

Rest with REST APIs
Pg 30

Coding with 6th Sense

Nothing to Something
Pg 44

Open Source? I Hardly Know Her
Pg 49

PREFACE

As we present the second edition of inFinite Insider, we are filled with a sense of joy, pride, and purpose. What began last year as a bold step into something new has now become a part of who we are. This magazine stands as a reflection of how far we have come and how much we continue to grow as a club and as a community.

Finite Loop Club has always been a place where curiosity leads to creation. Since 2016, it has welcomed students who are eager to learn, to build, and to share their ideas with the world. With each passing year, this spirit has only grown stronger. inFinite Insider is our way of capturing that spirit and sharing it with everyone who believes in the power of technology and teamwork. Our motto, Inspire the Rest, is not just a phrase we say. It is the reason we stay motivated. It pushes us to think beyond the ordinary and to help others do the same. Through this edition, we hope to spark that same feeling in you. Whether it is a project that surprises you, an idea that challenges you, or a story that moves you, we want these pages to leave you inspired.

This magazine is a celebration of effort and imagination. It is for the students who stay late to perfect their code. It is for the teams who push boundaries during our events. It is for the thinkers, the doers, and the quiet builders who keep learning without needing the spotlight.

Every voice matters, and every contribution shapes who we are becoming. We are truly grateful to our Principal, Dr. Niranjan N. Chiplunkar, Vice Principal, Dr. Nagesh Prabhu, Dean (Academics), Dr. I. Ramesh Mithanthaya, Controller of Examinations, Dr. Shrinivasa Rao B.R., and Dean-Student Welfare, Dr. Narasimha K. Bailkeri. Their continued encouragement, both thoughtful and financial, has given us the confidence to take on new challenges and bring our ideas to life. We also thank Dr. Shashank Shetty and Dr. Puneeth R.P. for being the backbone of our journey. Their belief in us has helped the club grow into something we are all proud to be part of.

To every member of Finite Loop Club, this edition is a result of your time, your passion, and your belief in what we do together. Thank you for making this club a place where ideas turn into impact. We hope this magazine gives you a glimpse of what is possible when students come together with a shared purpose. There is more ahead, and we are just getting started.

With heartfelt thanks and hopeful energy ,
The Editorial Team of inFinite Insider
Finite Loop Club

INSIDE THE LOOP

1.	Preface	i
2.	Inside the Loop	00
3.	Editorial	01
4.	Message	02
5.	Introduction	06
6.	Core Team 2024-25 and 2025-26	07
7.	Events	14
8.	Articles	20
9.	Our Projects	53
10.	Achievements	55
11.	Conclusion	58

EDITORIAL

Dear tech enthusiasts, innovators, and curious minds, I'm Akhil Manoj, and I'm thrilled to present this edition of inFinite Insider, a magazine that captures what makes the Finite Loop Club truly special.

Back in 2016, our faculty coordinators Dr. Shashank Shetty, Dr. Puneeth and Prof. Krishna Prasad Rao, had a vision that went beyond just another college club. What started with a small group of students passionate about technology has grown into one of NMAMIT's most recognized technical communities. It's remarkable how that initial spark continues to inspire new generations of students.

Being chosen as editor has been both overwhelming and incredibly rewarding. This magazine exists because of the countless contributions from our core committee members throughout the years; from those early pioneers in 2016 to our current team in 2025–26. Their passion is genuinely contagious, and working with them reminds me why I love this community. What you'll find in these pages isn't just our projects and achievements. This is about the late-night coding breakthroughs, the collaborative energy during hackathons, the satisfaction of solving that stubborn bug, and the friendships forged over shared challenges. It's about transforming ideas into working solutions and discovering that learning never stops; it just gets more interesting.

Our current Core Team 2025–26 has poured their hearts into making this edition special. Their creativity and determination have brought these stories to life, proving that everyone brings something different to the table while sharing that fundamental curiosity about how things work.

Technology moves fast, but that's exactly what makes FLC valuable. We're not just learning frameworks or languages; we're developing the mindset to adapt, innovate, and solve problems that don't even exist yet. Whether you're debugging your first program or architecting complex systems, you're part of a community that believes in pushing boundaries together.

To everyone who's been part of FLC's journey; thank you for creating something worth celebrating. To those discovering us for the first time—welcome to a community where your ideas have room to grow and your contributions matter. The loop may be finite in name, but the possibilities within it are anything but limited.

Akhil Manoj
Editor, inFinite Insider
Secretary, Finite Loop Club
Academic Year 2025–26

MESSAGE

Dr. Niranjan N.Chiplunkar
Principal

I am very happy to note the very vibrant activities of the Finite Loop Club (FLC). I am aware about various student-led workshops, coding events, hackathons, etc., which were conducted during the previous academic year through FLC. A sizable number of student volunteers of this club, under the able leadership of faculty coordinators, have organized seminars in the latest trending topics of Generative AI and its applications. I am sure that many students have utilized this forum for learning new things from industry experts.

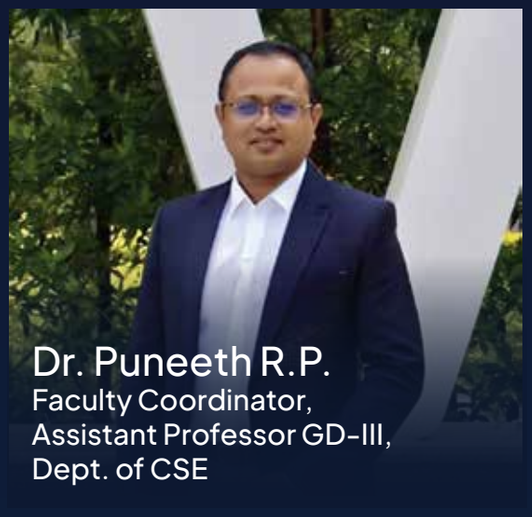
My hearty congratulations to all student members of FLC and I wish a productive and eventful year ahead.

MESSAGE

FACULTY COORDINATORS

Dr. Shashank Shetty
Faculty Coordinator,
Associate Professor,
Dept. of CSE

I am delighted to see the Finite Loop Club (FLC) launch the second edition of its magazine, Infinite Insider 2.0, which showcases the creative and technical contributions of its members through insightful blogs, along with highlights of the club's events and activities. Having been part of FLC since its inception, it is truly inspiring to witness its remarkable growth and the relentless efforts of the team in creating a strong platform for NMAMIT students to enhance their coding and problem-solving skills. From conducting technical workshops on the latest industry-relevant tech stacks, organizing month-long hackathons like HackLoop, and hosting weekly DSA sprints, to fostering active engagement through Discord, the club has consistently delivered quality initiatives. The flagship national-level hackathon, Hackfest, has become a standout event, attracting participants from across the country and providing a unique space for innovation and collaboration. In addition, the club has made significant contributions to real-time project development, including Yakshagavishti and the Incridea Fest Web Application, which transformed the event into a paperless and efficient experience with a dedicated jury portal. What makes FLC truly special is its vibrant culture of peer-to-peer learning, where seniors, juniors, and alumni collaborate, mentor, and inspire each other, creating an ecosystem that nurtures technical excellence and community spirit. I am proud to have been part of this journey and look forward to seeing the club reach even greater heights.



Dr. Puneeth R.P.
Faculty Coordinator,
Assistant Professor GD-III,
Dept. of CSE

Being a coordinator of the Finite Loop Club has been a great learning experience. As a student-driven club, it has given us the freedom to explore new ideas, take initiative, and grow together as a team. I've seen students support each other, share knowledge, and work on projects that go beyond the classroom.
It's a space where everyone is encouraged to try, fail, learn, and improve. I'm proud to be part of a club that truly reflects student passion and teamwork.

TESTIMONIALS

Sathwik Prabhu
President , 2024-25

FLC transformed me from a hesitant introvert to a confident leader who could address crowds and organize major events. What started with attending workshops out of curiosity became a journey of leading the team, conducting technical sessions and building lifelong friendships.

FLC isn't just a coding club - it's an environment that pushes you beyond your comfort zone. The culture of mentorship, hands-on learning through real projects, and the network of talented peers shaped not just my technical skills but my entire personality. From struggling with public speaking to conducting workshops, every challenge became a stepping stone.

For any student seeking genuine growth, technical excellence, and a supportive community - FLC is where your journey begins.

Sinchana S.H.
Secretary , 2024-25

FLC was never just a club, it was the place where my college story came alive. From being a clueless first-year in awe of the seniors, to eventually becoming one myself, every FLC meet, Hackfest hustle, and late-night GCs will be in my scrapbook of memories. From nervously planning our first events to confidently leading as Secretary, FLC gave me a space to explore, create, and grow.

Even today, the FLC spirit lives rent-free in my mind and probably in my Google Drive too XD. Now that I'm an alum, it feels like graduating from a sitcom you never wanted to end. But the bloopers, the late-night Google Meets, and the 473 versions of “final_final_doc” will always be part of me. I’m endlessly grateful to FLC for being my launchpad and support centre all in one.

Prathama S.J.
Event Manager, 2024–25

Joining the Finite Loop Club in my first year was a turning point in my college life. From being interviewed by one of the most supportive seniors, Padmashree akka, to leading a team during HackLoop, every experience helped me grow not just technically, but also personally. The club provided a platform where I could explore leadership, team collaboration, and the excitement of building real-world projects—all while feeling supported by a community of passionate and driven individuals.

FLC gave me the opportunity to turn a dream into reality—organizing a national-level hackathon. Hackfest'24 and Hackfest'25 were milestones in my journey, filled with challenges, teamwork, and unforgettable moments. With the support of amazing mentors like Shashank sir and Puneeth sir, and the constant motivation from my peers and seniors, FLC became more than just a club—it became a family and the foundation of my college journey.

Nihal Mohan
Digital Manager, 2024–25

I was eager to join FLC from the very first year of my engineering journey, and finally became a part of it in my second year. The experience has been truly amazing—it gave me the opportunity to explore various tech stacks and develop my professional leadership skills. I am also proud of my role in being part of the 36-hour Hackathon, "Hackfest" where I was part of the core team, volunteered, and gained various ideas and insights from the participants. Thank you, FLC, for this incredible experience—I am excited to see FLC reach even greater heights.

INTRODUCTION

In the fast-moving world of technology, NMAM Institute of Technology, a unit of Nitte (Deemed to be University) is proud to be home to one of its most vibrant and driven student communities, the Finite Loop Club. Established in 2016, this premier coding club has grown into a guiding force for aspiring developers, problem-solvers, and technology enthusiasts.
Led by the dedicated faculty coordinators Dr. Shashank Shetty and Dr. Puneeth R P, along with the support of Prof. Krishna Prasad Rao the club has evolved into a space where creativity meets discipline and learning leads to innovation.

The mission of the Finite Loop Club is simple yet ambitious. It aims to provide a strong foundation in software development, inspire original thinking, and promote excellence in competitive programming. Through regular hands-on workshops, students gain the practical skills they need to meet real-world challenges in the tech industry.

In addition to learning and development, the club takes pride in executing real-time consultancy projects. These initiatives allow students to apply their knowledge in meaningful ways while also contributing to the technological progress of the institute. The club also hosts a vibrant online community through its active Discord server. Here, students connect, collaborate, and support each other through peer-to-peer learning, discussions, and project work.

Over the years, the Finite Loop Club has successfully completed several impactful software consultancy projects, leaving a mark on the technological landscape of NMAMIT.

As we continue forward, the club remains committed to nurturing the next generation of tech innovators and proving that with vision and effort, the possibilities are infinite.

CORE MEMBERS
2024-25


Satwik R Prabhu
PRESIDENT


Nandan R Pai
VICE -PRESIDENT


Sinchana S H
SECRETARY


Bhavya Nayak
Joint Secretary


Akhil Manoj
Joint Secretary


Anindya Hegde K
Treasurer

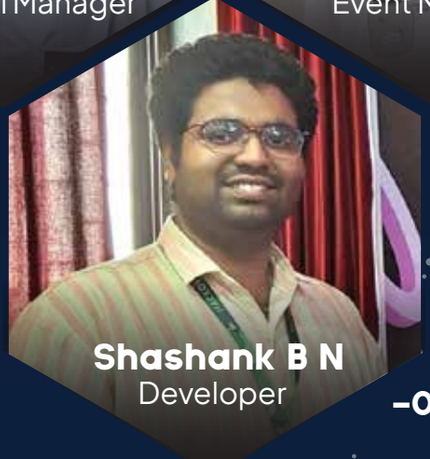

Srivatsa R. Upadhya
Technical Manager


Aniruddha Upadhya K
Technical Manager


Prathama S J
Event Manager


Nihal Mohan
Digital Manager


A Omkar G Prabhu
Lead Developer


Shashank B N
Developer


Karthik S Salian
Developer


Amrith R Naik
Developer

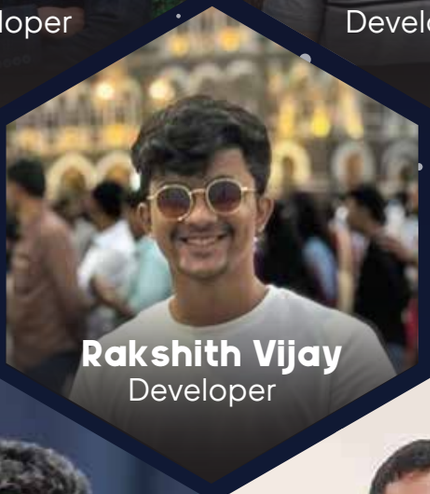
Dinesh Acharya
Developer

Len Mendonca
Developer

Vishnu Bhat
Developer

M Sayeem Ahmed
Developer

Chaithra S Nayak
Developer



Rakshith Vijay
Developer

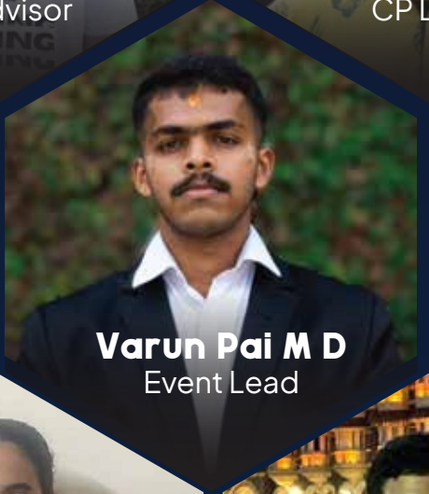
Srujan Acharya
CP Advisor

Adithya Rao K.
CP Advisor

R Ajay Prabhu
CP Lead

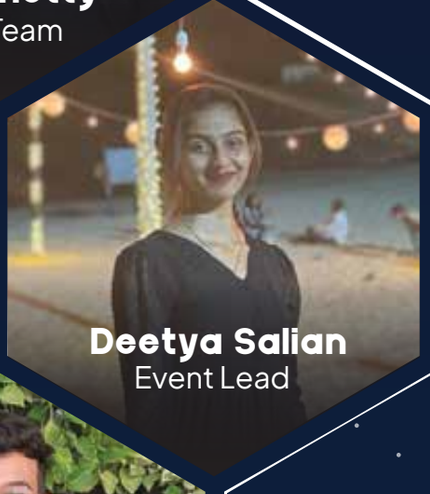
Ananth Ravi Raj Shetty
CP Team

Aryan Shenoy
Event Lead

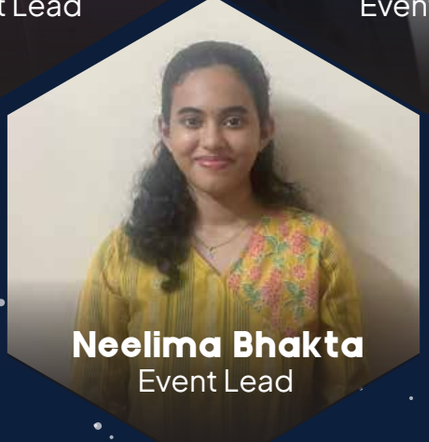


Varun Pai M D
Event Lead

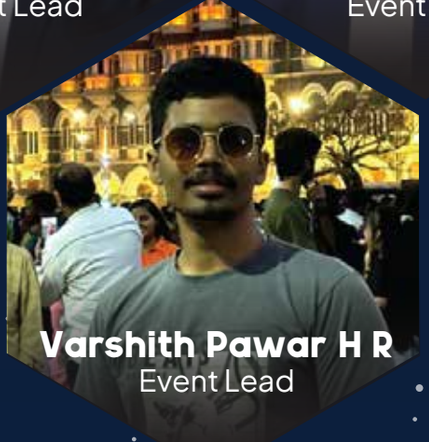
Shawn Dsouza
Event Lead



Deetya Salian
Event Lead



Neelima Bhakta
Event Lead



Varshith Pawar H R
Event Lead

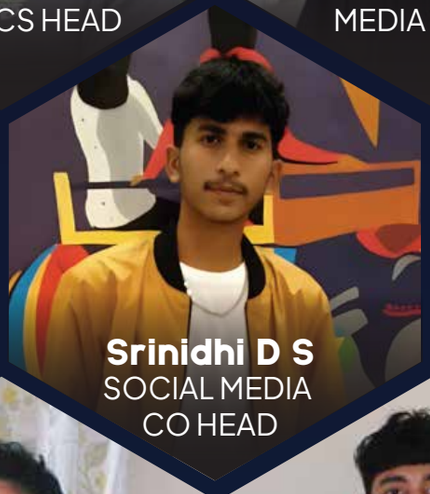
Ninaad Lobo
Event Lead

Ashish Hebbar
GRAPHICS HEAD

Arkal Varun Hegde
MEDIA HEAD

Suryanshu Choudhary
CONTENT HEAD

Anuj Pai
SOCIAL MEDIA HEAD



Srinidhi D S
SOCIAL MEDIA
CO HEAD

Adithya A
SOCIAL MEDIA
CO HEAD

Aryan Singh
GRAPHIC DESIGNER

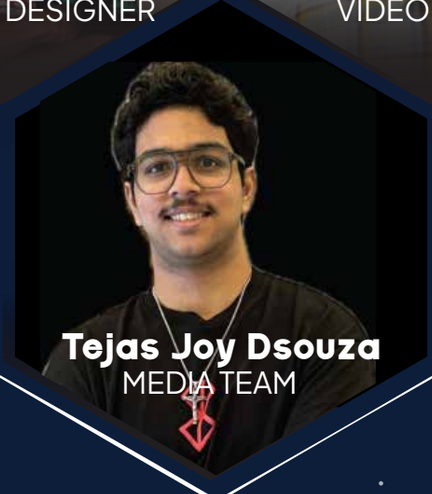
Akshay S Mayya
GRAPHIC DESIGNER

Ashish Shankar
GRAPHIC DESIGNER

Srujan Pattar
VIDEO EDITOR

Mohammed Abid
MEDIA TEAM

Aakash
MEDIA TEAM



Tejas Joy Dsouza
MEDIA TEAM

Shishir Karkera
MEDIA TEAM

CORE MEMBERS

2025-26

Nandan R Pai
PRESIDENT

Shishir G Karkera
VICE -PRESIDENT

Akhil Manoj
SECRETARY

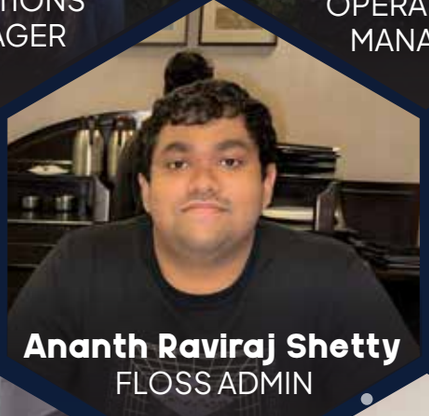
Bhavya Nayak
TREASURER

Chalana A Shetty
OPERATIONS MANAGER

Sampanna
OPERATIONS MANAGER

Arya G Bhat
OPERATIONS MANAGER

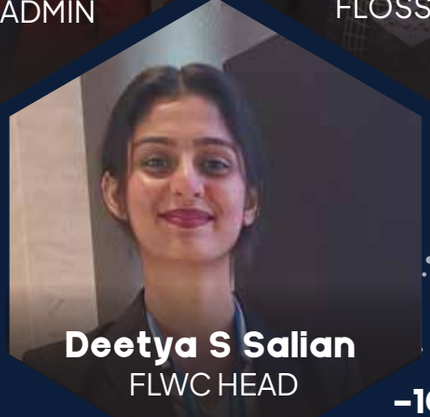
A Omkar G Prabhu
FLOSS ADMIN



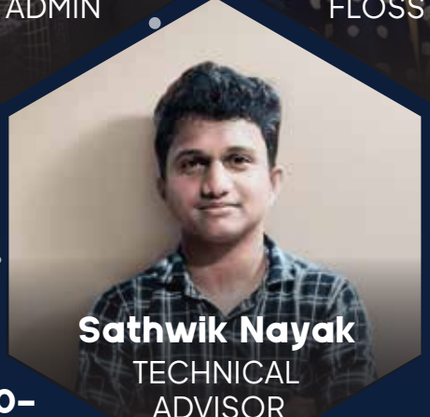
Ananth Raviraj Shetty
FLOSS ADMIN

Karthik S Salian
FLOSS ADMIN

Chaithra S Nayak
FLWC HEAD



Deetya S Salian
FLWC HEAD



Sathwik Nayak
TECHNICAL ADVISOR

Sayeem Ahmed
TECHNICAL ADVISOR

R Ajay Prabhu
DSA ADVISOR

Varshith Pawar H R
EVENT ADVISOR

Varun Pai M D
EVENT ADVISOR

Aryan Shenoy
EVENT ADVISOR

Arkal Varun Hegde
DIGITAL ADVISOR

Ashish Hebbar
DIGITAL ADVISOR

Tejas Joy Dsouza
DIGITAL ADVISOR

Asthon Mathias
TECH LEAD

Rahul N Bangera
TECH LEAD

Sushan Shetty
DEVELOPER

Mohammed Araan
DEVELOPER

Sumukha Rao
DEVELOPER

Shreesha Aithal
DEVELOPER

Paripoorna B
DEVELOPER

Keerthan K
DEVELOPER

Parimi Saketh Kumar
DSA LEAD

Dharshan S Kotian
DSA LEAD


Chinmay Kulkarni
DSA COORDINATOR


Elvin
DSA COORDINATOR


Samarth Pai
DSA COORDINATOR


Himanshu
DSA COORDINATOR


Thanmayi
EVENT LEAD


Vruddith R Shetty
EVENT LEAD


Jesmitha K
EVENT COORDINATOR


Sujnan D Devadiga
EVENT COORDINATOR


Shreya C
EVENT COORDINATOR


Aditi H Nayak
EVENT COORDINATOR

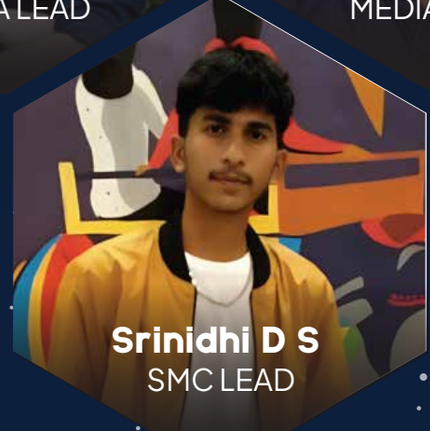

Suvidha S Karkera
EVENT COORDINATOR


V Mithun Mallya
MEDIA LEAD


Manmith
MEDIA TEAM


Thushara Balakrishna
MEDIA TEAM


Srujan M Pattar
MEDIA TEAM


Srinidhi D S
SMC LEAD


Likhit V Shetty
SOCIAL MEDIA TEAM

Mohith L A
SOCIAL MEDIA TEAM

Sharvari K R
SOCIAL MEDIA TEAM

Aryan Singh
GRAPHICS LEAD

Ghanashyam N V
GRAPHICS TEAM

Anirudh Pai
GRAPHICS TEAM

EVENTS

The Finite Loop Club has had an eventful and highly productive year in 2024–25, marking its presence as a hub of innovative student developers at NMAMIT. Over the course of the year, FLC successfully organized 10 events, each designed to foster a spirit of learning, creativity, and collaboration among its members and the wider community.

Smart India Hackathon Ideathon 2024

The Finite Loop Club, NMAMIT hosted the Smart India Hackathon (SIH) Internal Ideathon 2024 on August 31st, 2024, at the Phalguni Seminar Hall, NMAM Institute of Technology. The event served as a preparatory platform for teams aiming to participate in the government-backed Smart India Hackathon 2024.

A total of 37 teams comprising 222 students participated in the idea-pitching sessions. The inaugural ceremony was graced by Mr. Manoj S., CEO of CogniMuse, alongside faculty dignitaries Dr. D. K. Sreekantha, Dr. Puneeth R. P., and Dr. Shashank Shetty, who set the tone for the event with their inspiring remarks.

Induction Program 2024

The Finite Loop Club at NMAMIT conducted its annual induction program on August 3rd, 2024, at the Sambhram Auditorium, NMAMIT. The event welcomed new members and introduced them to the club's vision and activities for the academic year 2024–25.

The program featured Captain Ganesh Karnik, retired Indian Army officer and Chief Spokesperson of BJP Karnataka, and Mr. Karthik Rao K S, Senior Software Developer at EG India Pvt. Ltd., who shared insights on leadership, technology, and career development. The club's annual report was presented, followed by ID card distribution for the newly appointed core team.

Highlights included the launch of the club's first annual magazine, inFinite Insider, and the felicitation of student developers behind two major projects — a SaaS-based inventory system and a multilingual website for Krishnaveni Ashrayadhama. The event concluded with a tech talk on Apache Kafka and interactive sessions such as quizzes and fun tech-based activities.

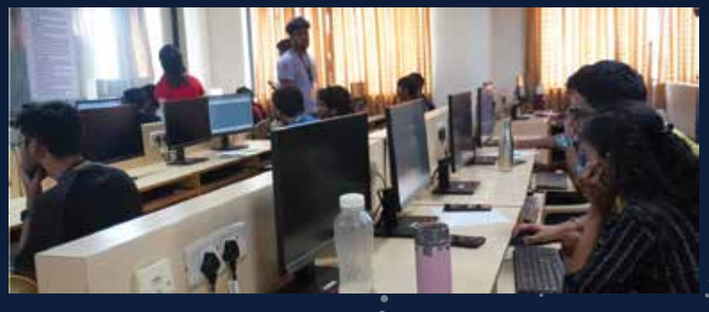
Digital Hunt 2.0

The Finite Loop Club conducted its signature puzzle-based treasure hunt, Digital Hunt 2.0: Kitty Chronicles, on October 19th, 2024, across six venues within NMAMIT. Open to all students, the event saw 63 teams (126 participants) tackle a series of digital puzzles through an entertaining and mysterious storyline involving Putin and Jeff Bezos.

Organized by R. Ajay Prabhu, Omkar G. Prabhu, and Sayeem Ahmed, the hunt was split into two rounds. Round 1 featured 18 puzzles categorized by difficulty and scored with a variable point system based on speed and hints used.

Round 2 allowed teams to choose between paths of varying difficulty, continuing the humorous quest to help Putin recover his secret Hello Kitty obsession from Bezos.

The event introduced quirky twists like genie tasks and a surprise Rick Roll, adding to the fun. Winners received cash prizes and Amazon vouchers, with Dr. Shashank Shetty presenting the awards. The event received overwhelmingly positive feedback, thanks to its creative storyline and engaging structure.



Git – Github Workshop

The Finite Loop Club organized a Git and GitHub Workshop on September 28, 2024, aimed at introducing first and second-year students to version control and collaborative development. The session was led by Nandan R.Pai and Omkar G.Prabhu, who guided participants through the core concepts of Git and GitHub. Attendees learned how to fork and clone repositories, switch branches, make commits, and push changes.

They also explored the process of submitting pull requests through a hands-on project. Over 140 students actively participated in the workshop, gaining practical experience with real-world tools. The event concluded with overwhelmingly positive feedback, reflecting its clarity, relevance, and impact.

DSA Essentials Workshop

The Finite Loop Club organized a DSA Essentials Workshop on September 14th, 2024, at ADL04 Lab, NMAMIT. Led by R. Ajay Prabhu and Ananth Ravi Raj Shetty, the session introduced participants to key DSA concepts including Time & Space Complexity, Bit Manipulation, Binary Search, Sorting, and basic Dynamic Programming.

With live coding demos and interactive quizzes, the workshop emphasized real-world applications of DSA in placements and programming contests. The event concluded with a sneak peek into the upcoming DSA Sprint, leaving participants enthusiastic for what's next.

Code Relay

Code Relay was successfully conducted on October 5th, 2024, from 9:00 AM at the SMVL51 and SMVL52 labs of NMAM Institute of Technology.
The event featured 45+ teams, each comprising three members, competing in a fast-paced, logic-driven coding showdown. The first round followed a unique "relay" format, where teammates tackled DSA problems one after another without any communication.
The top-performing teams advanced to The Final Showdown, where participants individually solved hidden-difficulty coding problems. The event concluded with three winning teams walking away with cash prizes, certificates, and goodies. Organized by the Finite Loop Club, Code Relay was a thrilling mix of strategy, silence, and speed, making it one of the standout events of the year.

HackXpo S03

HackXpo S03 was successfully held on November 30th, 2024, at the Shambhavi Seminar Hall, marking the grand finale of the month-long internal hackathon HackLoop, organized by the Finite Loop Club.
A total of 16 teams—10 from the second year and 6 from the first year—presented their innovative solutions across themes like Fintech, Healthcare, Edutech, and Sustainable Development.
The event began with a formal inaugural ceremony in the presence of Dr. I. Ramesh Mithanthaya, Vice Principal of NMAMIT, and faculty coordinators Dr. Shashank Shetty and Mr. Nagaraj, alongside judges Mr. Nagaraj Pandith (SDE - 1, Wipfli India) and Mr. Swastik Shetty (SDE - I, RedBus), both distinguished alumni of the club. Participants impressed the audience and panel with their technical creativity and practical implementations. Winning teams were awarded cash prizes, certificates, and stickers, followed by an engaging feedback and networking session with seniors. HackXpo S03 stood out as a vibrant celebration of student innovation, mentorship, and the collaborative spirit of the Finite Loop Club.

Blockchain: 0 – 1

Blockchain: 0 - 1, a hands-on workshop organized by the Finite Loop Club, was successfully held on 18th January 2025 at ADL 03, SMV Block. Spearheaded by Dinesh Acharya (3rd year, CSE), the session aimed to demystify blockchain technology and highlight its real-world applications. The workshop explored blockchain's relevance in combating corruption, building transparency, and transforming industries like healthcare, real estate, and forensics.

With references to events like the Harshad Mehta scam and the IT bubble, attendees gained a deeper understanding of how decentralized systems can prevent systemic failures. The session also introduced tools such as Solidity and IPFS, sparking curiosity among participants about Web3 development. With over 40 students attending, Blockchain: 0 – 1 proved to be an insightful and empowering experience for budding tech enthusiasts.

Crack The Placement

Crack The Placement, an online interactive session organized by the Finite Loop Club, was held on April 5th, 2025, via Google Meet. The session featured seven accomplished final-year students who shared their placement journeys into top companies like Oracle, Infineon Technologies, Samsung, JP Morgan, and more. From coding strategies and interview tips to domain-specific preparation and resume building, the panelists offered valuable insights. Over 150 students from all years participated, engaging in live Q&A and gaining clarity on their placement paths. A curated Drive resource with prep material was also shared for further guidance. Hosted by Akhil Manoj, the session was a resounding success, leaving attendees better equipped and motivated for their own placement journeys.

The Last Pair Standing

The Last Pair Standing, a non-technical team event by the Finite Loop Club, was held on January 25th, 2025, at Shambhavi Hall. The event saw enthusiastic participation from 38 teams, each consisting of two members, competing in three rounds designed to test quick thinking, teamwork, and creativity. The first round featured a tech-leaning rapid-fire quiz, followed by a dynamic Snake and Ladder challenge that incorporated tasks and surprises on a virtual game board. The final checkpoint—a thrilling campus-wide treasure hunt—had teams solving riddles to locate hidden codes. Frolic Falcons secured first place, followed by Team Black and Jaats, with cash prizes awarded to the top three. The event was lauded for its fun yet competitive format, leaving participants and organizers with an unforgettable experience.

DSA Sprint

DSA Sprint was one of the club's most exciting events, designed to bring together people who love learning and enjoy a bit of friendly competition in coding. We were thrilled to see over 300 people sign up for the challenge. The mission was simple but ambitious: solve 300 problems from a mix of topics and patterns to get everyone interview ready. To make the

journey smarter and more focused, we picked topics from the internet's most popular resources, Striver's SDE Sheet and A2Z Sheet, carefully selecting those that cover the most concepts in the least amount of time. The idea was to give participants a strong start in DSA, spark their interest, and set them up to explore advanced topics and competitive programming later. Everything ran online through our WhatsApp community, which became a lively space for sharing doubts, tips, and encouragement, and it is still active today. Participants teamed up in small groups, starting with the basics like loops, arrays and sorting, then moving on to essentials like linked lists, stacks and queues, supported by contests, curated problems and a clear roadmap to grow together.

Hackfest 2025

The highlight of the year was the second edition of NMAMIT's flagship national-level hackathon, Hackfest 2025, held from April 18th to 20th, 2025. Themed "Tech Olympus", the event celebrated lost tech and innovation. Organized by the Finite Loop Club and powered by Niveus Solutions Pvt. Ltd., with support from Palo Alto Networks, Inflow Technologies Pvt. Ltd., and Rakuten, the hackathon attracted an overwhelming 3,897 registrations from 623 teams representing 94 colleges across 8 states. After a rigorous shortlisting process, the top 60 teams competed in a 36-hour coding marathon across five tracks: FinTech, Sustainable Development, Healthcare, Logistics, and Open Innovation. Faculty advisors Dr. Shashank Shetty and Dr. Puneeth R.P., along with the student organizing team, led an event packed

with highlights including DevLoop tech talks by industry experts, engaging icebreaker activities, a vibrant jam session, and the Overtime Valorant gaming tournament. Comprehensive arrangements for onboarding, food, and accommodation ensured a comfortable experience for all participants. The event concluded with a valedictory ceremony where the winners in each track and the overall champions were recognized. With exceptional participation and industry collaboration, Hackfest 2025 reaffirmed NMAMIT's position as a hub for innovation and technological excellence.

FIGMA

A Beginner's Walkthrough

- Sanidhya H D
2nd year, AIML

Holaa !!
Welcome to the world of never-ending assignments, confusing code, and… design tools?! That's right whether you're into coding, UI/UX, app dev, or just want your project report to look like you care,
Figma is your new best friend.
Before you start sweating,.. no, you don't need to install anything. Just head to figma.com, sign in with Google, and boom, you're in. No "setup.exe", no updates, no crying... yet.
Unless when you'll suddenly hear someone say: "Just do the UI in Figma." And you're like: "Wait, do the what in the what?" That's exactly how I felt too. But don't worry I've survived that confusion and made it to the other side. If you've never designed anything in your life don't worry. This guide is made just for you, by someone who Googled "how to align things in Figma" at 3 AM.

What Even is Figma?

Figma is an online design tool. Think of it as the Canva of the tech world, but smarter and more organized. It's a tool for designing everything from app UIs to posters to wireframes… all in your browser. And yes, it auto-saves. It's used to create things like:
i. App interfaces (those nice-looking login pages you see in projects)
ii. Website layouts
iii. Posters, slides, infographics
iv. Prototypes (aka clickable fake apps that look real)
And guess what? It works right in your browser, no downloads, no installations, no "please wait while we update your soul."

Why Should you Care?

You might be thinking: "I'm a coding person. Why would I design?"
Here's why Figma will become your new best friend:
i. Mini Projects: Your team's app works… kinda. But with a beautiful Figma design? It'll look like it actually does.
ii. Hackathons: You need quick, clean UI mockups. Figma = instant visuals = bonus points from judges.
iii. Teamwork: Multiple people can work on the same design at once. It's like Google Docs, but for buttons and rectangles.
iv. Clarity in Ideas: When you can't explain your idea in words, show it visually. Figma lets you build quick mockups that speak louder than 1000 lines of explanation.
v. Resume Skill: Knowing Figma adds value to your resume especially if you're into app development, UI/UX, web design, or product thinking.

Some Tips Before You Dive In

i. Start small. Don't try to design a full app on your first day. Start with one screen or one idea.
ii. Use frames to organize your screen (like choosing a phone or laptop layout).
iii. Organize your layers, name your elements and group them neatly. It'll save you confusion later.
iv. Use templates and plugins. Figma has built-in resources and community templates. No need to design everything from scratch.
v. Watch beginner videos. A quick YouTube tutorial can save you hours of guessing.
vi. Ask for feedback. Share your work with a friend or senior they might spot things you missed.

Bonus Tips

i. Using Auto Layout keeps everything neatly spaced and easy to manage.
ii. Make Components: Reuse buttons, cards, and icons without redesigning every time.
iii. Explore Figma Community Tons of free templates, icons, and design inspo.
iv. Practice Recreating popular app screens (Instagram, Zomato, etc.) to learn layout ideas.
v. Teamwork: Share the file link. Your teammate can edit it live, and you can watch them destroy everything you made. Use version history just in case.

Remember

i. Keep things simple. Don't try to make the next Apple website in your first try. Start with 1 screen.
ii. Use YouTube if you're stuck. Literally type "how to make a login screen in Figma" there are 1000 videos.
iii. Ask seniors. We've made all the mistakes. Let us feel helpful once in a while.

Wrapping Up

Figma will confuse you. It will test your patience. You'll rage. But after a while it'll click. You will feel lost in Figma at first. That's normal. But soon, you'll be designing amazing stuff and saying words like padding and alignment like a pro.
So go ahead, open Figma, mess around, break things, learn and most importantly, have fun.
Remember, if you can drag a rectangle, you can design in Figma. (And if it crashes, just pretend it was abstract art.)
Whether you're working on a project, joining your first hackathon, or just curious to try something new , give Figma a shot.

Resources Which Will Help You Start

Figma (Official Site)
https://www.figma.com

Figma Community
https://www.figma.com/community

Figma Learn (Official Tutorials)
https://help.figma.com/hc/en-us

VIBE CODING EXPLAINED

- Shreesha Aithal
2nd year, RAI

Traditional Coding: Building a House Brick by Brick

To grasp vibe coding, let's first look at traditional coding. Imagine building a house yourself. You'd need to learn design, choose materials, and lay every brick. In coding, this means mastering a programming language (like Python or JavaScript) and writing detailed instructions, line by line, for the computer. It's a lot of work and takes time to acquire the necessary skills.

Vibe Coding: Describing Your Dream House to an AI

Now, picture a different approach to building that house. Instead of doing everything yourself, you tell a super-smart AI assistant what you want: "A two-story house with three bedrooms, a big kitchen, and a garden." The AI then handles the blueprints, hires workers, and oversees construction, all based on your description. You just share your idea; no need to know how to mix cement or hammer nails.

This is exactly what vibe coding is in programming. Instead of writing code line by line, you describe your desired program in plain language. For example, you could say, "I want a website that shows the weather in my city," and the AI will generate the code. It's called "vibe coding" because it's about capturing the "vibe" or essence of your idea without the technical details. Tools like Cursor or GitHub Copilot make this possible by converting your words into working code.

Agentic Coding: Building an Entire Neighborhood with a Team of AIs

Agentic coding takes vibe coding a step further. Imagine not just one AI assistant for your house, but a whole team working together. You tell them, "Build me an entire neighborhood with houses, shops, and parks," and they manage everything from planning to construction with minimal input from you.

In programming, agentic coding means the AI can tackle larger, more complex tasks with less supervision. For instance, you might say, "Build an online store that sells clothes and handles payments," and the AI will create the website, set up a product database, and even deploy it online. Platforms like Emergent are designed for this, allowing AI agents to draft, code, test, and launch your app based on your instructions.

How Do These Tools Work? (The Cake Analogy)

Vibe coding and agentic coding use artificial intelligence to understand your instructions and convert them into code. Think of it like making a cake:

i. Traditional coding: You're the baker, measuring ingredients, mixing batter, and baking the cake yourself.

ii. Vibe coding: You tell an AI, "I want a chocolate cake with vanilla frosting," and it bakes it for you.

iii. Agentic coding: You say, "I want a bakery that sells chocolate cakes," and the AI builds the entire bakery—oven, ingredients, and all.

What You Can Create

i. With vibe coding: A personal blog, a to-do list app, or a simple calculator, simply by describing what you want.
ii. With agentic coding: More advanced projects like an online store, a social media platform, or a game with multiple levels.

Popular Tools

i. Cursor: Helps you write code by understanding your instructions.
ii. GitHub Copilot: Suggests code as you type, and with vibe coding, it can do more of the work.
iii. Emergent: A platform where you describe your idea, and AI agents build, test, and deploy the app for you.

Why These Tools Are Useful

Vibe coding and agentic coding are exciting because they make programming more accessible and faster:
i. Anyone can create software: You don't need to be an expert programmer to build something useful. Non-programmers can describe their ideas and see them realized.
ii. Saves time: Get a working app in minutes, instead of hours or days.
iii. Focus on creativity: Spend more time on your ideas and less on technical details.

Research shows that 78% of development teams are already using AI-assisted coding tools, a number that's growing (GitLab, 2025). These tools lower entry barriers, making it easier for beginners like you to start creating.

The Catch: Things to Watch Out For

While powerful, vibe coding and agentic coding aren't perfect:
i. Check the AI's work: Just like with AI-written essays, you need to review for mistakes. The AI might not always perfectly understand your intent, so test the code to ensure it works correctly (e.g., a weather app might use outdated data or miss a feature).
ii. Learn programming basics: These tools simplify coding but don't replace the need for understanding. Think of vibe coding as training wheels they help you start, but you still need to learn to balance (traditional coding). Knowing the basics helps you fix errors, customize projects, and understand the AI's actions.
iii. Tools are still improving: Vibe coding and agentic coding are new and evolving. The AI might make mistakes or struggle with complex projects (e.g., a simple blog might work perfectly, but a game with advanced graphics could be trickier). Over-reliance on AI without understanding the code can lead to lower-quality software or security issues. Staying involved and continuously learning is crucial.

Should You Use Vibe Coding or Agentic Coding?

As juniors, you're in a great position to explore these tools while building a strong programming foundation. Here's some advice:
i. Start with the basics: Learn a simple programming language like Python. Practice writing small programs to understand how coding works. This will equip you to guide the AI and fix issues.

ii. Experiment with vibe coding: Try tools like Cursor or Emergent for small projects. Describe a simple app idea, like a to-do list, and see what the AI builds to understand how these tools operate.
iii. Don’t rely only on AI: Use vibe coding and agentic coding as helpers, not substitutes for learning. Traditional coding offers more control and flexibility for creating exactly what you want.
iv. Have fun and be creative: These tools are excellent for bringing ideas to life quickly. Experiment with building websites, apps, or games without needing years of coding experience.

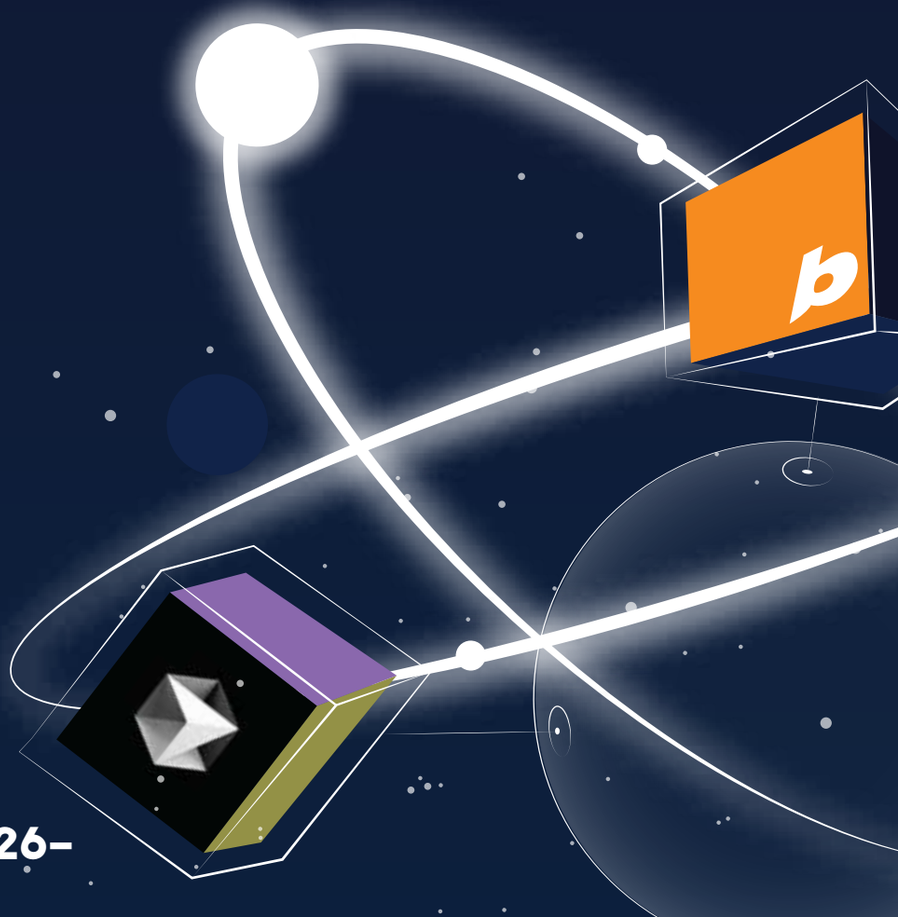
A Final Analogy (Back to the Cake!)

i. Traditional coding: You learn to bake, buy ingredients, and follow a recipe to make the cake yourself.
ii. Vibe coding: You tell an AI, “I want a chocolate cake with vanilla frosting,” and it bakes it for you. You just need to taste it to ensure it's good.
iii. Agentic coding: You tell the AI, “I want a bakery that sells chocolate cakes,” and it builds the entire bakery for you.
By learning to bake (code) yourself, you can tweak recipes, fix mistakes, and even create new flavors. Vibe coding and agentic coding are like having a super-smart assistant, but you're still the chef in charge.

Looking Ahead

Vibe coding or agentic coding are transforming software development, making it easier for everyone to create apps and websites. Experts predict that by 2027, more companies will use agentic AI to automate complex tasks (GitLab, 2025). For juniors like you, this is an exciting time to jump in, learn the basics, and experiment with these tools.

So, don't hesitate to try vibe coding, but keep learning traditional programming too. With a blend of both, you'll be ready to create amazing things and perhaps even help shape the future of these AI tools.



A student's journey to the world of DATA SCIENCE

- Nishmitha Shetty
2nd year, AIDS

What Even is Data Science?

Let's say your best friend binge-watches horror movies, drinks black coffee every Sunday, and somehow ends up doom-scrolling Instagram at 2 AM. Now imagine if Netflix knew all of this. Spoiler: it does. And that's thanks to Data Science.

Data Science is the art of turning messy, chaotic data into meaningful stories and smart decisions. It's like being a digital Sherlock Holmes except instead of a magnifying glass, you have Python, Pandas, and a strong craving for clean datasets.

So... What Makes a Data Scientist?

First of all, you don't need to be a math wizard or a hoodie-wearing genius to dive into data. What you do need is curiosity and a bit of patience. Here's your starter kit:

i. Python – Your main weapon. Easy to learn, super powerful
ii. Pandas – For manipulating data like a spreadsheet ninja
iii. Matplotlib/Seaborn – For turning numbers into cool charts
iv. Scikit-learn – For building basic machine learning models
v. SQL – For talking to databases (basically the Google of data storage)

It's not about memorizing formulas - it's about thinking, "What can this data tell me?"

Where Is Data Science Used?

Here's the cool part - data science is everywhere:

i. Netflix recommends shows you might actually watch
ii. Hospitals predict patient risk and improve treatments
iii. Cricket teams analyse performance stats to plan better
iv. Banks detect fraud before you even see the suspicious transaction
v. Social media apps figure out what reels you can't stop watching

In short: If you've ever thought, "How did they know I needed this?" - the answer is probably some data scientist sipping coffee in the background.

Why Data Science Might Be Your Thing

It's not just for coders. If you're into psychology, business, biology, or even design - there's a version of data science waiting for you. You could end up as:

1. Data Analyst

What They Do:

i. Collect, clean, and analyse data
ii. Create dashboards and reports using tools like Excel, Power BI, or Tableau
iii. Help teams understand trends, customer behaviour, or performance
iv. Answer questions like: "Why did sales drop this quarter?" or "Which product has the highest returns?"

Roadmap:

i. Learn Excel (pivot tables, formulas)
ii. Learn SQL (queries, joins, filtering)
iii. Basic Python (Pandas, NumPy)
iv. Learn data visualization tools (Tableau, Power BI, or Matplotlib)
v. Learn EDA (Exploratory Data Analysis)
vi. Build dashboards using real-world data (e.g. sales, weather, sports)
vii. Learn business metrics (KPIs, ROIs)

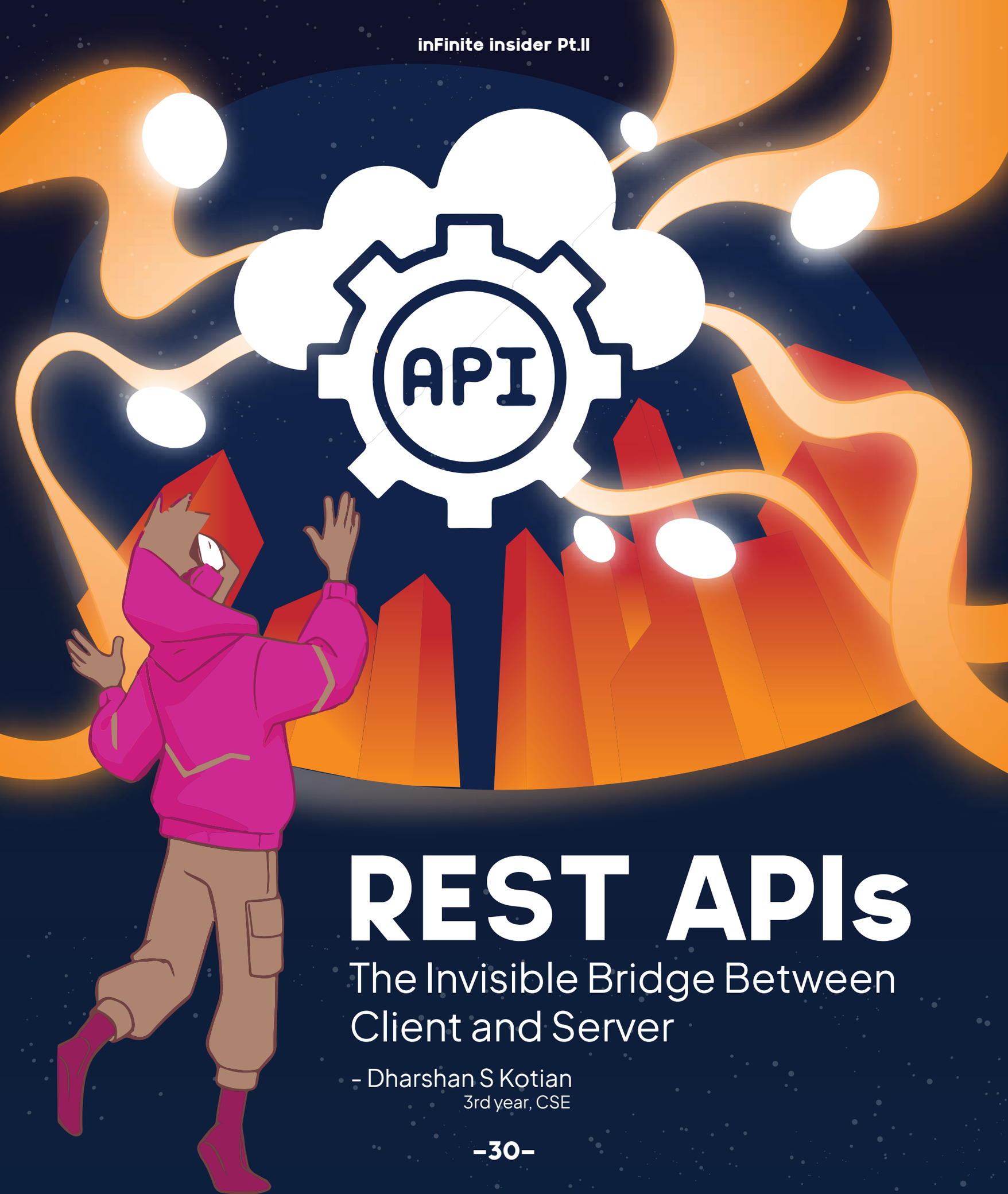
2. Machine Learning Engineer
What They Do:
i. Design and train machine learning models
ii.Work with large datasets (structured & unstructured)
iii.Optimize algorithms and evaluate performance
iv.Develop systems like recommendation engines, fraud detection models, or speech recognition
Roadmap:
i. Master Python
ii.Learn NumPy, Pandas, Matplotlib, Seaborn
iii.Study core ML concepts (supervised vs unsupervised learning)
iv.Start with scikit-learn
v.Learn algorithms: Linear Regression, SVM, Decision Trees, KNN, Clustering
vi.Learn model evaluation metrics (precision, recall, F1-score)
vii.Work on end-to-end ML projects using datasets from Kaggle

3. AI Researcher
What They Do:
i.Publish research papers on new models/techniques
ii.Work on cutting-edge AI (transformers, generative models, reinforcement learning)
iii.Collaborate with academic or R&D teams
iv.Often work with large-scale datasets and compute resources
Roadmap:
i.Strong foundation in Python, Math (Linear Algebra, Probability, Calculus)
ii.Study ML fundamentals thoroughly
iii.Start reading AI research papers (arXiv, NeurIPS, CVPR, etc.)
iv.Learn deep learning using PyTorch or TensorFlow
v.Implement classic papers (e.g., ResNet, GPT, BERT)

4. Domain-Specific Analyst
What They Do:
i.Use data skills + domain knowledge to solve real problems
ii.Help doctors predict disease, marketers target ads, or coaches analyse performance
iii.Combine domain context with data-driven decision-making
Roadmap:
i.Pick a domain: healthcare, sports, agriculture, fintech, marketing, etc.
ii.Learn core data science skills: Python, Pandas, visualization, statistics
iii.Learn domain-specific tools/datasets (e.g., medical records, stock data, sports APIs)
iv.Work on targeted projects (e.g., predict stock movement, analyse athlete performance)

Final Thoughts – Your Journey, Your Pace

Some people start in the first year. Others find it in fourth. And that's okay. Don't rush because someone else is ahead. And please don't trust only one YouTuber like they're your guru explore, ask, read docs. And if you start and realize it's not your thing? That's fine too.
Try cybersecurity, web dev, cloud whatever excites you. But if something about patterns, predictions, and data puzzles makes you smile… welcome to the world of Data Science.



REST APIs

The Invisible Bridge Between Client and Server

- Dharshan S Kotian
3rd year, CSE

I remember building my first proper web project: a simple To-Do List using HTML, CSS, and JavaScript. It wasn't easy. I jumped between YouTube tutorials, paused a hundred times, rewound even more, but somehow I made it work. I was proud of it. I could add tasks and delete them. It felt like I had just built something real. But then... I refreshed the page. And everything was gone. All the to-do items I had created just vanished.

At first, I thought I'd made a mistake. Maybe I forgot to save something? But no that's just how it was.Every time I reloaded the page, all my hard work disappeared into thin air. I paused for a second and thought,

“Wait… a To-Do List is supposed to last. It's supposed to be saved. What am I missing?"

So I went back to Google and typed in the most beginner question ever:

"How to make data not disappear on refresh in JavaScript"

I don't remember everything I read that day, but one word stuck in my mind : "Databases".

Apparently, to remember things like tasks, you need to store them somewhere and that's where databases come in. So I opened YouTube again, searched for "To-Do list with Database", and found a whole new world waiting for me. That's when I heard this sentence in a tutorial:

"We'll need to create an API to fetch and send the data from the backend."

And just like that, I was introduced to REST APIs, the invisible bridge between the frontend code I write and the data stored on a server.

What is a REST API?

So there I was, a beginner web developer who had just discovered that building a frontend wasn't enough. I needed a way for my app to remember, to save, to fetch things, not just display them.

That's when it hit me: if my to-do list was going to be more than just a pretty interface, it needed to talk to something, basically a REST API.

So … what even is a REST API?

An API (Application Programming Interface) is how two pieces of software communicate like a messenger that lets them send and receive data. REST (Representational State Transfer) is a set of rules that defines how that communication happens using standard web methods like GET and POST.

Put together, a REST API allows your frontend to make requests to your backend and get structured responses typically using HTTP and JSON.

Here's how I like to think of it: **you're the frontend (the client), the API is the waiter, the backend is the kitchen, and the menu represents the available endpoints**. You don't walk into the kitchen and cook your own meal instead, you tell the waiter what you want from the menu. The waiter takes your order to the kitchen, the kitchen prepares it, and the waiter brings it back to you. You never see the kitchen or worry about how the meal is made. You just expect the right dish to arrive. That's exactly how a REST API works: **it takes your request from the frontend, passes it to the backend, and returns the result clean, organized, and ready to serve.**

Hello Server! My First API call

I initially started my backend development journey with Django. Already fascinated by Python, Django offered a clear solution. With it, I could build full-stack applications that served dynamic content directly, no separate API needed. It tightly connected the backend and frontend, allowing me to render HTML pages with logic baked in. I built several projects using Django and learned a lot. But eventually, I grew curious about frontend development. I wanted

to explore React, but to build a full-stack app with it, I needed a separate backend. Although I considered switching to Node.js and Express (which I eventually did) at the time, I didn't want to leave the comfort of Django. As always, I turned to YouTube and searched for a full-stack project using Django and React. That's when I found a blog tutorial built with React and Django REST Framework (DRF). It became my first real experience creating a proper REST API backend and I still remember my first GET request returning the response:

"Hello, Server!"

If you want to start building your own APIs, there isn't a single ''best" framework or tool that fits everyone. The right choice often depends on what language you're comfortable with, the scale of your project, and how much structure or flexibility you need. For example, if you're into Python, Django REST Framework (DRF) is great for building structured APIs quickly, while FastAPI offers modem async support and automatic documentation. If you are more familiar with JavaScript, Express.js is a minimal and flexible option that's perfect for learning and small projects. For larger applications, NestJS adds a layer of structure with a modular, TypeScript based design. Java developers often prefer Spring Boot, and PHP developers might use Laravel. All of these tools have strong communities and rich ecosystems. Rather than searching for the perfect tool, it's better to pick one that aligns with your current skills and learn the core concepts of API design they translate across frameworks.

Why APIs matter?

You might be wondering, are APIs just about connecting the frontend and backend? That's what I thought in the beginning too. But once I started working on real-world projects, I discovered their true power.

For one, APIs allow you to reuse the same backend across multiple frontends, whether it's a web app, mobile app, or even a third-party integration. They also help you design cleaner, more maintainable code by separating concerns between logic and UI. And REST APIs are naturally scalable and easier to deploy across different environments. Tools like Postman made it super easy for me to test endpoints, send requests, and debug responses without needing a full frontend. Along the way, I also learned the importance of following best practices like using proper status codes, versioning APIs, adding authentication, and structuring error responses clearly. These are just a few of the advantages there's much more to explore as you dive deeper into building modem, production ready APIs.

When it comes to designing efficient and scalable APIs, I personally follow the Microsoft Azure Web API Architecture guidelines along with other modern best practices like stateless design, versioning, proper status codes, and layered architecture. These patterns have helped me build cleaner, more reliable APIs that scale well in real-world use .

Final Word

APIs might seem intimidating at first, but they're just tools that, when used right, can power anything from simple apps to large-scale systems. The key is to stay curious, build consistently, and embrace the learning curve. Every API you design is a step toward becoming a better developer. Keep building, keep refining your journey has just begun.

HYPE, HOPE OR HALLUCINATION

Navigating the new AI reality

- Vishwas Sharma
2nd year, CSE

Remember a time, not so long ago, when "AI" was a buzzword mostly confined to sci-fi movies and niche tech circles? It felt like a distant, futuristic concept. Then, almost overnight, it felt like someone flipped a switch. Suddenly, AI wasn't just a concept; it was a tool on your browser, a feature in your apps, and the single biggest topic of conversation in every boardroom, classroom, and cafeteria. This isn't just another tech trend; it's a full-blown revolution unfolding in real-time, and we've all got front-row seats. The question is no longer if AI will change things, but how we adapt to the massive wave that's already here.

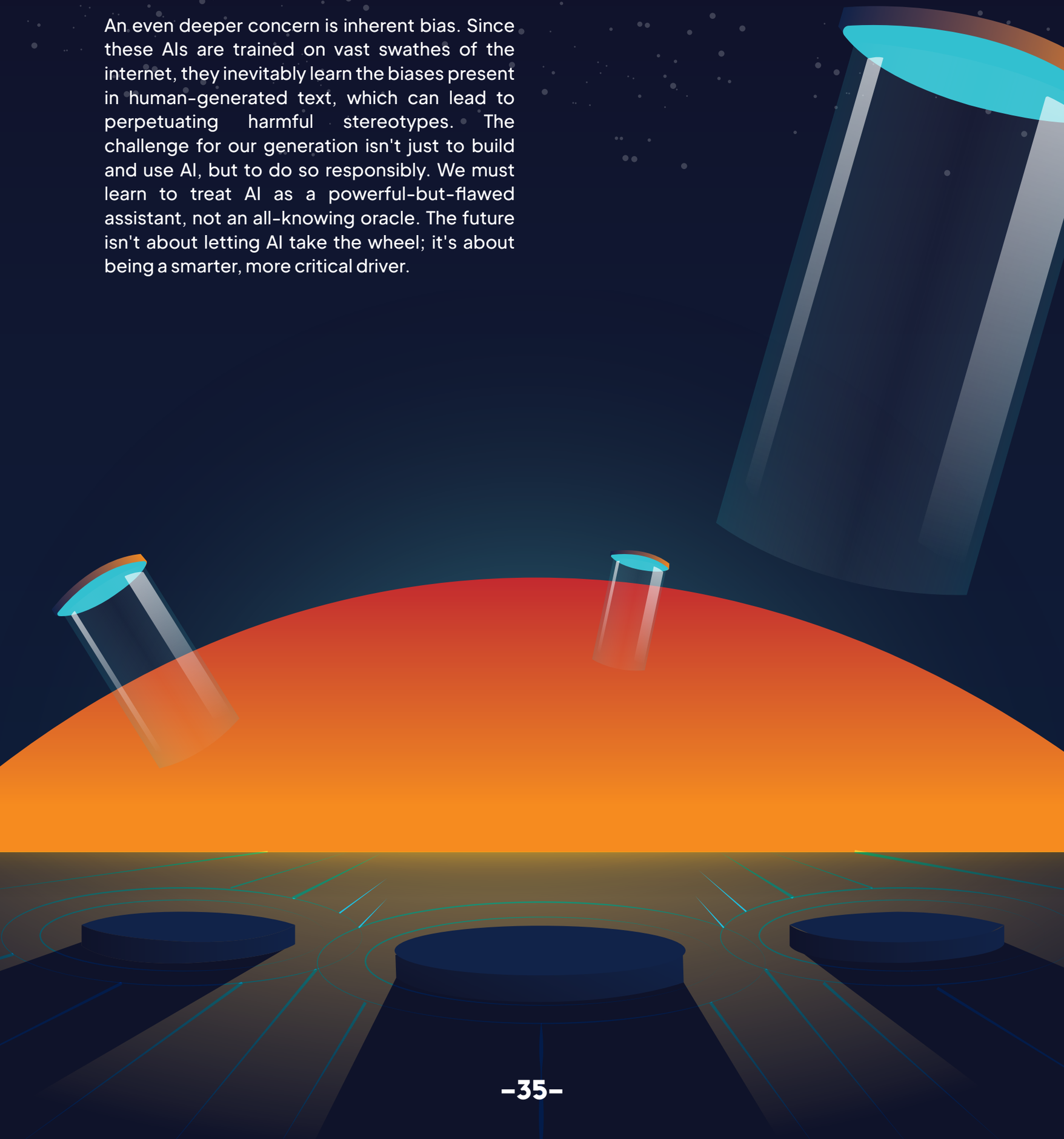
The ripple that became a tsunami can be traced back to a specific date: November 30, 2022. This was the day OpenAI unleashed ChatGPT upon an unsuspecting public. Powered by its Generative Pre-trained Transformer 3 (GPT-3) model, which was trained on a staggering 175 billion parameters (think of them as knobs the AI tunes to learn patterns), it could write essays, debug code, and converse with stunning fluency. The tool hit 1 million users in just five days and 100 million in two months, triggering a "code red" across Silicon Valley. The race was on. Microsoft, already a major investor in OpenAI, integrated its tech into the Bing search engine (now Copilot). Google, caught on the back foot, scrambled to release its own model, Bard (now powered by its Gemini family of models). Meanwhile, Meta took a different route, releasing its LLaMA model to the research community, which was subsequently leaked and kickstarted a vibrant open-source AI movement. This revolution wasn't just limited to text; AI models for image generation like DALL-E, Midjourney, and Stable Diffusion also exploded in popularity, proving that AI could be creative in ways we never imagined.

This technological arms race is already reshaping our world, and its biggest immediate impact is on productivity. There's a popular saying circulating now: "AI won't replace you,but a person using AI will." This isn't just a catchy phrase; it's the new reality. For students and professionals, AI assistants are becoming indispensable co-pilots. Need to write a research paper? A conversational search engine like Perplexity AI can sift through hundreds of academic sources in seconds, providing summarized answers with citations. Struggling to draft a professional email or a creative blog post? An LLM can turn your bullet points into a polished draft instantly. For developers, the change is even more profound.

Tools like GitHub Copilot are now integrated into coding environments, writing boilerplate code, suggesting entire functions, and catching bugs. Studies have shown that developers using Copilot can complete tasks up to 55% faster, freeing them up to focus on complex problem-solving rather than tedious syntax. AI is essentially a force multiplier for human intellect, automating the mundane so we can focus on the meaningful.

However, with great power comes the need for great caution. The other side of this shiny AI coin is the danger of over-reliance and misuse. If we let AI do all our thinking, we risk the atrophy of our own critical skills-much like relying on a calculator can make us forget basic math. Furthermore, these models are not infallible. They are notorious for "hallucinating" -confidently stating incorrect information as fact. Trusting an AI's output without verification can lead to spreading misinformation.

An even deeper concern is inherent bias. Since these AIs are trained on vast swathes of the internet, they inevitably learn the biases present in human-generated text, which can lead to perpetuating harmful stereotypes. The challenge for our generation isn't just to build and use AI, but to do so responsibly. We must learn to treat AI as a powerful-but-flawed assistant, not an all-knowing oracle. The future isn't about letting AI take the wheel; it's about being a smarter, more critical driver.



CODING with 6th SENSE

-By Dharshan S Kotian
3rd year, CSE

Why the Best Coders Don't Always Think–They Feel

What if I told you there's a superpower every great developer quietly cultivates one that can't be taught in lectures or captured in textbooks? That power is intuition. And it's real.This article is a personal journey into how that 'gut feeling' has repeatedly come to the rescue in high-stakes coding moments, complex projects, and everyday debugging.

The Gut Feeling that Guides Us

Picture this: you're coding late into the night, frowning at your screen,and every approach you try feels like hitting a wall. We've all been there, haven't we? Then suddenly, like a light bulb flicking on in a dark room, a solution sparks in your mind. You can't explain why, but it just clicks and, often, it works!

That inexplicable "gut feeling"? That's intuition; your inner compass guiding you through the tech maze, even when logic seems to take a coffee break. We have grown up with the mantra of "study hard" and "understand everything before applying." But in the fast-paced world of tech, especially during those adrenaline-filled coding contests, I've learned that intuition is our silent superpower. It bridges the gap between what you've learned and what you've truly absorbed through experience. We might not talk about it much in tech circles, but trust me intuition has saved me more times than I can count.

The Pattern Recognition Game

In the beginning, Data Structures and Algorithms felt like a foreign language. I'd wrestle with problems for hours, lost in confusion. But as I tackled more challenges, patterns began to emerge.

Suddenly, it was instinctual!

i. "I bet this is a binary search problem ... "

ii. ''This input is practically screaming for a prefix sum ... "

iii. "Ah, a two-pointer technique will do the trick!"

These weren't logical deductions I sat down and thought through. They were instant reactions. A kind of muscle memory of the brain. That's when I realized I wasn't just solving problems. I was training intuition. It's like chess. Beginners see moves. Masters see patterns. Over time, I stopped "solving" every problem and started "recognizing" them. And that shift is what unlocks real fluency in DSA.

When I jumped into college projects or my websites, I often gave myself a headache trying to figure out my next step. Should I choose this tool? How do I shape the layout? Why isn't my logic panning out? But with time and experience, I developed a sense of understanding:

i. ''This condition isn't structured quite right."

ii. "Wait, users might get confused here; let's simplify the flow."

iii. ''This layout just doesn't feel like the right time to rethink it!"

I couldn't always pinpoint why; I'd simply seen enough examples, read the documentation, and faced enough failures to instinctively know what might be off. These weren't things I was taught. They were things I absorbed. From building, from breaking, from reading, from watching. Your mind stores those tiny moments and eventually turns them into intuition.

When Time's Up

In a high pressure coding interview, you don't get time to research, map out 3 approaches, and slowly eliminate the wrong ones.

You get a problem.

You get a clock.

And you get one shot. The silence of the room feels louder than ever.

Every second ticks like thunder. Your brain wants to panic and wants to overthink. But the people who perform best aren't the ones with the best notes, they're the ones with the best feel for the problem.

They've seen enough.

Tried enough.

Failed enough.

They've internalized patterns most can't even describe. They don't stop to analyze every possibility. They recognize the core. They don't start from zero – they start from instinct. From a place deep in their subconscious where experience speaks faster than logic. And that, more than anything, is what makes the difference.
You see, most candidates walk in with formulas memorized, tricks polished, and patterns revised the night before. And that's important but it's not enough. Because when the pressure rises, memory slips.

Logic stutters.

Notes blur.

What remains is your mental muscle memory, your ability to recognize a problem, sense a structure, and feel your way toward a solution without needing to consciously think through every line. Every bug you fix. Every problem you fail. Every discussion you join. Every contest you mess up. It all counts.
Because in that interview, when time's up, nerves are high, and clarity vanishes, intuition is the only tool that still works.

I BUILT MY OWN CLOUD

A GenZ Story About setting up your own Home Servers

- Varshith Pawar H R
4th year, CSE

Ever caught yourself at 2AM thinking, “Why am I letting Google hold my cat pics (UwU) hostage when I could just run my own server?” Yeah, same or doomscrolling reels and thought “yeah my life sucks let’s make it worse”. That’s how half of us end up spiraling down the r/homelab hole, staring at setups with more blinking LEDs than our local ISP. It starts with a random Reddit post, then a YouTube binge, then suddenly you’re wiping an old laptop, googling “How to SSH? What is SSH?” and bragging in Discord about your private cloud (telling you are setting it up).

So, What’s a Server?

Here's the thing a server isn’t some giant rack in a cold warehouse. It’s literally any computer that serves stuff files, movies, your dumb side projects to you or your friends. That dusty old laptop you nearly threw away? Boom. Instant server. All it needs is power, a network cable, and the will to run 24/7 while you sleep peacefully (Do you?) knowing your hidden memes stash (Ahem Ahem) are safe under your roof.

Before you go digging up that crusty old laptop, read this first.

You gotta understand a few networking basics so your shiny new server doesn’t disappear every time your router sneezes. Every device in your house phone, TV, smart light that never listens gets an IP address from your router’s DHCP. By default, that IP can change whenever the router feels cute, which is a nightmare when you’re trying to SSH in from bed and your server ghosted you with a new address (if you are used to ghosting then it's fine).

So, Step one: Give your server a static IP either from its OS settings or by reserving it in your router’s DHCP table (better reserve one in your router’s DHCP table). Now it always lives at the same address on your local network. Easy, isn’t it? Golden tip by your homie Search networking for dummies, learn IPV4, IPV6, OSI layer, TCP, UDP, Subnetting.

Next up: your new bestie or your future arch-nemesis the OG port.

Think of them like tiny doors (not a door to your heart, it’s wide open). SSH? Port 22. Plex? 32400. Minecraft server? Probably 25565. If you want to reach your server from outside your WiFi bubble, you’ll have to open the right ports and forward them in your router settings. But don’t get reckless, every open port is an open invite for weird bots to poke around. Open what you need, lock down the rest or uninvited guests will be in your home.

Okay, you know what’s in, what’s out now unleash chaos and hit that setup (not hard you need your laptop) like a sleep-deprived goblin. Time to pick your OS and if you’re not feeling spicy or if you are not a masochist, just go with Ubuntu Server (just go for it if you’re a first timer you can use a fancy distro later). It’s popular for being beginner-friendly , Google has an answer for every dumb question you’ll ask at 3AM, and it won’t trash your old hardware (use 22.x.x LTS for 8+ year old machines). If you’re mostly storing files and want a chill UI, Unraid or TrueNAS are like training wheels for selfhosting. Click more, type less, break fewer things (maybe or may not be).

Plug your server into your router with an Ethernet cable and start with step 1. You may ask why not connect to wifi trust me; WiFi works but you’ll regret it when your Plex buffers mid-movie. So Burn your OS image to a USB (USE VENTOI), boot it up, and run through the install wizard and do as Step 1 tells. When that’s done, you never need a monitor again, your new bestie is SSH. Type “ssh yourname@server.ip” from your main PC and bam, you’re hacking into your own house. Hacker vibes achieved.

Before you get carried away spinning up 10 random apps Calm down take a deep breath and do the things below:

Lock it down. Update your OS, install UFW (firewall) (UFW is the G.O.A.T), set up fail2ban so

bots get booted when they spam your SSH port. While you're at it, learn how to use SSH keys. They're basically super passwords that bots can't guess, so you sleep better at night (google SSH for dummies).
Just like that your server lives. Flex mode: ON. So, what's next?
Run The classics: Plex or Jellyfin for streaming your shows anywhere so you can flex your local Netflix when the real one hikes prices again. Maybe spin up "Nextcloud" your own Google Drive, minus the ads and creepy scans. Then comes Docker, the holy grail (the other G.O.A.T) runs tons of apps in tiny containers that won't trash your whole server if you break one. Want an easy way to see and manage them? Install Portainer or Yacht and thank yourself later (I never did because I love trouble).
"Wait, so am I stuck being a Discord mod, glued to my home WiFi 24/7?" Nah, calm down there's a way to peek at your server while you're out grabbing coffee. For this question everyone screams "just use a VPN!" Sure, I tried (I promise I did try). I read all the WireGuard docs (Almost all), generated keys like a wizard and then something always broke (I don't know maybe I am noob, give it a try yourself, maybe you could set it up) but VPN is more secure.
So instead, I did what every half-burnt-out home labber does. I used a quick tunnel workaround. SSH tunnels or a Cloudflare Tunnel are lifesavers when your VPN setup won't cooperate. Basically, you open a secure pipe from your laptop to your home network, sneak in, get what you need, close it. Not the fanciest solution, but hey it works and keeps your brain cells alive. Downside you need to
have your basic auth with 30 chars username and 50 char length password or build a stronger auth
(Tip: run the tunnel in the background so the domain name never changes).
Meet DDNS you're another homie to keep your server reachable from anywhere. Your ISP probably loves to mess with your public IP changing it whenever just like your crushes. With DDNS services like DuckDNS or No-IP, you link your ever-changing IP to a custom name like myhomeserver.duckdns.org. So instead of memorizing boring numbers, you just type a cute name and you're in.
So, to everyone still here after all this chaos here's the real tea.
Honestly stuff will break. One day your Plex library disappears, the next day your static IP pulls a Houdini. Or you lock yourself out with your shiny new firewall. But here's the deal every time you fix it, you get smarter. You start reading logs instead of panicking. You learn why your ports matter, what NAT is, how to do things properly. And slowly, you start explaining to your friends why their WiFi sucks and they actually listen because now you're the "network wizard (or a Madman)" in the group.
So yeah, building your own home server is messy, chaotic, and extremely fun. It's the kind of project that teaches you more real-life tech than a hundred Reels ever will. And the flex? Immaculate. You'll have your own piece of the internet, your own data, your own streaming empire. One day you're Googling "SSH command for dummies", next day you're showing off your fancy Grafana dashboards with rainbow graphs (yes rainbow graphs with cute gifs (Hehe I am building my own dashboard with agentic AI)) of your CPU temps and everyone on Discord wants your setup to play Minecraft (Yes, Minecraft is fun, use TLauncher).
So, here's your sign to dig up that old laptop. Burn that ISO. Lock down that static IP. Open a port, close it again, break stuff, fix stuff, and brag shamelessly when it works. Congrats you're no longer just another user. You're the sysadmin or sys-slave of your own tiny kingdom. And trust me,
you're going to love it. Happy building!!

EVERYONE hates their work.

– Aryan Singh
3rd year, AIML

Everyone has a weird emotional bond with their own output, **they never consider it done**. If you try doing any kind of creative or technical work yourself, especially if you are a novice, you will feel this massive skill gap that exists between what you imagine and what you build... **which is actually a really good sign of your potential.** This feeling is the exact reason that pushes your craft forward, this is not something you need to get discouraged about. It is the same feeling that also drives many people in other fields, developers often feel dissatisfied with their code, seeing how much cleaner it could be if they just had a bit more time, skill or experience.

I personally feel this while doing all sorts of work be it graphic design, video editing, sketching or heck writing this article. I do not write (or type) much, so I know for a fact that if I get better, I am going to dislike this piece a lot. Some of you are probably going “The flow here is total chaos”, “Does this guy even know English?” or if you enjoy Hindi cinema “Arre Kehna Kya Chahte Ho?” and I do not blame you, as the title suggests, even I hate this article, but I just hope that someone (out of the seven people that will actually read this) will understand that it is normal and sometimes good to not like what you produce.

But this does not mean that you hide your work from the world just because you dislike it, **criticism and support are also equally important in your journey to become a better engineer or artist as anything else.** Support helps you build confidence in yourself and criticism... well, breaks that confidence and keeps you grounded. **You need to learn to defend your code against haters but also collect all feedback that actually matters.**

As the French poet Paul Valéry said, **"A work of art is never finished, merely abandoned."** This suggests that there is always room for improvement, even at the highest level. Every poem you write, every microservice you deploy, every game level you paint will have small imperfections that you just will not be able to resolve... at your current skill level. **You will get better, just stay patient and consistent.**

In this world of tutorials and everybody telling you to make projects for your resume you might forget the fact that **it's not about the things that you create rather, it's more about you and your journey to become a better developer, artist, designer or anything you want to be.**

“You Are the Greatest Project You’ll Ever Work On”

~ Josh Ether

NOTHING to SOMETHING

- A Omkar G Prabhu
4th year, ISE

Author's Note

This article might not be fully true to the facts mentioned, but serves the purpose of trying to get the reader to understand why things are the way they are today. The more you read this article, the more technical it gets. Feel free to google stuff at your will to fact check and learn more.

Prologue

It's the year 2000, You wake up and it's your 5th birthday and your dad gets you a new computer, Yahoo, but what do you do with it? You ask dad, if he could teach you something to do with it? "Ok, use this paint brush and make a beautiful house" says your dad. Days go by, your artist skills aren't improving and you find out something interesting, something intriguing, THE INTERNET. You try to use it without letting your dad know, You find out there is something called a website, and server and hosting... It's like you just teleported into a new world, full of unknown words confusing you more than the "Call me Dada" your dad used to say 3 years ago. Somehow by the age of 10 you have understood all these terms, what they mean, what they do, everything. You find out your hobby, your purpose of life, PROGRAMMING.
This prologue is used to set the reader as a character in the story. In short you as a budding platform engineer, Your life story awaits below.

Chapter 1 – What are you watching

Since your parents are Asians, you are born smart, you quickly learn HTML and PHP, even before your mom makes her grocery list you have your TODOs ready in your website.

One fine day, as you are staring into the computer, your parents catch you, since you are a teenager they believe..., "Ah what's that?", "a TODO website", "you made this?", "sorry dad", "no its great, why don't we host it".
Hosting means to run a piece of code on the internet, which has a purpose, in this case a TODO website, that can be accessed on the internet either protected behind a firewall or many a times not. It just requires a CPU, power supply and an internet connection that exposes the piece of code running on your CPU to the internet. Since you have a PHP server, which serves your simple but complex (since you are a kid) backend and simple but amazing (since you are a kid) frontend, you want to host it with your dad's motivation. You set up all necessary tools with your dad's help on your PC, set up the router to expose your IP address, buy a domain and link to your IP. All set. Tomorrow is the day when you get to brag about your creation proudly with your classmates. As you open the website in the school PC, it shows blocked.
This was your first time learning, you don't own THE INTERNET. Your next goal, learn what are networks, how can your school block your website but allow theirs. It takes you an entire year, but you have mastered networks, physical and virtual networks (which your school uses). This time you take your friends out to a Cafe to show your website, and it works!!

Chapter 2 – 503 Service Unavailable

It's been a year now, and you no longer look at your TODO as an amazing website, it looks pale compared to the web game you just made. Full of animations, cool characters, a very enthralling storyline. Since you have the habit of bragging by now, you show it to your friends, they love it, next morning you wake up to play the game, and

it shows “503 Service Unavailable”. This time, you don't know what has happened, you go back to your basics, “Dad, can you help me?”. He helps you troubleshoot and you guys understand that, your game needs a lot of RAM, at least 8GB and your PC supports only 4GB, you ask your dad for an upgrade, he says “No, I have a better plan”. He introduces you to Cloud. Not the airplane cloud, “THE INTERNET CLOUD”. You guys buy something called “Virtual Private Server (VPS)”. Dad says it's cheap, lets buy for 16GB RAM, You are more than happy, “Dad when will it come?”, “Come, it won't come, it's just there on the internet and you use it, as a service”. You learn SSH (cool). And set up your web game server, and cross your fingers. It works!!
Virtual Private Server, think of it like a computer with a lot of RAM and storage, let's say 1000000000000000GB RAM and.. Whatever you get it. It provides a part of it to you on pay to use as if you owned it yourself, accessing remotely. No tension of upgrades, electricity cut, bad internet. Any issues, sue the cloud provider and chill (Just kidding, it ain’t that simple, but yes better than your home PC)

Chapter 3 – Infatuation

You are now in your high school, there is this new girl in your class, ah, lets just say she is cute. You want to impress her, but she says she doesn't like nerds, and you know by now that you are one.
On a side note, it's totally fine if you are a nerd, someone will definitely find you cute ;)
Since your web game is very popular, you make a master plan, “Let me add a multiplayer mode, and I'll join as a teammate of this girl and save her from enemies, Yess”, you no need to let her know you are a nerd, but you become her HERO.
And hence you start adding the upgrade, it takes you a month, but you are finally done with it. All like it a lot. It's cool, playing with friends, never heard of it. It's your big day, your day to impress, All your classmates join the game to start playing, your master plan in progress, Around the same time, some russian kids find out your game, and they hop on a match as well, suddenly, your and all your classmates games are stuck, they are lagging. This time however you anticipated for stuff like this to happen and had precautionarily set up an auto scaling load balancer with your dad’s help. It takes about 5 seconds but suddenly all the lag gets resolved, and seems like the girl is enjoying the game. Well done Soldier!!
When there is a high demand of resources, one can set up autoscaling with load balancing. Instead of having a 128GB RAM VPS, where only at times the entire 128GB is utilized, we can have 18 8GB RAM VPS, where only 1 runs usually, and a monitor is setup that observes CPU and RAM usage and based on these parameters, either spins up a new VPS or stops an old one. Since many servers are serving similar requests, we set up a special server which takes all the ingress (incoming requests) and diverts it to these scaled-servers based on which is least utilized currently and other parameters.

Chapter 4 – Hero returns

Years pass by, you are a grown adult now, you have come a long way, have a good stash of money and want to get out of the JOB and get into BUSINESS. You have a million dollar idea, start a photo sharing website, where people could share images, tag each other and connect socially online, basically a phonebook for photos, perhaps a facebook??

You set up a company, ahh a startup for now, employ smart people (easy to find, asian? You are in, not asian, we are sorry). You have a big stash of technical knowledge, of how load balancers work, how auto scaling works, etc etc. It takes some months before you guys go live, but it takes just hours before you guys are famous. Within a month you have become the fastest growing IT startup. But your competitors are not behind, they have come with better solutions catering those who are not satisfied with you. You guide your developers to bring in a lot of updates based on customer reviews, and reviews are getting better.
Although, as time goes on, there are a lot of bugs found, a lot of things going wrong, some customers are frustrated, it's either now or never. Yes, you guessed it right, “Dad, I need help”. During this growth of yours, dad has become a tech architect, he knows how to scale systems for high availability and stability. He suggests splitting the system into micro systems managing their own objectives. It takes effort, but with your developers' grit and your motivation, somehow, you guys have converted your system from a monolithic architecture to a micro-service architecture. You now have few servers dedicated to sending mails, some to handle authentication, some to handle 3rd party APIs and so on.
Somehow the bugs are getting solved, you suspect it's because now, fixing one bug has a very low chance to induce another, since the logic is split and different teams work on different aspects of your system. If it works, don't question it!

System Design is a concept in Computer Science which theorises the pros and cons of various system architectures. The most commonly used is a Monolithic architecture, One server receiving the request, doing whatever is necessary and responding with the response. Straight forward. Although it proposes its own challenges, since one bug introduced by someone might affect the working of a piece of code written by a totally different developer. Hence, architecture like micro-services exist, where we split the logic into meaningful smaller logic and execute the requests in a pipeline. Usually in the real world we see a mixture of many system designs.

Chapter 5 – Trapped in a container

You are married, your business is on the boom and have recently gifted a brand new PC to your daughter on her 5th birthday. You have news, competitors are developing a new technology called K8s, but you have no idea what K8s are. Your daughter is drawing something in MS Paint, yes you taught her how to use her PC, Kudos! It looks like some hexagons, some are touching each other, some are inside each other, “This is the master and these are slaves” shouts your daughter.
Weird for a 5 year old to talk about slavery, ahh but for a second you think, what if instead of a load balanced, auto scaling VPS system, there was a system where, some servers, lets call them control plane, controls other servers, lets call these worker nodes, and these workers run micro-services based on their capability, lets call them pods. With this we can easily onboard a variety of VPS. Also why not group servers from Europe together for faster communications and so on, let's call these clusters. This is a risky move, but developers like it, so your company adopts this architecture. After a few months, your competitors have revealed what K8s are.

And that's exactly what you had built a few months before. Thanks to your daughter. Kubernetes, often called K8s, ‘K’ followed by 8 alphabets and a ‘s’. It’s not a platform, not an operating system, it's an architecture design (not in literal terms). Some packages are set up in a server which helps it behave as a master and some other packages are set up in other servers which helps them behave as a slave. Technologies like Docker, ContainerD help run services within a single worker node separately, utilizing the resources to the max.

Chapter 6 – Mood swings

2 years pass by, your daughter is learning tech faster than you, must be the genes. She asks a question, “If I have an app to manage the share market, the traffic keeps varying quickly, that too in bursts, and I doubt if auto-scaling could catch up with demand variations, what if the server crashes?”. First of all you are happy that she is a techie, second of all, her question really intrigues you, what if traffic is having mood swings? (metaphor intended). Some days later your daughter comes running to you,
“Here, I found a solution. It’s a server itself, but it is really fast, it caches and starts really quickly, I call it serverless”. Since you have a team of great developers, you take their help to assess her solution. Lets just say they believe it's done by some 50 year old experienced programmer. They are amazed!!
Serverless is an idea where one sets up servers such that they boot up in milliseconds, serve the request and die, unlike traditional servers which are alive forever, these servers cache previous calculations and data, and only serve one request before shutting down. If traffic is varying but expectable, traditional servers are preferred since they are easy to run and cost effective, but if traffic is unpredictable, serverless systems come to rescue, since they can very quickly boot and help maintain the traffic.

Conclusion

All the technology around us might intrigue us a lot, some are amazed while some are scared of them. But almost always there is a reason and a story as to why they come to be. There are many more nuances to things at scale, like Infrastructure as Code (IAC), Continuous Integration and Continuous Deployment (CICD), Observability, DevSecOps etc. but those stories are for a different day.

OPEN SOURCE?
I HARDLY KNOW HER

– Ananth Raviraj Shetty
4th year, CSE

What open source?

What do all those 15 Linux distros you've rage-installed at 3AM (but still can't get the microphone working on), that JavaScript framework with 7 stars on GitHub and 800 dependencies, and the AI tool you used to generate your anime waifu all have in common? Their source code is free, free as in speech, and sometimes as in "free to ruin your weekend".

Yes, all of these are examples of open source software: code that's out in the wild, available for anyone to read, modify, break, fix, fork, or make worse. Open source is the wild west of software, equal parts innovation, collaboration, and occasional chaos.

By definition, open source means the source code is public, and you're allowed to redistribute it, improve it (or at least in your opinion), and build on it. It's like group projects, except the group is the whole internet, and some people are actually helpful. But why are we so obsessed with whether some code is floating around on the internet for free? Can't I just live a peaceful, blissful life using software without ever poking around in how it works? Isn't that the sane, normal, mentally-stable thing to do?
Well... yes. And also, absolutely not. Because behind every "magically working" app is a black box full of bugs, undocumented behavior, and sometimes a while(true) loop or a Thread.Sleep(1000) keeping the world from collapsing. And in this article, I'm going to try and convince you why open source actually matters, not just as a nice idea, but as something you, future engineer, should care deeply about.

Why open source?

Alright, so the code is free. Big deal. Why should you, a college student juggling deadlines, imposter syndrome and placement fear care?
Because open source is the reason half your tech stack exists and the other half is secretly duct-taped together with it. Take any web framework you've used to spin up yet another portfolio site, or that AI library powering your machine learning project, the one you proudly claim "built AI" for your resume, it all runs on code that's been shaped by countless contributors, bug reports, late-night commits, and a surprising amount of coffee. It's not just code, it's collaboration at scale. Every tool you use is standing on the shoulders of hundreds (if not thousands) of developers who decided to give their code to the world, just so you could pip install or npm install to download sophisticated packages like is-even-odd or left-padding.

Another reason? You get to work on real, production-grade problems without ever stepping into an office or writing a "Dear Hiring Manager" email. No dress codes, no awkward HR interviews, just you, your laptop, and a GitHub issue tagged as a good first issue. Contributing to open source means you're not just solving toy problems from textbooks, you're helping build and maintain the same tools people actually use in the real world.
Contributing to open source isn't just about code, it's a flex. It shows that you can work with real-world systems, read other people's code without crying (too much), and make meaningful contributions. More importantly, it puts you in the same GitHub threads, and Discord servers as seasoned developers from across the globe.

People you can learn from, collaborate with, or maybe even accidentally impress. Over time, this becomes more than just networking, it's a support system for your growth. Whether it's mentorship, job referrals, or just people who understand your obsession with semicolons, open source gives you that community. But hey don't expect overnight fame or job offers after your first pull request. Sometimes your PR gets ignored, sometimes it gets rejected. That's okay. The real win is that you're learning, growing, and slowly becoming the kind of developer people want on their team.

Finally, here's a theory I've been holding onto: open source mirrors the way human creativity works. Think about every book, movie, game, or piece of art you love? It wasn't created in a vacuum. Every great creator was inspired by others. Artists borrow styles, writers remix tropes, directors reimagine stories. If your favorite creator had never been exposed to those influences, what you love today might not exist or it might've looked completely different. Humans don't thrive in isolation. We thrive when ideas flow freely, evolve, and build on each other. We've always done our best work in open, collaborative environments not closed rooms with padlocked doors. That's what open source is. It's not just about code, it's about letting ideas mutate, improve, and live on. Without it, we risk slowing down the very engine of innovation.

Where open source?

The next obvious question: where do you actually find these open source projects? Well... everywhere. Seriously. A good rule of thumb? Start with what you already use.

Whether it's your favorite web framework, that CLI tool that you extensively used, or a library you used in a project, you already understand its features, its quirks, maybe even its bugs. You know its soul. That makes contributing easier and way less intimidating. Even big companies follow this logic, they actively contribute to open source projects that are part of their tech stack. Why? Because it's smart to improve the tools you depend on. You can do the same. So pick something familiar, check out their GitHub repo, look for issues tagged good first issue, and dive in. No need to start with Linux kernel-level madness. Baby steps.

How open source?

Now that you're armed with all this knowledge, you're probably ready to make your first open source contribution. But hold up, let's make sure it's a drama-free experience (trust me, open source has its share of drama). Here's how to get started without accidentally angering a project maintainer:

i. Read the Community Guidelines
Every decent project has a CONTRIBUTING.md or CODE_OF_CONDUCT.md. Read them. These aren't just legal fluff, they tell you how the project wants things done.
ii. Understand the Code Style
Tabs or spaces? Semicolons or no semicolons? CamelCase or snake_case? Matching the existing style is a sign of respect and helps your PR get merged faster.
iii. Check Out the Issues Tab
Start with good-first-issue, help-wanted, or anything with clear instructions. These are made for beginners and often come with guidance.

iv. Ask, Don’t Assume
If something’s unclear, open a discussion or comment on the issue. Most communities are happy to help as long as you’re respectful and not just shouting “IT’S NOT WORKING.”
v. Test Your Code
If the project has tests, make sure you don’t break them. If it doesn’t… well, don’t be the one to break production. (That’s for your future full-time job.)

Conclusion

Open source isn’t just about code, it’s about collaboration, learning, and building something bigger than yourself. Whether you're fixing typos, squashing bugs, or writing entire modules, every contribution counts. It’s a chance to grow as a developer, connect with a global community, and maybe even leave your mark on the tools the world runs on. So go ahead, clone that repo, read that README, and make that first commit. The internet is waiting.

PROJECTS

The club is all about diving into real-time projects that give our budding developers hands-on experience. We're excited about the diverse range of projects we've tackled across various domains, and we can't wait to showcase a few of the highlights.

Incridea Website

The Finite Loop Club once again had the privilege of developing the official website for Incridea, NMAMIT's flagship techno-cultural fest. Aligned with this year's theme, "Echoes of Eternity", the website delivered a rich and immersive user experience with smooth transitions, engaging visuals, and intuitive navigation. More than just a fest portal, it became a central hub - from registrations and schedules to result updates and accommodation info. This year's highlight was an interactive digital hunt where participants explored the site in search of hidden Timestones, accompanied by Ryoko, the central character of the theme storyline. The hunt sparked excitement, as users raced to solve clues, unlock surprises, and climb the leaderboard. With its seamless performance and creative integration, the website once again became a signature element of the fest, drawing praise for its design, interactivity, and execution.
Incridea : https://incridea.in

Digital Hunt

Digital Hunt 2.0: Kitty Chronicles took the thrill of cryptic puzzles to a whole new level with a quirky storyline and a custom-built platform developed by the FLC web team. The site seamlessly handled multi-round progression, time-based scoring, and hint penalties, all wrapped in a smooth, intuitive interface. With added elements like a dynamic leaderboard and branching puzzle paths, the platform brought depth and excitement to the gameplay—making the hunt a standout digital experience.
Digital Hunt :https://digitalhunt-24.vercel.app/

Hackfest `25

The official HackFest 2025 website offered a seamless experience with real-time updates, intuitive registration, and clear event guidelines. A custom-built digital evaluation portal allowed judges to assess submissions efficiently, eliminating the need for paperwork and ensuring quick, accurate scoring. Integrated GitHub repositories streamlined project submissions, while continuous on-ground tech support kept everything running smoothly throughout the 36-hour event. The robust digital framework played a key role in making HackFest a streamlined, high-impact experience for all involved.
Hackfest Website : https://www.hackfest.dev

Digital Heist

The thrilling Digital Heist event brought together participants from all skill levels for an immersive, beginner-friendly digital treasure hunt. Featuring a dynamic storyline woven with challenges in web development, puzzles, logic, and more, the event combined both digital and real-world elements for a truly unique experience. A real-time interactive map guided teams through engaging tasks, while live tracking and fun physical activities added to the excitement.The competition unfolded in two intense rounds. Round 1 tested problem-solving skills with timed puzzles, strategic use of hints, and precision scoring, selecting the top teams for the next stage. Round 2 elevated the challenge with a graph-based puzzle map, varying difficulty levels, and the adrenaline of Speedrun Encounters—mini-games that offered point bonuses or penalties. The format encouraged teamwork, strategy, and adaptability, with fair play ensured through clear rules and time-bound execution.

Yakshagavishti

Blending culture with code, a dedicated platform was developed for Yakshagavishti, a one-of-a-kind Yakshagana competition. The website showcased event details, registration, and updates while preserving the essence of this traditional art form. Designed with simplicity and elegance, the platform offered a smooth user experience and highlighted the event's unique identity, reflecting the commitment to support diverse initiatives through digital innovation.
Yakshagavishti : https://www.yakshagavishti.in

ACHIEVEMENTS

Members are the heartbeat of our club. Without its dedicated developers, FLC would not be what it is today. Their continuous triumphs bring pride to our community.

Sambhram 2024

Sambhram 2024, an intercollegiate fest held on the 6th and 7th of December at Shree Devi Institute of Technology, Mangalore, featured a series of technical competitions that tested the problem-solving and programming skills of participants.

A Omkar G Prabhu and Amrith R Naik claimed 1st place in Code Clash (Code Relay), showcasing excellent coordination and coding speed. In Error Extermination, Satwik R Prabhu secured 2nd place for his strong debugging skills. In Project Nexus, Satwik R Prabhu and A Omkar G Prabhu secured 2nd place for their innovative project work. The event provided a valuable platform for students to demonstrate their technical talent and creativity. FLC members Shubham Vaikunt Raikar and Sujan S Ganiga also participated in the Webverse event, gaining valuable experience and exposure.

Synergia 2024

Synergia 2024, hosted by Sahyadri College of Engineering, featured DevHost, a flagship tech fest held on the 8th and 9th of November, 2024. Among its major attractions was HackNight, a 20-hour hackathon.

Aryan Singh, Keerthan K, Rishi Bhati, and Neelima Bhaktha secured first place and won a cash prize of ₹30,000. Athul D Bhandary, Samarth H Shetty, Nagvarapu Saarvari, and Rakshith N Poojary secured the second runners-up position and bagged a prize of ₹10,000. Additionally, in the Bit Breaker (CTF Event) conducted at Synergia 2024, A Omkar G Prabhu and Amrith R Naik secured second place, highlighting their strong skills in cybersecurity and problem-solving. FLC members Darshini Shetty, Abhisha D Hegde, Aneesh Sanil, and Mithali also participated in the event, gaining valuable exposure and experience.

CodeFury 7.0

CodeFury 7.0, a 24-hour hackathon, was held on the 10th and 11th of August, 2024.
Nandan R Pai, A Omkar G Prabhu, Amrith R Naik, and Pratham A Kadekar emerged as winners, showcasing their technical expertise and collaborative spirit. FLC members Athul D Bhandary, Samarth H Shetty, Nagvarapu Saarvari, and Rakshith N Poojary also participated in the event, further reflecting the club's enthusiasm for competitive programming and problem-solving.

DevRev Women AI Hackathon

DevRev Women AI Hackathon, held from the 18th to the 27th of November, 2024, celebrated innovation and inclusion in the field of artificial intelligence. Bhavya Nayak, Chaithra Nayak, and Rashmi N emerged as winners, showcasing exceptional problem-solving skills and domain knowledge. Amulya G Rudresh secured the runners-up position, while Deetya S Salian and Shravya A Prabhu claimed the second runners-up spot. Their success stood as a testament to the growing presence and impact of women in technology.

SAP iTech Hackfest

SAP iTech Hackfest, a national-level innovation challenge, was held on the 25th and 26th of July, 2024. Two teams from FLC reached the final round, marking a significant achievement at the national level. The first team included Akhil Manoj, M Sayeem Ahmed, Rahul AR, Pratham A Kadekar, and Dsouza Shawn Joseph Collins, while the second team comprised Aishik Roy, Ameya Kowshik, Rakshith N Poojary, Samarth H Shetty, and Ashish Hebbar. Their performance reflected strong problem-solving abilities, innovation, and teamwork. The event provided valuable exposure to real-world challenges and fostered collaborative development.

InFeynite Hackathon

InFeynite Hackathon, held on the 30th and 31st of August, 2024, brought together young innovators to work on dynamic technical challenges. FLC members Anand Bobba, Vedant Mahalle, and Hariharanath participated in the event, demonstrating enthusiasm for hands-on development and collaborative problem-solving.

Hack 4 Mini 2.0

Hack 4 Mini 2.0, held from the 27th to the 30th of May 2025, brought together creative minds to build impactful solutions through collaborative development. Ameesha J Ruzario, Varshith Pawar H R, Karthik S Salian, and Athul Bhandary secured the runners-up position, demonstrating strong technical skills, teamwork, and innovation under pressure.

Gromo Finarva AI Hackathon

Gromo Finarva AI Hackathon, held from the 30th of May to the 1st of June 2025, challenged participants to develop AI-driven financial solutions. Sushan Shetty, Abhin Balakrishna, Samar Rihan, Shaun Marvell, and Sudarshan J reached the finals, placing among the top 10 teams. Their achievement reflected a strong understanding of AI applications and a commendable ability to innovate in high-impact domains.

DSU DevHack

DSU DevHack, a national-level hackathon held on the 27th and 28th of September 2024 at Dayananda Sagar University, Harohalli, Bengaluru, brought together bright minds to innovate in domains such as AI, ML, IoT, Blockchain, Cybersecurity, and Cloud Computing.

The event saw active participation from FLC members Varshith Pawar H R, Karthik S Salian, Ameesha J Ruzario, Dinesh Acharya, Ananth Ravi Raj Shetty, M Sayeem Ahmed, Snehal Shetty, Pratham Chaatra, Karan Karkera, Ayush Chaudhary, Prathwik, and Amrith R Naik, who demonstrated technical proficiency and collaborative spirit.

The Great Bangalore Hackathon 2025

The Great Bangalore Hackathon 2025, held on the 16th of March, provided a competitive platform for budding developers to showcase innovative thinking and technical skill. FLC members Sushan Shetty, Sharva Dhanvi V, Rakshith LK, and Sudarshan J participated in the event and were shortlisted to Phase 2, reflecting their strong problem-solving capabilities and potential in high-stakes environments.

Hacksagon

Hacksagon, held from 27th to 29th June 2025, was a competitive platform that challenged students to build innovative tech-driven solutions within a limited timeframe. FLC members H. Varsha Ravishankar, Disha Y H, Vishwas Sharma, and Nishmith participated in the event under the team name Nexus Navigators, showcasing their enthusiasm for hands-on problem-solving and collaborative development.

THE FINAL BYTE

As we close the second edition of inFinite Insider, we reflect on a journey built on curiosity, creativity, and an unwavering passion to explore the ever-expanding world of technology. From decoding concepts to capturing the pulse of our events, this magazine mirrors the spirit and drive of the Finite Loop Club.

What began as a new venture with our first edition has now become a meaningful tradition that not only records what we do but captures who we are as a community. Through each byte of content, we celebrate the learners, builders, and thinkers that define us.

Each story shared and every idea expressed helps code the evolving narrative of a club that thrives on collaboration and purpose. The energy, dedication, and imagination of our members continue to fuel everything we achieve together.

To everyone who contributed their time, effort, and creativity, thank you. You are the logic in our structure, the heartbeat of our progress, and the reason this magazine continues to grow. Here's to more editions, more breakthroughs, and the infinite loop of learning ahead.