Annual IT Technical Magazine 2025-26

Vol. 04

TECH CHRONICLES



FROM IDEAS TO IMPACT

TURNING INNOVATIVE CONCEPTS INTO REAL WORLD VENTURES

BEYOND CLASSROOMS

REAL PROJECTS. REAL CHALLENGES. REAL LESSONS.

CRACKING THE TECH INTERVIEW

STRATEGIES, EXPERIENCES & LESSONS FROM THE PLACEMENT JOURNEY

BUILD. BREAK. INNOVATE.

INSIDE THE PRESSURE, TEAMWORK & EXCITEMENT OF HACKATHONS

IT DEPARTMENT

OUR VISION

The Department of Information Technology aims to develop Information Technology engineers who work professionally and creatively for the advancement of technology and betterment of society.

OUR MISSION

- To impart quality education by developing information technology facilities, faculty and resources that generate professionals who are leaders for a dynamic information society.
- To develop a collaborative culture, so as to nurture an environment of increased research amongst the students and faculty.
- To encourage hands-on learning by fostering industrial partnerships to create real-world solutions through innovation, product development, entrepreneurship and consultancy services.
- To enhance human potential by encouraging transparency and accountability amongst all stakeholders, in order to nurture ethical values in students

TABLE OF CONTENTS

YOUR GUIDE TO THE LATEST IN TECH, CAMPUS INNOVATIONS, AND THE STUDENTS DRIVING THE DIGITAL FRONTIER FORWARD.

Placement Tips & Interview Experiences

New & Emerging Technologies

Start-Up Ideas & Entrepreneurial concepts

Project Experiences

Internship Stories

Hackathon Experiences

EDITOR'S NOTE



It has been a pleasure serving as the Editor and being part of the Editorial Team this year. Watching individual submissions come together into a complete edition was a rewarding experience, and I am grateful to everyone who contributed their time and work to this magazine. This magazine is a genuine snapshot of what our branch looked like this year, the work, the people, the moments that do not usually make it into official records. Thank you for being part of it, even as a reader.

Naeva Setia
B.E.(I.T.), 7th Semester

CO-EDITOR'S NOTE



Laying out this issue meant I got to see it first, before the covers, before the final polish, when it was still just scattered files and half-finished drafts on my screen. There's something quietly special about that. Watching it all click into place, page by page, article by article, was honestly one of my favorite parts of this year.
To everyone who wrote, edited, contributed, or just picked this up to read, thank you. This issue exists because of you.

Vrinda Dubey
B.E.(I.T.), 7th Semester

LEAD'S NOTE



It has been a wonderful experience being a part of the Editorial Team and serving as the Third-Year Lead. From putting ideas and experiences together to seeing them come alive, every little effort has been a memory in itself. I hope this edition reflects the energy, achievements of people from our branch, and moments that made this year truly special for all of us. I hope you all enjoy reading!

Dhruva Vermani
B.E.(I.T.), 5th Semester

START-UP EXPERIENCES & STORIES

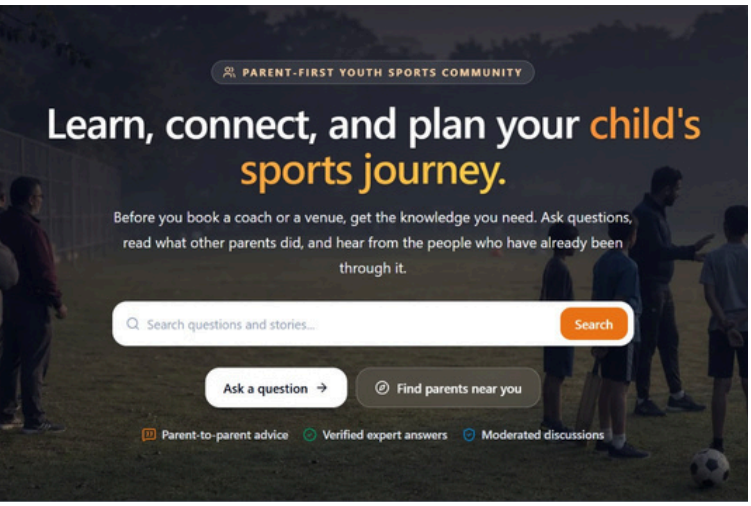


WHY I THREW AWAY 2 MONTHS OF CODE (AND WHY IT WAS THE RIGHT CHOICE)



Written by
Jatin
B.E.(I.T.), 5th Semester





It started with a LinkedIn message that turned into a phone call, and that phone call turned into the technical spine of a startup I now lead the engineering for. A sports parent reached out with a problem he'd lived firsthand: finding the right coaches, academies, and guidance for his kids was harder than it should have been, scattered across word of mouth and outdated directories with no real way to compare or trust what you were looking at.

That conversation became PowerMySport, and my job was to turn a frustration into something people could actually use. I built the technical foundation from scratch, the architecture, the codebase, the user flows, the deployment pipeline, all of it, because that's the part of building a company that never shows up in a pitch deck but decides whether anything else is even possible. We launched as a booking platform first, betting that the fastest path to value was connecting parents directly to coaches, venues, and academies through the site.

For two months we built on that assumption, refining the booking flow, wiring up the infrastructure, treating it as the obvious starting point. But building infrastructure before you have users is a trap that looks like progress while you're in it. We were pouring effort into supply, listings, venue partnerships, booking logic, without the demand to make any of it matter, and it became clear the platform was getting harder to build, not easier, because we were solving the wrong problem first. So we pivoted. Instead of leading with bookings, we flipped the model to lead with guidance, giving parents and young athletes real direction on sports, pathways, and potential before ever asking them to book anything.

That shift taught me something no course or internship could have: the best technical decision is worthless if it's solving a problem users don't have yet, and being willing to tear down two months of work you built with your own hands is not failure, it's the actual skill that separates people who ship products from people who just ship code. If you're a student sitting on an idea, waiting for the perfect plan before you start, don't. Build the ugly first version, watch where it breaks against reality, and have the discipline to pivot when the data tells you to. That's the whole game.

BUILDING A STARTUP WHILE CHASING A DEGREE: LESSONS FROM MY JOURNEY

Written by Mehtaab Singh
B.E.(I.T.), 5th Semester



When people hear the word startup, they often imagine a well-funded company, a large team, and a carefully planned roadmap. My experience was very different. For me, a startup began with a simple idea: to create something that people would genuinely enjoy. That idea eventually became Jackarrow Games, an independent game development studio that I started while pursuing my engineering degree.

Starting was exciting. We had ideas, motivation, and a vision for the kind of games we wanted to build. But very quickly, I realized that having an idea is the easiest part. Turning it into reality is where the real journey begins. Unlike established companies, our team was extremely small. Every task—whether it was development, testing, planning, building the website, maintaining our online presence, or solving unexpected technical problems—had to be handled by us.

Every day presented a new challenge, and almost every challenge required learning something completely new. The biggest obstacle, however, wasn't technology. It was time. As an engineering student, academics always demanded attention. Semester examinations, assignments, internships, and project submissions often took priority. There were periods when game development had to pause completely because other responsibilities came first. Initially, this felt discouraging. It seemed as if every interruption pushed the dream a little further away.

Over time, I realized that progress is not measured by how fast you move—it is measured by whether you keep moving. Instead of giving up whenever development paused, I learned to restart. Every time there was an opportunity, I returned to the project with new ideas, fixed old problems, and continued building. Even small improvements accumulated into meaningful progress. This journey taught me lessons that no textbook could have. I discovered that startups are not built solely with technical knowledge. They require patience, discipline, adaptability, and the willingness to solve problems that you never expected to face.

Some days were spent writing code. Other days were spent debugging for hours, planning future features, improving the website, or sharing updates with the community. Every responsibility became part of the learning experience. One of the most valuable lessons was understanding that perfection is not a prerequisite for starting. Many students postpone their ideas because they believe they need more experience, more resources, or the perfect moment. In reality, the perfect moment rarely arrives. The best way to learn is to begin with what you have and improve along the way.

It comes from repeatedly returning to your goal, even after setbacks, delays, or long breaks. Every challenge became an opportunity to learn something new, making both the project and me stronger with each iteration. Today, when I look back, I don't just see a game development startup. I see a journey of continuous learning, problem-solving, and personal growth. It has helped me become a better developer, a better teammate, and a more confident problem solver.

If there is one lesson I would like to share with fellow students, it is this: Don't wait for perfect circumstances to pursue your ideas. Start with the resources you have, learn from every challenge, and keep moving forward. Sometimes the smallest beginnings lead to the most meaningful journeys. One quote that has always inspired me throughout this journey is:

**"SIC PARVIS MAGNA"—
GREATNESS FROM SMALL BEGINNINGS.**



INTERNSHIP EXPERIENCES

HOW I BUILT AN AI RESUME SCREENER TO SOLVE REAL-WORLD HIRING CHALLENGES



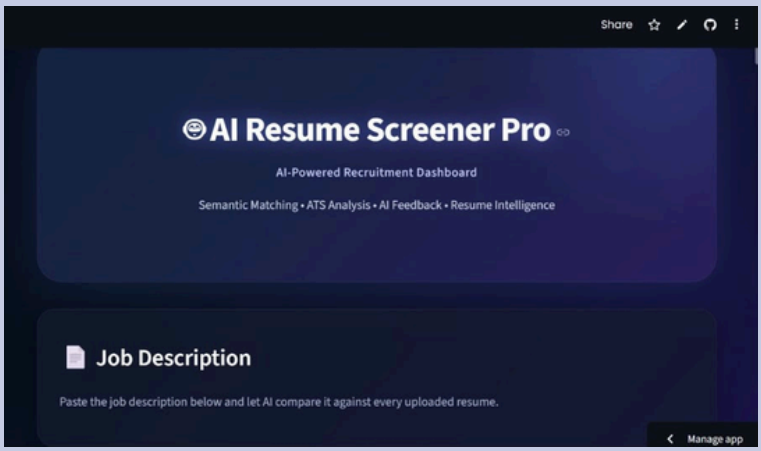
Shreya Hasija
IT, 5th Semester

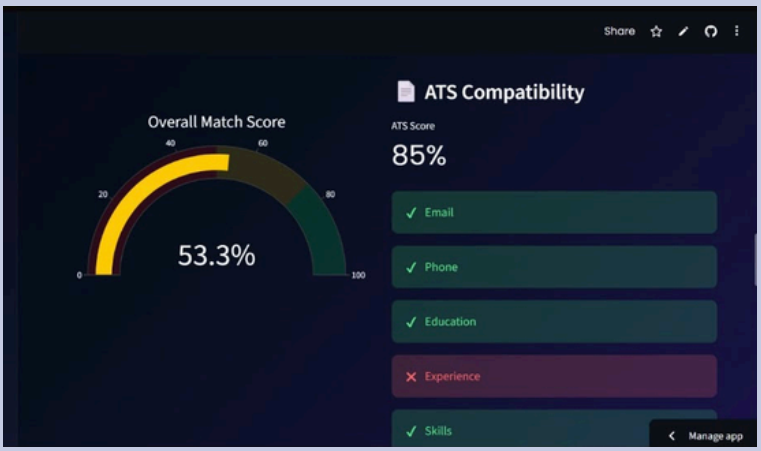
During my internship at Krivikisa Pvt. Ltd., I got the opportunity to work on a project that combined two things I'm really interested in: AI and solving real-world problems. I built an AI Resume Screener Pro, a web application designed to make the initial resume-screening process faster, better & more efficient. The idea was to help recruiters compare resumes with a given job description without having to manually go through every resume.

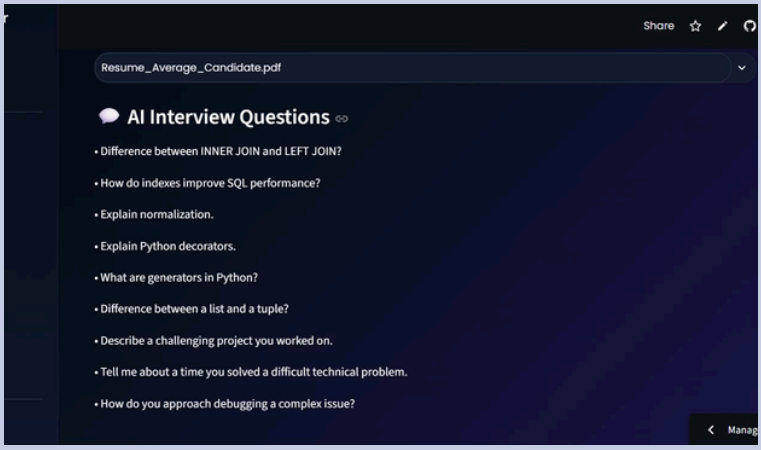
One of the main features I worked on was semantic matching using BERT, which allows the system to understand the context and meaning of a resume rather than relying only on exact keywords. The application also extracts skills, provides candidate scores and ATS analysis, and generates relevant interview questions.

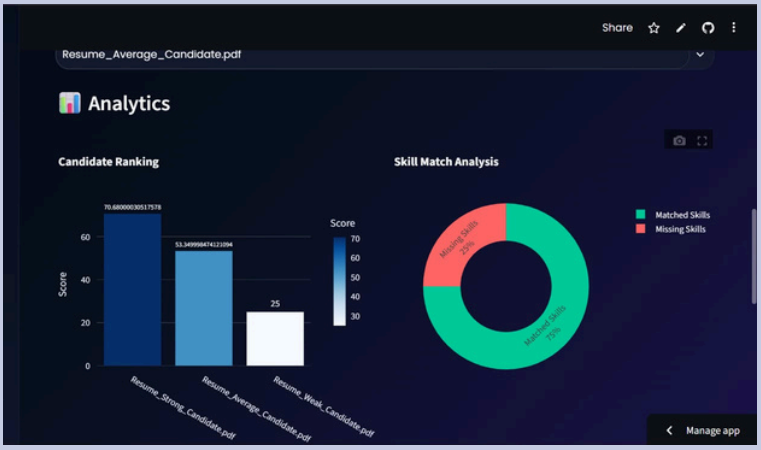
I built the application using Python, NLP, BERT and Streamlit, while also working on the UI to make the results simple and easy to understand.
What I enjoyed most about this project was seeing something I had learned theoretically actually turn into a working application. It also taught me that building an AI project isn't just about the model — making the technology practical, understandable and useful is equally important.

This internship gave me a much better understanding of how AI can be applied to everyday problems, and it definitely made me more interested in exploring what I can build with it next.









SIMULATING THE BOARDROOM: HOW I BUILT SKILLSPRINT AI USING LLMS



Priya
B.E.(I.T.), 5th Semester

During my internship at C-DAC Mohali, I had the opportunity to work on a project that combined two of my greatest interests—Artificial Intelligence and developing technology-driven solutions for real-world challenges.

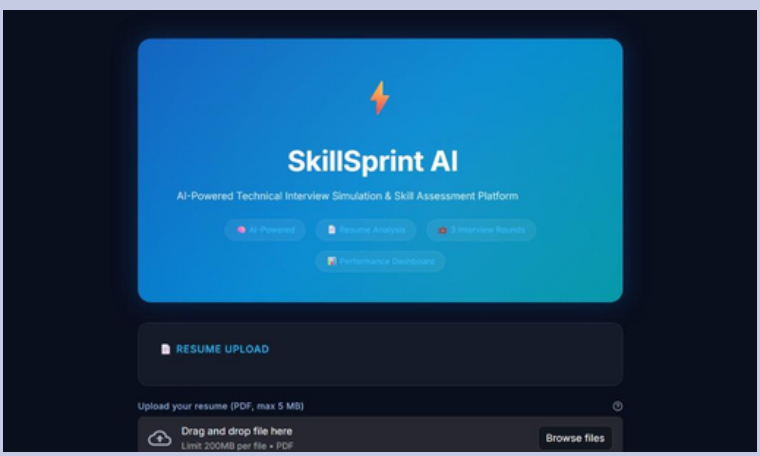
As part of my internship, I developed SkillSprint AI, an intelligent mock interview platform designed to make interview preparation more personalized, interactive, and effective. Unlike conventional mock interview systems that rely on a single interviewer, SkillSprint AI simulates a three-member interview panel comprising Technical, HR, and Managerial interviewers, providing candidates with comprehensive feedback from multiple perspectives.

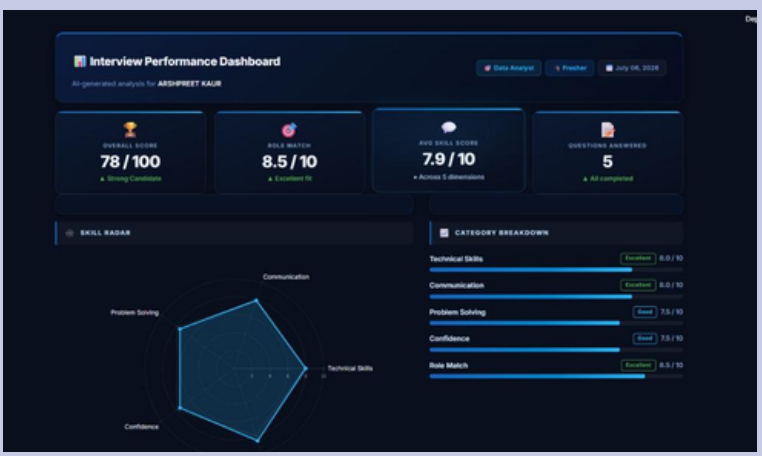
One of the key features of the platform is its AI-powered resume-based personalization, which analyzes a candidate's resume to generate role-specific interview questions. Using a Large Language Model (LLM) along with curated interview datasets, the application delivers personalized interview experiences, evaluates responses from each interviewer's perspective, and provides detailed performance analysis, structured feedback, and progress tracking across multiple sessions.

The project was developed using Python, Streamlit, Large Language Models (LLMs), SQLite, Plotly, PyPDF2, and python-docx. My primary contributions included backend development, implementing the resume upload module, question retrieval, session flow management, testing, and documentation.

Working on this project allowed me to transform theoretical concepts into a practical application that addresses a real-world challenge. More importantly, it taught me that building an effective AI solution is not only about intelligent models but also about creating systems that are practical, user-friendly, and impactful.

My internship at C-DAC Mohali was an enriching learning experience that strengthened my technical knowledge, enhanced my problem-solving skills, and deepened my understanding of AI-driven application development. It has further inspired me to continue exploring emerging technologies and building innovative solutions that create meaningful impact.





SENTINEL GO: THE FUTURE OF REAL-TIME CYBER DEFENSE WITH EBPF, GO, AND AI



Prashant Kumar Singh
B.E.(I.T.), 7th Semester



Over the past few years, the threat landscape in cybersecurity has undergone a fundamental transformation. For decades, endpoint security relied heavily on static indicators: file hashes, known signatures, explicit rules, and IP blacklists. This approach worked well when attacks were straightforward and easily categorized. Today, however, adversaries rely on far more discreet techniques. By utilizing living-off-the-land binaries, fileless malware, and rapid multi-stage exploits, modern attackers bypass legacy defenses without dropping traditional signature files on disk.

This shift inspired the creation of Sentinel Go, a research-oriented security platform designed for Linux systems. Sentinel Go operates as a behavioral anomaly detector at the operating system level, built by combining eBPF, Go, and Python-based machine learning. Rather than waiting for suspicious actions to surface in user space, Sentinel Go hooks directly into Linux kernel tracepoints to record process execution, file access, process creation, and network socket activity with microsecond-level precision—all without the overhead or risks associated with traditional kernel modules.

Once raw telemetry is captured at the kernel layer, it is passed into a concurrent Go processing pipeline. The pipeline cleans the noisy data and groups raw events into brief behavioral time windows. These structured features—tracking metrics like fork rates, process diversity, sensitive file hits, and new network connections—are then forwarded to a Python FastAPI service hosting an Isolation Forest model.

The core advantage of using an unsupervised model like Isolation Forest is that it bypasses the need for a pre-labeled malware dataset. Instead of searching for predefined attack patterns, the system continuously learns what normal, baseline behavior looks like on the specific host. Whenever a process or execution tree strays significantly from that baseline, Sentinel Go flags the anomaly and issues a real-time alert before the activity can develop into a full-scale security incident.

Building Sentinel Go highlights an essential lesson in modern security engineering: raw telemetry alone is insufficient. Security data only becomes valuable when it is structured, aggregated, and transformed into actionable decisions. The gap between raw kernel events and automated triage is precisely where modern defense thrives.

Looking forward, endpoint detection will inevitably become more behavior-aware and deeply integrated with kernel-level observability. Projects like Sentinel Go demonstrate how combining light footprint telemetry, system-level visibility, and adaptive AI offers a practical path toward resilient, signatureless defense. Modern attackers move fast, but by understanding behavior rather than waiting for signatures, modern defenders can stay ahead.

Project Repository: GitHub - Scorpio21Sec/Sentinel-Go

PLACEMENT TIPS







Shared by
Jasleen Kaur

I think interviews can feel a bit anxiety-inducing and nerve-wracking at first. However, after participating in numerous non-technical interviews—and recently appearing for a specific job profile—I would say it is more about vibing with the interviewer. It is about your ability to express yourself, your work, and your vision with clarity. In my opinion, flawless communication plays an important role alongside your skills and talent. Additionally, preparing in advance for non-technical aspects—like logical reasoning, puzzles, and HR or personality-based questions—is a key task. As far as my interview with ZS was concerned, the few minutes before it started were filled with anxiety, but the interviewer tried to make me feel comfortable. As soon as it kicked off, my anxiety was gone. I think what worked for me was my last-day revision of important concepts, combined with the support of my family, friends, and seniors.



Company: ZS
Role: BTSA

Interview Experience
Difficulty: Easy–Medium
The interview was mostly based on my resume, with no out-of-the-box questions.

Questions asked:
1. In-depth discussion on my projects
2. Questions about my internship project
3. Why a particular tool/framework
4. Reason for choosing my projects
5. Teamwork and leadership questions
6. Basic questions on JS, DBMS, SQL JOINS, DSA sliding window

Interview Tips:
1. Keep your communication skills sharp.
2. Don't fake anything. If you don't know something, don't mention it on your resume.
3. Have at least 2 strong projects worth discussing for 15 minutes.
4. Be prepared to explain everything mentioned on your resume.
5. CS fundamentals are important (OOPs, DBMS, OS, and Computer Networks).

Shared by
Ankit Kumar

I was a bit nervous initially because my CGPA wasn't as high as I wanted it to be. However, the moment the interview actually started, all that anxiety vanished. I found my rhythm and answered every question with complete confidence.

The biggest takeaway for me was that your profile doesn't need to match the job description 100%. What truly matters is absolute honesty and mastery over your own background. You must know every single detail written on your resume inside out and never try to bluff.

As for the interview itself, the panel was fantastic. Both interviewers were highly conversational and kept the environment very chill. In terms of technical depth, the Data Structures and Algorithms (DSA) portion was relatively light. Instead, they focused heavily on Core Computer Science concepts and threw in some genuinely tough logical puzzles. I was also grilled on the specifics of my project, along with several targeted questions on JavaScript.





Shared by Saransh

My interview experience for Policybazaar consisted of two rounds that heavily prioritized core computer science fundamentals and project depth.

The first round focused on my internships and projects, where the interviewer probed my architectural decisions—specifically why I chose FastAPI and LangGraph—before moving into a system design question about storing merchant email reports and tracking transaction failure rates. This led into a related SQL query task, followed by deep dives into DBMS, OS, and Data Structures, covering indexing, concurrency, synchronization, hashmaps, B+ trees, and how Policybazaar handles large production data.

The second round was a quick 15-minute session centered on OOPs concepts (abstract classes, interfaces, inheritance, and virtual functions), two DSA problems (string subsequence and an array window function), OS concepts like mutexes and semaphores, and a final puzzle. The only HR question asked was "Why Policybazaar?", making the overall experience a smooth process that focused on strong basic concepts and deep project knowledge.



Shared by
Shruti Tyagi

Placements can seem overwhelming at first, especially when you see the competition around you. However, after going through the recruitment process myself, I realized that success depends much more on preparation and consistency than on luck.

The first and most important step is to build strong fundamentals. No matter which role you are applying for, companies expect you to be comfortable with SQL, Python, databases, and analytical problem-solving. These are skills that repeatedly appear throughout online assessments, technical interviews, and project discussions. Instead of memorizing solutions, focus on understanding why a query works, how a program executes, and when to use different approaches.

For aptitude and logical thinking, I highly recommend practicing puzzles from GeeksforGeeks. Many companies test your ability to reason through unfamiliar situations rather than recall formulas. Along with puzzles, practice guesstimates. Start with questions related to your hometown or even UIET , for example, estimating the number of tea cups sold on campus in a day, the number of vehicles entering the university, or the annual electricity consumption of the college. These exercises improve structured thinking and teach you how to make logical assumptions under pressure.

Your projects are another crucial part of your resume. They do not have to be revolutionary or completely unique. A well-executed project with a clear problem statement is far more valuable than an overly ambitious project that you cannot explain. Before adding a project to your resume, compare it with the job description. If the role mentions technologies such as SQL, Python, AWS, Power BI, APIs, or machine learning, try to incorporate those technologies naturally into your project wherever appropriate. More importantly, be prepared to explain every technology you mention. Interviewers often spend more time discussing your projects than asking theoretical questions.

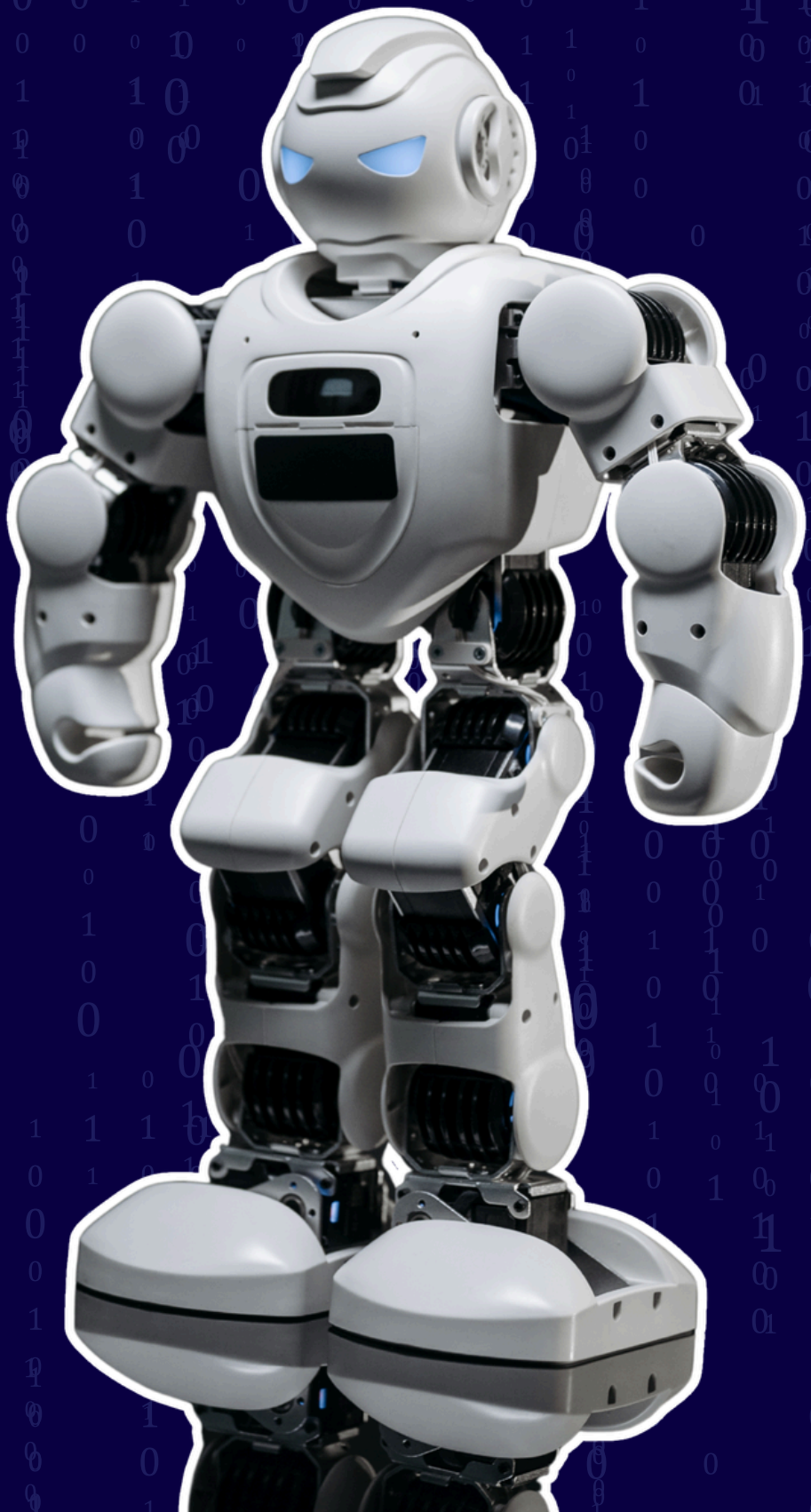
One mistake many students make is listing technologies they have only briefly used. If a technology appears on your resume, assume the interviewer can ask any question about it. Understand your project's architecture, design choices, challenges you faced, and the trade-offs behind your decisions. Being able to explain *why* you built something a certain way leaves a much stronger impression than simply naming tools.

Finally, communication matters as much as technical knowledge. During interviews, explain your thought process instead of rushing to the final answer. If you are solving a SQL problem or a guesstimate, walk the interviewer through your assumptions and reasoning. Even if your final answer is not perfect, demonstrating structured thinking and confidence often makes a positive impact.
Placements are not about knowing everything , they are about proving that you can learn, think logically, and solve problems. Stay consistent, keep improving your fundamentals, and treat every interview as a learning opportunity. With the right preparation and mindset, the right opportunity will eventually come your way.



Shared by
Mrityunjay

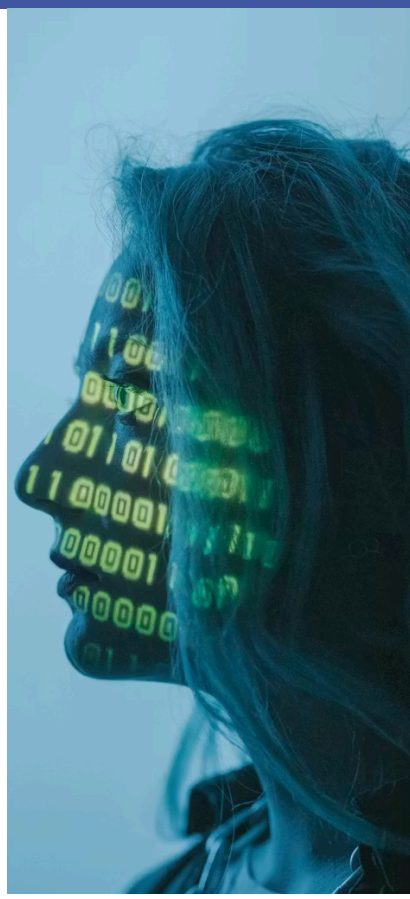
TECHNOLOGICAL ADVANCEMENTS



ENGINEERING MISOGYNY?

Pankhuri Bhatia
B.E.(I.T.), 7th Semester





AI, Algorithms and Optimizing Biases

Artificial Intelligence has become remarkably good at recognizing patterns. It predicts what we'll watch next, generates realistic images, recommends job candidates, and even carries conversations. Although these systems seem very different, they all work in a surprisingly similar way. They learn patterns from data and optimize for a specific goal. Sometimes that goal is engagement. Sometimes it's efficiency, accuracy or convenience.
None of those goals mention women. Yet some of AI's most controversial failures today disproportionately affect them.

The common explanation is that AI is "biased." While that is partly true, I think it misses a deeper context. Engineers don't necessarily build systems intending to discriminate. Instead, they teach models to optimize a goal using data collected from the real world. If that world already contains inequality, AI doesn't simply inherit it. It can repeat it again and again, consistently and at scale.

The real question isn't whether AI is misogynistic. It's whether the things we've asked AI to optimize were ever neutral in the first place.

Who Owns Your Face?

For most of photography's history, cameras captured reality. If a photograph existed, it usually meant something had actually happened.

Generative AI changes that relationship completely. Modern image-generation models, called diffusion models, learn what faces, expressions and lighting look like by training on millions of images. Instead of editing an existing photograph, they can generate an entirely new one that looks real. Your face is no longer just something that can be photographed. It becomes data that can be recreated, altered and placed into situations that never existed.

At the same time, devices like Meta's AI smart glasses are making cameras smaller, smarter and easier to wear throughout the day. Computer vision models running on these devices can recognize objects, landmarks and even help describe what the wearer is seeing. Individually, these technologies are impressive. Together, they raise difficult questions about identity itself. They make collecting facial data almost effortless, often without the subject ever realizing they have been recorded.

Deepfakes disproportionately affect women, not because the technology targets them, but because it dramatically lowers the cost of creating non-consensual imagery. The technology doesn't care whether someone is famous. Anyone with a few photographs online can become a target. Privacy once meant controlling who could see you. Today, it increasingly means controlling who can recreate you.

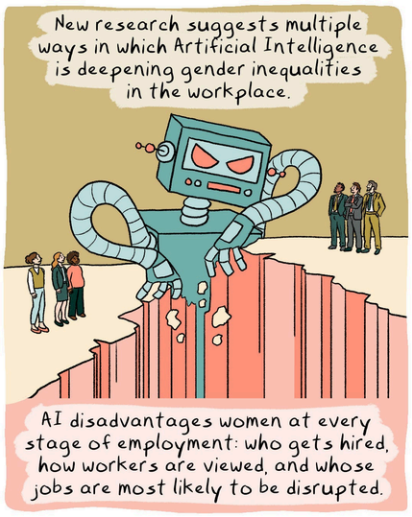
What is the Algorithm Learning to Reward?

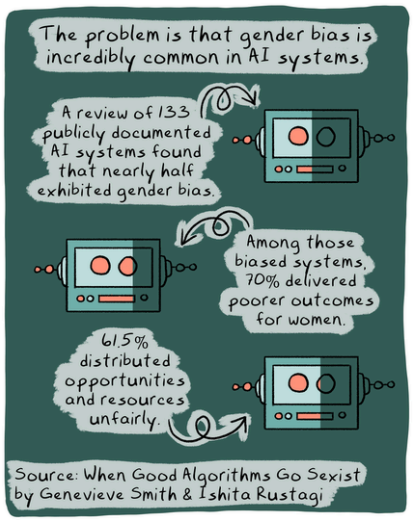
Recommendation algorithms don't watch videos or read posts the way we do. They don't understand sexism, politics or morality. They simply look for patterns in behaviour.

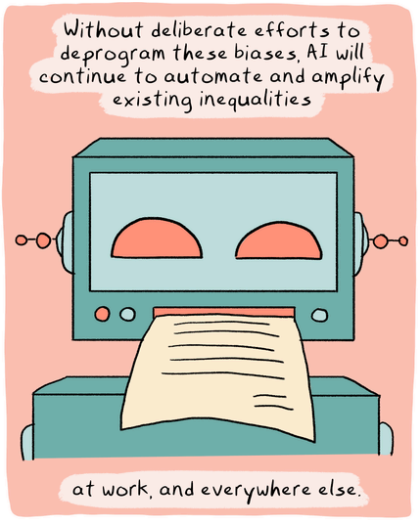
Every time you pause on a reel, like a post or watch a video until the end, that interaction becomes training data. The algorithm learns which pieces of content are most likely to keep someone scrolling. Its goal is simple: predict what you'll engage with next. Each interaction becomes another data point. Across millions of users, these tiny signals are enough for the model to predict what people are most likely to consume next.

That sounds harmless until you ask what people actually engage with. Strong emotions such as outrage, fear and controversy tend to hold attention longer than calm discussion. If users who watch one type of content consistently move towards more extreme material, the model quietly learns that pattern.

Nobody explicitly programmed misogyny into the system. The algorithm simply became very good at finding the fastest route to another click.







Can AI Learn Without Learning Our Biases?

Every machine learning model begins the same way: it learns from examples. Whether the task is detecting cancer, predicting fraud or screening resumes, the model searches for patterns in historical data. During training, it adjusts millions of internal parameters until it becomes better at making predictions on new examples.

Most of the time, this works well. People, however, are not as predictable as cats or traffic signs. Historical datasets don't just record facts. They also record decades of unequal opportunities, social norms and human decisions. If women were consistently underrepresented in leadership roles, promoted less often or diagnosed later for certain medical conditions, those patterns become part of the data itself.

The model cannot tell the difference between a useful pattern and an unfair one. To the algorithm, both simply improve prediction. This is why the phrase "AI is only as good as its data" is incomplete. The data may be perfectly accurate. That doesn't mean the context behind it is fair.
We already know that bias exists in data. What comes next depends on how we choose to build AI. It can reproduce historical patterns, or we can design it to account for them in pursuit of fairer outcomes.

Why Are Voice Assistants Still Female?

More than a decade after Siri and Alexa entered our homes, one question still hasn't disappeared.
Why do so many voice assistants sound female? Speech AI has improved enormously. Modern systems understand multiple languages, regional accents and natural conversation. Large Language Models generate responses that feel increasingly human. Yet the default personality has changed remarkably little: calm, patient, polite and endlessly available.

Companies often argue that users find female voices warmer and easier to trust. But defaults matter. When billions of interactions reinforce the image of a female assistant whose primary role is to serve, apologize and help, technology isn't just responding to social expectations. It's also quietly reinforcing them. Defaults are rarely just technical decisions. Over time, they become part of what society begins to see as normal.

Conclusion
AI technologies were not built to spread misogyny. Generative AI creates images, recommendation algorithms maximize engagement, and voice assistants simplify tasks. The issue is that humans, not AI, define what these systems optimize.

When design choices rely on societal biases, AI scales those prejudices with massive efficiency. Engineering misogyny rarely involves writing explicitly sexist code. Instead, well-intentioned systems repeat historical assumptions so perfectly that bias starts to look like objective truth.

As AI integrates deeper into daily life, the critical question is no longer how intelligent our systems can become, but whether we are teaching them to optimize the right things.

INNOVATION THRIVES ON TRUST, AND BLOCKCHAIN IS REDEFINING TRUST IN THE DIGITAL WORLD.

Written by
Priya
B.E.(I.T.), 5th Semester





As digital technologies continue to evolve, Blockchain has emerged as one of the most transformative innovations, offering a secure, transparent, and decentralized approach to managing and exchanging information. Originally introduced as the underlying technology for cryptocurrencies, blockchain has now expanded into numerous industries, driving innovation far beyond digital finance.

Blockchain is a distributed ledger technology that records transactions across a network of computers, ensuring that data cannot be altered without the consensus of the network. Unlike traditional centralized systems, blockchain eliminates the need for a single controlling authority, making transactions more secure, transparent, and resistant to fraud or unauthorized modifications.

The applications of blockchain extend across sectors such as finance, healthcare, supply chain management, education, government services, and cybersecurity. It enables secure digital payments, transparent product tracking, tamper-proof medical records, digital identity management, smart contracts, and efficient record-keeping. By enhancing trust and accountability, blockchain is helping organizations improve operational efficiency while reducing costs and risks.

The technology is built on key concepts such as decentralization, cryptographic security, consensus mechanisms, distributed ledgers, and smart contracts. Together, these components create a reliable system where data integrity is maintained, transactions are verified without intermediaries, and information remains accessible yet highly secure.

Studying blockchain demonstrates how technology can address real-world challenges by ensuring transparency, security, and trust in digital ecosystems. As organizations increasingly adopt decentralized solutions, blockchain is expected to play a significant role in shaping the future of digital transformation and secure information management.

Learning about blockchain has broadened my understanding of emerging technologies and strengthened my appreciation for secure, innovative systems that solve practical problems. It has inspired me to continue exploring advanced technologies and contribute to building solutions that are efficient, reliable, and impactful.

HACKATHON EXPERIENCES



FROM CHAOS TO FIRST WIN: HOW WE BUILT A SMARTER HIRING PLATFORM AT TECHNOVATE 3.0

Krishna Raghwani
B.E.(I.T.), 5th Semester

The Genesis

The hiring landscape for students today is broken — and we felt it every day. **LinkedIn, once a powerful job portal, has become a noisy feed of AI-generated posts, inflated achievements, and copy-pasted bios.** Every second post screams "Thrilled to announce…" with the same ChatGPT tone.
Meanwhile, real recruiters are buried under 10,000 applications for a single legitimate role, and developers like us are juggling five separate platforms — GitHub, Kaggle, LeetCode, Codeforces — with no single place to show who we truly are.
That frustration became our problem statement, and Technovate 3.0 became our stage.

The Stage: Technovate 3.0

The Hackathon Technovate 3.0, organized by Google Developer Groups on Campus (GDGC), UIET Panjab University, was a 24-hour open innovation hackathon with knockout rounds and a rigorous judging panel.
We got the news just 7 days before the event. With open innovation as the theme, my team — Binary Bees — started brainstorming immediately. The problem we chose was one we lived: a fragmented, overhyped, and inefficient hiring ecosystem for students and developers.
Team Binary Bees deep in the build, 24 hours in.



The Problem We Solved

- LinkedIn has drifted from a job portal to a performance platform full of exaggerated credentials and AI-generated content.
- Legitimate job postings receive thousands of applications, most from ineligible candidates.
- Developers must maintain multiple profiles (GitHub, Kaggle, LeetCode, Codeforces) with no unified system.
- Recruiters have to manually visit resumes, portfolios, and multiple platforms before shortlisting a single candidate.



Our Solution

We built a full-stack platform with two portals — one for candidates and one for companies — designed to make hiring honest, efficient, and skill-verified.

Candidate Features:

- **AI Skill Extractor** — Every project, certificate, or course you add is parsed by AI to extract and store verified skills.
- **Smart Profile Allocation** — Based on extracted skills, the platform recommends the profile type that best fits the candidate.
- **Task Tracker & Goal Achiever** — With streak integrations across GitHub, Kaggle, and other platforms, candidates can maintain consistency and track growth in one place.

- **Structured Portfolio** — A fixed-format portfolio where candidates upload GitHub repos and live projects with README-style explanations.
- **Global & Per-Skill Ranking** — Skills are weighted by projects, verified certificates, and contributions across platforms, giving each candidate a transparent, merit-based rank.
- **Smart Job Notifications** — Candidates are notified only for roles they are eligible for and open to — based on skills, location, remote preferences, and company requirements.
- **AI Resume Maker** — Generates an ATS-friendly resume from the candidate's profile data automatically.
- **AI Profile Recommendations** — Suggests improvements to the candidate's profile based on target job requirements.

HOW TO DOMINATE YOUR NEXT HACKATHON

- **Build a Diverse Team:** Do not just recruit coders. A winning team needs a designer for visual appeal, a developer for building, and a charismatic presenter to sell the idea.
- **Solve a Real Problem:** Judges favor practical utility over complex, unfinished tech. Find a specific pain point and build a direct solution.
- **Scope Down Ruthlessly:** You only have 24 to 48 hours. Build a working Minimum Viable Product (MVP) with one flawless core feature rather than five broken ones.
- **Design the Illusion:** If a feature is too complex to code in time, mock it up. The user interface (UI) and user experience (UX) must look polished and functional during the demo.
- **Master the Pitch:** A brilliant project fails without a great story. Dedicate the last 3 hours purely to practicing the presentation, focusing on the problem, your solution, and the business value.

Company Features:

- HR portals to post jobs with specific filters — CGPA, ATS score, required skills.
- Candidates are auto-ranked and shortlisted based on how well they match the job requirements.
- Upon selection, the work experience is added to the candidate's profile directly by the company — a verified proof of employment.
- Referral System — Selected candidates can open referral tickets; other candidates approach them, and referrals are given based on ranking, not favouritism.
- AI-based sorting for both candidates and HR across all sections of the platform.

The Night That Changed Everything

Building this in 24 hours was not just a technical challenge — it was a test of everything. I had been preparing for days, working 18-hour stretches, burning through espresso shots and Red Bulls.
The night of the hackathon, running on zero sleep for nearly 48 hours, I almost convinced myself we wouldn't make it. I was in my second year, still figuring things out, and the self-doubt was loud.
Then I called my parents. That conversation quietly reset something in me. I put the doubt aside, went back to my screen, and just worked — no anxiety about the result, just the work itself.
The next morning, when they called Binary Bees to the stage, I didn't process it immediately. It took a second. And then it hit — my first win.

What I Took Away

- Hackathons are not just about code. They are about identifying real problems, staying calm under pressure, and trusting your team.
- A special thanks to the judges — Amanpreet Kaur, Kanishk Tyagi, Simar Preet Singh, Ayush Kumar Tiwari, and Veer Pratap Singh — for the sharp, honest feedback that pushed us to refine our thinking.
- And to my team — Arshi Khan, Arsh Khan, and Aamna Goyal — this win belongs to all of us equally.
- This is just the beginning.

GLIMPSES OF THE CHANDIGARH POLICE HACKATHON 2026

Organised by Chandigarh Police In association with:
University Institute of Engineering and Technology (UIET), Panjab University &
Punjab Engineering College (PEC), Chandigarh













CREDITS & ACKNOWLEDGMENTS

A Special Thanks to Our Faculty Mentor

We extend our deepest gratitude to our mentors, **Prof. Dr. Amandeep Verma (Coordinator, Information Technology) and Prof. Dr. Mandeep Kaur (Faculty incharge, Tech Magazine)** for their unwavering guidance, profound technical insights, and constant encouragement throughout the creation of this magazine. Thank you for inspiring us to look beyond the code and push the boundaries of our creativity.

The Core Team

Editor: Naeva Setia
Co-Editor: Vrinda Dubey
3rd Year Lead: Dhruva Vermani

Data Collections Team: Mohit, Gaurav
Tech & Research Team: Balbina, Shruti, Saman



Department of Information Technology
University Institute of Engineering and Technology
South Campus, Panjab University
Sector 25, Chandigarh

That's a wrap on this edition of our Tech Magazine. Thanks for reading! We put our hearts into these pages, but we're always looking for the next big idea and it might be yours.

Got an experience to share, or just a question about what we do? Drop us a line. We read every single message.

Email us at: iteditorialuiet@gmail.com
Instagram: iteditorial_uiet