

C
S
E
S

A CSES INITIATIVE FOR SOFTWARE ENTHUSIASTS

THE SUBTLE ART OF GETTING A J**B**

A COMPREHENSIVE GUIDE ON
INTERNSHIP & PLACEMENT
STRATEGIES

Table Of Contents

Chapter 1: Welcome to the Chaos

Chapter 2: Internships: Because Apparently College isn't Enough

Chapter 3: Preparation: The Part Everyone Wants To Skip

Chapter 4: The Intern Drive: Rules of the Game

Chapter 5: Your Resume: A Document you'll Agonize Over and They'll Skim in 30 Seconds

Chapter 6: Interviews: The Art of Selling Yourself

Chapter 7: Applying Off-campus: You're on Your Own Now

Chapter 8: Advice You Probably Don't Wanna Hear

Chapter 9: You Got In! Now Try Not To Blow it Up

Chapter 10: Post Intern: You Don't Get To Relax Yet

Credits

Disclaimer

This guide is a student-led initiative by members of the Computer Science and Engineering Society (CSES), built from our experiences, peer insights, and surviving the intern drive in one piece. It is not affiliated with, endorsed by, or officially associated with the National Institute of Technology, Warangal, its administration, or any academic department. Take it as advice from seniors, not an official rulebook.

CHAPTER

Welcome to the Chaos

If you're opening this right now, chances are you've either:

- suddenly realised everyone around you is talking about “the drive” like it’s a boss battle in a video game,
- heard seniors whisper “And so it begins...” with dramatic expressions,
- or your WhatsApp is exploding with words like CCPD, shortlist, OT, and you’re wondering whether you accidentally enrolled in a reality show.

The intern drive feels mysterious at first. One minute you’re minding your own business, the next minute you’re filling out forms at lightning speed, checking the CCPD group like it’s your blood pressure, and praying for shortlists like festival offers.

This ebook aims to decode the chaos and to help you understand what actually happens, what to expect, what to ignore, and how to survive without crying into your laptop at 3 AM. (But if that happens... It's fine. After all, it's tradition.)

By the time you finish this, you’ll hopefully be thinking,

“Okay, this is manageable (*maybe*).”

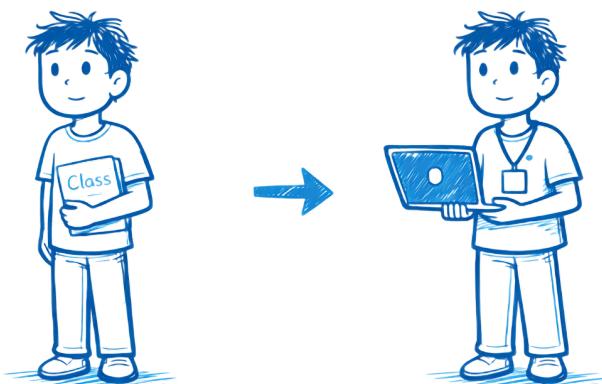


CHAPTER

Internships

Because Apparently College Isn't Enough

An [Internship](#) is basically this magical phase when a company lets you enter their world, gives you real work, and pays you for it too (the money is nice, but let's pretend we're doing this for the “learning”). Think of it as a short-term adventure where you get to stop being a student for a bit and start acting like someone who knows what they’re doing. Spoiler: no one actually knows what they’re doing at first.



You work on real-world projects, sit in meetings, and use practical tools. It’s like the industry saying, “Come on in, kid, let’s show you how things really work,” and then ever so gently pushing you into the deep end with Jira tickets and Git commits.

And honestly, it’s important, not in the “change your life overnight” way, but in the “oh wow, so this is how the real world functions” way. You finally get to see the difference between what’s taught in class and what companies actually do. Concepts that once felt abstract now have purpose, and you start picking up new technologies, tools, and skills that make your resume look like it’s been working out a ton.

You also get to meet professionals who’ve been doing this for years and sometimes even impress them enough that they remember your name (a rare achievement, treasure it).

These connections often become valuable later, especially when you're preparing for placements and can proudly say you've done something hands-on.

In short, an internship is your sneak peek into corporate life, a mini-career trial run, and a pretty solid confidence booster.

CHAPTER

Preparation

The Part Everyone Wants to Skip

But why do companies even care about all this in the first place? How does DSA matter when nobody sits and writes sorting algorithms after joining a company? It is a fair question, and honestly, almost everyone has this doubt at some point.

What companies are really trying to figure out is whether you can learn on the job. The tech you'll be working with six months later will be completely different from anything you know today. So DSA is less about remembering every algorithm in existence and more about checking how you think when you're stuck. Can you break a problem down and slowly pull it apart into something solvable instead of panicking? Can you reason your way to a solution rather than guessing? Can you handle curveballs that you didn't prepare for?

These skills matter much more than memorizing the solution to the N-Queens problem without understanding the underlying logic.

Same thing with project discussions. A lot of people panic thinking interviewers expect them to know every library, framework, or internal mechanism in detail.

But that's not what they're checking for.

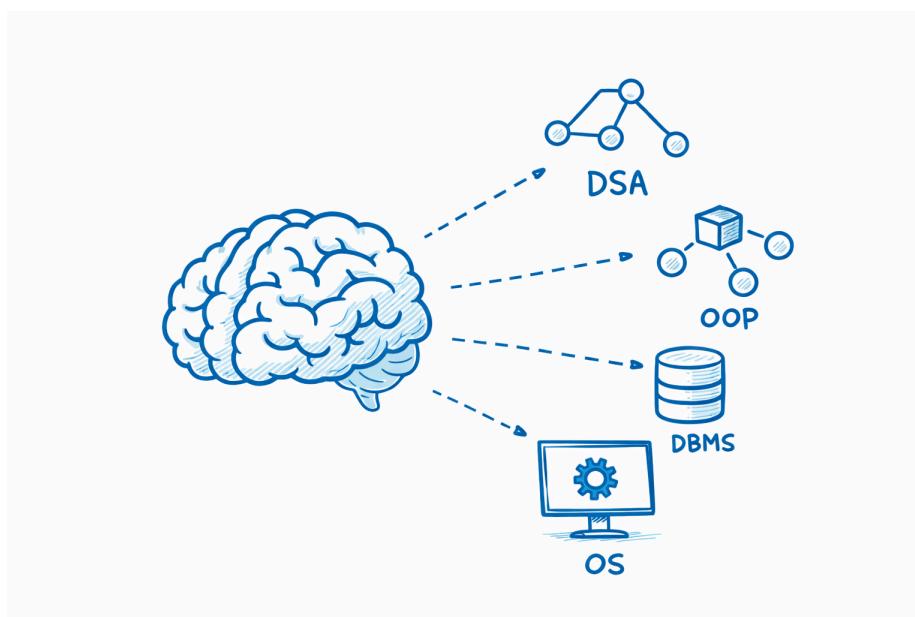
We're all college students, not researchers with 15+ years of experience. What they actually want to know is whether you really worked on what you built. Did you blindly follow a tutorial, or did you take decisions, fix issues, understand the trade-offs, and build it because you knew what problem you were solving? Your project is proof that you can take ownership, think independently, and debug your way through the unknown.

And then there are the CS fundamentals: DSA, OOPs, DBMS, and OS.

I know! They feel like boring theoretical courses that college forces on us. But here's the reality: they're literally the foundation of everything we build!

You can't make a real-world system without a database, and you need DBMS concepts to design one properly. You can't write scalable, clean software without understanding OOPs. And the second you deploy something, it's running on an operating system, so knowing OS basics like memory, processes, and threads means you'll write code that actually survives instead of crashing the moment it gets real traffic.

Fundamentals are what make you reliable. The kind of developer whose code doesn't collapse like a Jenga tower every time someone adds a feature.



Subjects to Study: Your Core Toolkit

Cracking OTs and interviews isn't just about speed or luck. It's about having a solid grip on the core subjects that show up everywhere, sometimes loudly, sometimes quietly, sometimes disguised as "just one small MCQ."

I. Data Structures & Algorithms (DSA)

First, and most obviously, comes DSA. It shows up in almost every online test and interview. Nearly every company places its strongest focus here, and it is the section that ends up carrying the most weight throughout the process.

The path to mastery in DSA is consistency. One good question a day beats 50 questions skimmed the night before the interview or OT. Don't speed run through questions by just reading solutions. Actually try solving them yourself, even if it takes a long time. And no, skipping graphs or DP doesn't magically make them disappear from the interviewer's mind :(

For practicing DSA, there's no "best" sheet. Striver's sheet is popular, LeetCode works well, and any structured list is fine as long as you finish it thoroughly.

Focus on: arrays, strings, recursion, linked lists, stacks, queues, trees, graphs, DP, greedy, hashing, heaps, sliding window, binary search.

Resources: [LeetCode](#), [Striver's Sheet](#), [Love Babbar's 450](#)

Competitive Programming

Competitive programming is not a separate skill from DSA. It is a way to practice solving DSA problems under pressure.

When you practice CP, you are training yourself to think clearly with time constraints, limited attempts, and unfamiliar problem statements. This closely mirrors interview conditions, where stress and time pressure often matter just as much as knowing the concept itself. CP helps you stay calm, identify patterns faster, and avoid freezing when a problem looks unfamiliar.

Over time, this habit of solving problems under pressure builds confidence. You stop panicking when you do not see the solution immediately and instead focus on breaking the problem down logically, which is exactly what interviews demand.

If you achieve a decent rating or rank, you can also mention it on your resume as proof of consistent problem solving.

Good platforms to practice include [Codeforces](#), [HackerRank](#), the [USACO Guide](#), and the [CSES Problem Set](#) (not us... unfortunately). You can also participate in CSES' (that's us!) year-long contests, CodeMon. CodeMon helps you practice regularly and gives you a realistic idea of where you stand among your peers in the batch.



II. Object-Oriented Programming (OOPs)

OOPs appears frequently in both Online Tests (OTs) and interviews, often in subtle ways. Interviewers are not interested in textbook definitions. They want to know whether you actually understand how these concepts are used in real code.

Instead of memorizing definitions, focus on being able to explain ideas with simple examples. You should be comfortable describing how you would structure a program, why you would choose a particular design, and how different objects interact with each other.

Focus on: encapsulation, abstraction, inheritance, polymorphism, classes and objects, overloading versus overriding, interfaces, abstract classes, and threads.

Resources: [TPoint Tech](#), [GeeksforGeeks](#), OOPs notes, course material, Java or Python documentation.

III. Database Management Systems (DBMS)

DBMS is a favourite in OTs and a frequent guest in interviews. It's also the easiest subject to showcase, just having a small DBMS project on your resume instantly signals that you understand how real applications store and manage data.

SQL is non-negotiable. You should know how queries work, how joins combine data, and why normalization exists. Interviewers often ask questions that test whether you understand what happens behind the scenes, not just whether you can write a query that works.

Understanding transactions, ACID properties, and indexing helps you explain why certain systems are reliable and efficient while others fail under load. These concepts also come up frequently in MCQs and interview discussions.

Focus on: SQL queries, joins, normalization, transactions, ACID, indexing, ER diagrams.

Resources: LeetCode SQL, Hackerrank SQL, GFG, DBMS notes.

IV. Operating Systems (OS)

OS pops up occasionally in online tests and sometimes in interviews too, so try to include this topic in your internship preparation. Even a basic understanding of OS concepts can help you respond more confidently and clearly when these topics come up during discussions.

If OS is not part of your syllabus yet, that is completely fine. If a question comes up during an interview, be honest about it. Interviewers generally appreciate honesty more than guessing.

Focus on: scheduling algorithms, processes vs threads, deadlocks, memory management, synchronization.

Resources: [Neso Academy playlists](#), GFG OS concepts, your course notes if available.

At the end of the day, these subjects are not meant to scare you. They exist because they genuinely make you a better engineer. Think of them as your skill tree. Build it slowly, revisit it often, and strengthen it over time.

Study steadily, revise smartly, and remember that the goal is not perfection.

The goal is clarity, confidence, and walking into an OT or interview knowing you have put in the work.

PROJECTS

Projects are a way to show the recruiter that you actually know your stuff, and haven't just been writing exams for the past two years. They let you demonstrate your skills, and actually build something meaningful and practical.

Every interviewer has seen the usual projects like to do apps, calculators, and generic clones. To actually catch their eye, you must truly stand out. Here's the secret: The best project ideas come from your everyday frustrations. Think about problems you face during your routine and try to build a solution for them. Projects that solve real world problems naturally stand out and often lead to better interview discussions.

A lot of people fall into the trap of doing online courses to get various certifications, and stuff their resume with them. While there is nothing wrong with doing this, the problem arises when they just stop there. Instead of just following a course end-to-end, try to apply what you learn by building something with it. This shows that you did not just watch videos for a certificate, but actually understood the concepts and knew how to use them.

And if you follow a tutorial for a project, be sure to NOT blindly copy it. The interviewer will know. Understand what is happening, build your own version of the project and improve upon it. This also helps during interviews when you are asked about your thought process, design choices, or why you implemented certain features the way you did.

If you are interested in any specific sub-field of Computer Science, you can always build a project in that domain. This will give you an in-depth understanding of the domain, and in the end, you'll have something to show for it.

If you're not sure what you're interested in yet, a good starting point is web development or machine learning. These areas are well explored, beginner friendly, and many seniors have successfully built projects in them.

In the end, just remember to make something that feels like yours. That's the whole point!

CHAPTER

The Intern Drive

Rules of the Game

Now that you're done with your intern prep (hopefully), you're all set to begin! Firstly, you'll be added to your branch's CCPD group. Here, you'll receive all information regarding the companies coming to campus, their application deadlines and other important information. Do not ignore any message in this group.

So it begins...

A message will be shared on the group, whenever a company comes. This will include their job description, as well as eligibility criteria (CGPA, branch, etc.).



This brings us to the first criteria for eligibility (and the most dreaded): CGPA. Your CGPA is always the first thing that is checked while applying to any company. For freshers, the right time to start maximising your CGPA is in your first semester. For the rest of us, it's right now. Try your best to score high, and increase your CGPA as much as possible. It is tough, with constant assignments, projects, minors and whatnot, but none of that matters to the company. All they will see are those 3 digits, and decide whether you are eligible or not.

The CGPA cutoff varies from company to company. Most of them keep 7 or 7.5 as their cutoff, while some keep 8, this is rare, however. There are also some companies that have a cutoff of 6.5, 6 or even no cutoff!

I know I just spent the last paragraph rambling about your CGPA, but do make note of this: High CGPA does not guarantee you a good internship. CGPA is vital for your eligibility, but plays almost no role beyond that. After this point, your skills and projects matter a lot more.

Now, to all non-circuital branch students. You might get disheartened seeing a lot of companies coming with no branch criteria, but still select only circuital/CSE students. Just remember, the entire intern drive is a wait game. Eventually, something will come for you, just be patient. The harsh reality is that it simply is tougher for you guys. So grab every opportunity you can, and apply off-campus as well. Don't entirely rely on the campus drive.

Few companies also consider students pursuing a minor degree in CS/Maths along with circuital students.

With each company, the CCPD will also share relevant form(s) to be filled up for their role. Some companies require multiple forms to be filled, and they may pop up at random times with a very short deadline. Ensure that you fill them without fail in the given time frame.

After this, each company hosts a [Pre-Placement Talk](#) (abbreviated to PPT). Usually held online, employees host the meeting and tell you what the company actually does, and what to expect from the role. They are like a distant relative's wedding, it is not the most fun, but you will be made to attend. In a lot of interviews, companies ask you "Why this company/role". The PPT will enable you to easily answer such questions. Alongside this, some companies also talk about their OT format, which could give you an advantage as you know what to expect.

Online Test (OT): The Starting Point

The OT is usually the first real hurdle of the internship process. It is your first chance to show your abilities, and every company does it differently. Most tests are for one to two hours and are primarily focused on [DSA-based coding](#) questions, which form the core of most shortlists. Some companies design their OTs as a mix of DSA problems and [MCQs](#) from CS fundamentals like Operating System (OS), Computer Networks (CN), DBMS, SQL, or even ML, while others go entirely MCQ-based, include HR-style situational questions, or come up with their own weird combinations. So it helps to be mentally prepared for a wide range of question types.

Most platforms allow multiple programming languages, but the default language is not always the one you intend to use. The problem arises when you excitedly start coding, only to realise ten minutes later that the editor has been set to Java when you meant to write in Python. Switching languages at that point is painful, and also since copy-pasting is risky on most OT platforms, rewriting your code to correct this mistake costs more time than you'd expect. A quick check at the beginning saves you from this spiral.

Some coding questions come wrapped in long stories about kingdoms, factories, or families in distress. Do not get emotionally invested, extract the logic and move on.

If you get stuck, write a partial solution. Most platforms award partial marks if it passes a few test cases, and many shortlists are decided by those one or two hidden test cases. A partially correct solution is far better than nothing.

Attempt the aptitude questions carefully. Many students rush through them as they are easy, only to make silly mistakes and lose marks unnecessarily. A calm, composed, and steady approach works best.

OTs can feel straightforward or unpredictable, but the best approach is to stay calm, stay focused, and keep moving. Not being shortlisted is rarely a reflection of your capability. Luck often plays a larger role than people like to admit.

The Shortlist Phase: When Things Start Moving Quickly

Once your name appears on a shortlist, everything starts moving quickly. Interviews are scheduled within a couple of days, or sometimes even sooner. CCPD will share the necessary instructions like online/offline, timings, venue, etc.



You are also assigned a SPOC (Single Point of Contact). Think of them as your Gandalf, the guide who appears whenever the journey becomes chaotic. For each company, you'll be given an SPOC whom you can reach out to if you have any questions. If your SPOC is unavailable, you can reach out to any of the other CCPD representatives as well.

Before your interview, have a printout of your resume, your ID card, and a pen with you. For online interviews, ensure your laptop is charged and your internet stable. If the round is offline, arrive early and look presentable.

Since you have already submitted your resume to the company earlier, avoid making major changes to it right before the interview. Recruiters often bring the version you submitted initially, and large discrepancies can create avoidable confusion. Minor updates are fine, but reshaping the entire resume at the last moment is not advisable.

This stage can feel hectic, but it is also where the process becomes exciting!

The Interview Stage: Where Everything Counts

Interviews are the most dynamic and unpredictable part of the internship process. Some companies conduct two or three technical rounds followed by an HR round, while others merge everything into a single session. There is no fixed format, and even different panels within the same company can run interviews very differently.

One candidate might be asked pure DSA questions, another might be questioned almost entirely on projects, and someone else might get a mix of technical and HR discussions. The focus depends heavily on the interviewer and the role, which is why experiences can vary so widely.

Interviews are usually scheduled on short notice and move quickly, especially when multiple rounds are involved. The process can feel intense and chaotic, but that is normal. Many candidates walk out of interviews convinced they underperformed, only to be selected later. You are rarely the best judge of how an interview actually went.

The Final Shortlist: The Waiting Period

After the interviews, the waiting phase begins. It often feels slower than the rest of the process combined. During this time, avoid assuming anything. Continue applying and taking other tests, even if you feel your interview was perfect.



The opportunity that works out is often the one you least expect.

When the final list arrives, I hope your name is there. If not, it is important not to be discouraged, and remember that the rejection isn't personal. Companies have their own selection criteria, which we are unaware of.

Whatever the outcome, this is only one part of a much longer journey. There will always be more chances, and often the next opportunity turns out to be the right one.

CHAPTER

Your Resume

A Document You'll Agonize Over and They'll Skim in 30 Seconds

Okay, so we've finally reached the resume section. It sounds serious, it sounds official, and it definitely gives off the vibe of something meant for older, responsible adults who drink black coffee and read financial newspapers at 7 AM. Meanwhile, you're sitting here wondering how you are suddenly expected to whip out a polished document summarising your entire (non-existent) professional experience from thin air. You don't know what to include or whether you even have enough to fill a single page, and honestly, that confusion is completely normal!

A [resume](#) is simply a one-page document of what a company needs to know about. It should just have your education, the skills you've picked up, the projects you've built, any experience you have, and a few key details that show what you're capable of.



When you stop treating it like this big, intimidating document and see it for what it really is, it's just a neat, structured way to present what you can do at this point of time.

Your resume gets used multiple times during the intern drive for shortlisting. The first time is when CCPD circulates the initial application form for each company's role. Every company asks you to upload your resume before the OT. Most of the companies don't look at it too deeply here, but occasionally some do use it to shortlist who gets to write the test.

It appears again after the OT when companies prepare their interview shortlists. If many students perform similarly, the resume becomes a useful filtering criteria to decide who moves forward.

And then, of course, comes the [interview](#). This is where your resume becomes really important. Interviewers typically begin by skimming through your resume and use it to start the conversation. Whatever you include in your resume shapes how your interview unfolds. If you mention a project or skill, be prepared to be asked about it. Every line you include should be something you can explain with confidence.

This is exactly why a resume matters, because it becomes the foundation of your interviews. Your resume becomes their starting point for understanding your interests, your strengths, and the areas you've actually worked on.

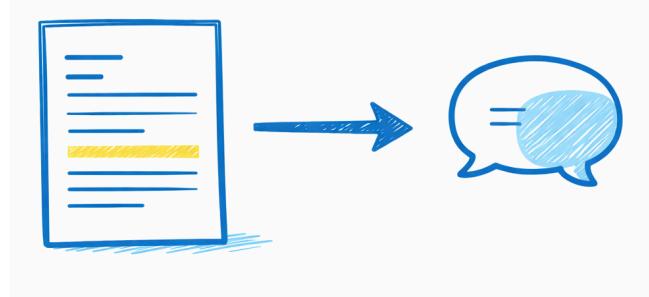
And because everything you write can turn into a question, a resume can either [work for you or against you](#).

If DSA isn't your strongest area and you'd rather not have the entire interview revolve around it, your resume can help steer the conversation toward your projects. You can't fully avoid DSA questions, but strong projects or meaningful hands-on work can catch the interviewer's interest and shift the discussion toward what you've built instead. This won't always work, but it can sometimes tilt the interview in your favour.

The resume can easily work against you, especially when you add skills and projects only to "sound impressive." Tools you barely touched, skills you can't explain, or vague lines that look good on paper but have no depth tend to backfire.

They lead to uncomfortable pauses, follow-up questions you weren't prepared for, and situations where the interviewer keeps digging because your answers sound unsure. That's when the resume that was meant to help you ends up doing the opposite.

In that sense, a well-written resume doesn't just summarise your profile. It quietly steers the interview in the direction that suits your skills. It gives you a bit of control to set the tone of the interview.



To cut to the chase, how do you even write a good resume?

You've probably heard about ATS scanners or seen a bazillion resume formats on LinkedIn. At this point, you might be wondering which one actually works. For on-campus drives, you don't have to stress about format, NITW already has a **standard format** which you are expected to use for on-campus opportunities.

NITW standard format 1

NITW standard format 2

NITW standard format 3

NITW standard format 4

The [personal information](#) section is pretty straightforward. If this part confuses you, the internship is honestly the least of your worries. Just fill in your personal details and add important links like LinkedIn or GitHub. All of this sits in the header part of your resume.

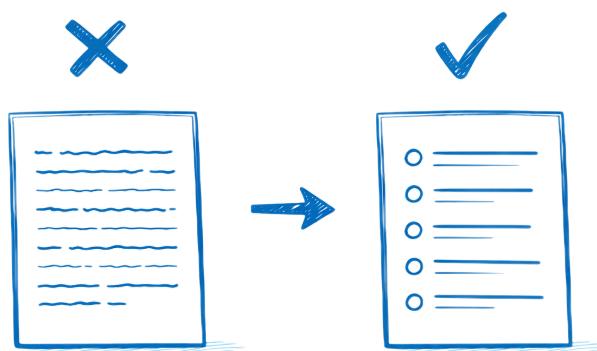
Next comes the [education section](#). Add your 10th, 12th, and NITW percentage or CGPA. Do not write anything incorrectly, and definitely do not round off your percentage or CGPA. Use the exact numbers. Interviewers barely look at this part, and it's there just for formality.

After that comes the [experience section](#). If you don't have any relevant work experience during your intern drive, that's fine, you can remove this section. For those who do have experience, avoid exaggerating your role or stuffing the section with trendy buzzwords. You don't want to sound clueless when interviewers dig deeper. Being unconfident or unclear about something mentioned on your own resume is a major turn-off for interviewers.

Then comes the [projects section](#). Always list your strongest or most recent projects first. Interviewers usually glance at the first project on your list and use that as their starting point. The ordering of your projects matters a lot, more than most people realize. If you put a half-baked project or something you barely remember at the top, that is the project you will likely be questioned on.

On the other hand, if your first project is something you genuinely understand well, you instantly create a strong starting point for the interview. Most interviews discuss only one project, so lead with something you can confidently defend.

Whether it's experience or projects, avoid paragraphs at any cost. Use short and clear [bullet points](#), which highlight the important details. Don't try to cram every single small detail, as your interviewer does not have the time to read through them. Stick to what you actually did and avoid adding fluff to score brownie points with the interviewer. Your wording matters more than you think. If you write that your project was "innovative" or "efficient", the interviewer can (and will) ask what made it innovative or how you measured the efficiency.



You'll see a lot of advice online telling you to "[quantify your impact](#)". Remember, anything you write can turn into a follow-up question. You really don't want to freeze when someone asks how you calculated a 271% improvement that you made up five minutes before submitting your resume.

If you're uncomfortable with a tool or language you used in a project, you can leave it out to avoid deep questions about it. Just don't remove anything essential. No one wants to see an ML project with no Python mentioned.

But if there were libraries you barely used and don't fully remember, it's fine to skip them.

Deciding what to include and exclude in your resume takes time. I suggest you make your initial draft of your resume, take a break, come back to it, and repeat that cycle.

A good resume usually goes through several rounds of small tweaks.

Next is the [technical skills section](#). When I filled this in for the first time, I had a moment of panic, thinking I didn't know anything well enough for in-depth questioning. You should avoid adding skills you're not comfortable with, but relax. Interviewers know you don't have industrial-level experience yet. Add whatever you feel reasonably confident using.

If you add soft skills here, make sure you have a story (real or made up) to back each one. Interviewers love asking for situations where you used a particular skill. If you write "teamwork," they might immediately follow up with, "Tell me about a time you worked in a team and faced a challenge." Soft skills are easy to add, but they're often the ones interviewers dig into the most because it helps them understand how you behave beyond the technical details.

For coursework, don't dump your entire course syllabus on the resume. Keep whatever subjects you've done from the [big four](#) that actually signal competence: DSA, OOPs, CN, OS. Nothing more. Adding extra subjects might feel tempting, but it just gives interviewers more bait to grill you on. A senior once added Design and Analysis of Algorithms to their resume and ended up deriving recursion time complexities on the spot. No need to volunteer for extra suffering.

If you have a strong CP profile with high ranks, you can add a separate CP section to highlight it.

Now for the Positions of Responsibility (PoRs) section, add any clubs you're part of or similar roles.

Don't include school-level roles unless they were genuinely significant. No one cares about your prefect days from school.

Finally, you reach the last section, the [Achievements section](#). This is your space to brag about yourself. Add any awards (doesn't necessarily have to be academic), hackathons, certificates or scholarships you received during college. Follow the same rule as PoRs and keep school-level achievements out of this unless they are important. Some people include their JEE ranks. You can do that too, but it does not add much value because they already know you study in NITW. Don't make it look like that's the only achievement you have.

For certain specialist roles like data science, cybersecurity, etc., if you have any relevant experiences or skills, do ensure to tailor your resume accordingly and highlight those skills appropriately.

The important thing is to make your resume your own. Don't write things just because everyone else is doing it. This document only needs to reflect your work, your skills, and your comfort zone!

CHAPTER

Interviews

The Art of Selling Yourself

So you go through the nightmare that is OT, wait for the shortlist to drop, and you're anxious for the results the entire time after that. And then, when the shortlist is out, your name is actually there! You made it! The only thing standing between you and that internship is the interviews. That's when the panic hits, the excitement is there, but suddenly your brain starts going "Wait a minute, what do I even SAY?"

Here's the first thing you need to accept: there's no standard format for technical interviews. Every company does its own thing, and sometimes different panels in the same company conduct interviews differently. Some wrap everything up in one chill round, mostly chatting about your projects and maybe some HR questions. Others put you through 3–4 rounds of pure DSA hell, with graphs, DP, and everything in between. Companies change their interview style often, sometimes on a whim. So even your seniors' experiences might not match yours.

The only mindset that works? Be ready for anything. I know that sounds stressful, but honestly, it's kind of freeing once you accept it.



During your interview, the first thing you will face, unsurprisingly, is DSA. When the interviewer gives you the problem, don't rush like you are in a speed coding contest. Take a second, and write down the points the interviewer mentions while explaining the question.

In offline interviews you will be given paper, and in online interviews you can just use your laptop notepad.

Once you are clear on the question, ask the right things before starting. Input ranges, edge cases, whether duplicates are allowed, sorted or unsorted input, maximum input size, anything that might change your approach. Many candidates get stuck later because they did not clarify these things at the start.

And then comes the part many people get wrong. Do not sit silently and code in your head. It is fine to ask for a minute if you need to brainstorm, but after that, **TALK!** Say everything you are considering, even if you know it is wrong. You can say something like, “I thought of doing it using this approach, but I realised it will not work because it does not cover this edge case.” This shows the interviewer that you know how to debug your own idea.

What they really want to see is how you break down a problem and how you move from confusion to clarity. When you explain your **thought process**, the interviewer also understands where you are stuck and might guide you with small hints in the right direction. Silence makes them assume you are stuck, and it makes them lose interest very quickly. Because in the end, they are not only looking for someone who can code well, but also someone who knows how to communicate effectively as well.

Start with the brute force solution, walk through the logic, analyse the time and space complexity, and then move to a more optimized version. Jumping straight to the most optimized solution only because you practiced the same problem on LeetCode looks like you’ve memorized a bunch of solutions.

Remember, companies want someone who can build solutions.

The next type of questions you might face are from CS fundamentals like OOPs, DBMS and sometimes OS. Here, demonstrating you know the concept and its practical uses is much more important than giving textbook definitions. What they want to see is whether you actually understand how these concepts are used in real development. So if they ask something like “What is a transaction” or “What is inheritance”, don’t memorize a line from the internet. Explain it in your own words, and then give a short example from real life or from your project.

For example, if the interviewer asks, “What is atomicity in DBMS?”, a good answer would be:

Atomicity means that a transaction is treated as a single, indivisible unit. All the operations inside it must complete successfully, or none of them should take effect. There is no partial success. For example, transferring money between two accounts. The debit and credit operations are part of a single transaction. If anything goes wrong after the debit, atomicity ensures that the system rolls back the entire transaction so the balances remain accurate.

For DBMS specifically, they might go beyond theory and ask you to design a real-life system. For example, they could ask you to create tables for a Hotel Management System, identify primary and foreign keys, or draw an ER diagram and explain the relationships. If they ask a SQL query, first explain what you are planning to do, check your assumptions, and only then write it. Interviewers are not impressed by five line queries, but by answers that show clarity and good reasoning.

Always keep your answers simple, confident and practical. You don't need heavy jargon or complicated sentences. Explain the concept in your own words, add a small relatable example and connect it to how it helps while building software. When you answer like that, the interviewer sees clarity which is exactly what they look for.

For students who aren't from a CS-related branch, seeing OS in the list might immediately trigger panic. But if OS isn't part of your course, you don't need to stress about it. It's a core CS subject, and interviewers know that non-CS students wouldn't have learnt it. Just focus on DBMS and OOPs. If you know these two well, that itself is a huge plus for someone outside CS. If OS does come up in your interview, you can simply say it isn't part of your curriculum.

Also, during your interview, if you get any language or tool specific question which you haven't used, just be honest. You can always mention that you haven't worked with those languages/tools yet.

Suppose they ask you to implement some OOPs related concept in JAVA and you learnt it in Python, you can tell them that and do it in your preferred language.

After going through a few technical interview rounds, you don't have the energy to explain even one more DSA question. Then you get a message from your SPOC saying you've been selected for the next round, and it's an [HR round](#). That is when the existential crisis creeps in.

At this stage, you've shown you can do the technical work. What's left is proving that you're easy to work with and that you would fit right into the company's culture.

The first thing in every interview, including the HR round, will be an introduction of yourself.

Your introduction should be prepped beforehand and should take only a few minutes. Don't ramble too much. It should cover your branch, your college (if off campus interview), your technical interests, a brief description of the projects you've done, and any extracurricular activities or positions you hold.



Remember, your interviewer will have a copy of your resume, so make sure your introduction isn't just your resume word for word. They don't need the audiobook version. Use this time to add context and expand on whatever you've mentioned in your resume. Your introduction should be succinct and well rehearsed. Don't try to wing it because your nerves will surely betray you.

Okay, so you've introduced yourself. Now what? Unfortunately, only the introduction is a standard part of every HR round. After this, your interview can go in any direction. But don't worry, this round is not as daunting as it seems. It is all about showcasing who you are as a person, your personality, your attitude, and your soft skills. Keep this in mind when answering questions.

So let's first talk about personal questions. These are the stereotypical questions that probably popped into your head the moment HR was mentioned. Questions like "*Introduce yourself*", "*What are your strengths and weaknesses?*", and so on.

You may also be asked about your hobbies and interests. All of these questions seem easy. After all, they're about ourselves, right? Wrong! Don't underestimate them. You don't want to be the person who gets nervous in the interview and suddenly blanks out and can't describe themselves.

Unlike your introduction, you should not have these answers mugged up as it will come across as disingenuous if the interviewer can tell it is rehearsed. Instead, remember the points you want to cover.

A very important question you'll be asked is why you want to work in that particular company. Make sure you've done adequate research about the company beforehand. Your answer should be specific to the company. A generic answer makes you look disinterested. "I wanna be employed" is not the vibe you want to go for.

Keep this in mind. The company is not hiring you for two months. They are hiring someone they might want to work with long term. Even internships are extended interviews for full time roles. So show genuine interest in growing with the company during your interview, regardless of what your personal plans may look like down the road (like switching companies or going for higher education).

The next type of questions are [situation based](#) and [behavioural questions](#). The interviewer may give you a hypothetical scenario like "What would you do if your teammate wasn't pulling their weight?" or they may ask you to describe a situation from your life like "Tell me about a time you faced a conflict in a team."

For such questions, use the STAR method to create a well-structured narrative. STAR stands for Situation, Task, Action, and Result. First describe the situation and what your task or role was. Then describe in detail what actions you took to fix the problem. Lastly, explain how those actions resolved the conflict or helped achieve the desired outcome. Basically, make yourself the protagonist of the story.

Sometimes, you may be asked [puzzles or brain-teasers](#). Even if you don't get the answer immediately, don't panic! Just like DSA, these are meant to assess how you approach a new problem. Explain your thought process and how you reached certain conclusions.

Be thorough with your resume because many interviewers will ask questions directly from it. Nothing signals a red flag faster than hesitation and confusion about something you chose to keep on your resume.

If they ask you to explain your projects, create a strong narrative starting with what problem you wanted to solve, then what you tried, how you came up with your solution, and tie it to a real world application if possible. If you feel like your DSA is weak, you can gently guide your interviewer into going more in depth into your projects.

At the end of every interview, including technical ones, the interviewer will ask you if you have any questions. Do not say "No, I think I'm good." This is a great opportunity to show your interest in the role.

Prepare a few thoughtful questions beforehand and make them company-specific.

You can say something like, “I would love to know more about your role in the company and the kind of work you’re involved in.” Once they explain what they do, you can ask follow up questions based on their answer.

But use common sense. If the interviewer looks like they are running out of time or clearly trying to wrap up, do not suddenly start a five minute Q&A session. Read the room.

You might have heard seniors say that one company loves DP and graphs while another leans heavily into specific CS fundamentals. Before you spiral into preparing for every possible question that could ever exist, CSES has [Internito](#).

Internito is a platform filled with detailed OT and interview experiences shared by students who were shortlisted or selected by different companies. It gives you a clear picture of company-specific interview structure, the type of DSA problems asked, and the depth of CS fundamentals they focus on. This insight can help you make your preparation more focused, especially right before a company’s OT or interview.

And once you get selected, you might find yourself logging in again, this time to add your own experience as the next story on Internito.

CHAPTER

Applying Off-campus

You're On Your Own Now

At one point during the intern drive, you notice the number of companies visiting campus slowly dropping. The excitement in the CCPD group fades. Fewer forms show up. Shortlists become rarer. And suddenly, it hits you.

“What if I don’t end up getting an internship through campus?”

If you’re in this position, first of all, breathe. You’re not alone, and this situation is far more common than it feels when you’re living through it.

This is where off-campus applications come in.

Off-campus opportunities are not a sign that you’ve failed or missed your chance. They simply exist outside the campus system and operate differently. There are no fixed schedules, no common shortlist dates, and no shared sense of progress. It’s just you, your resume, and companies hiring on their own timelines.

Because of this, there are a few things you need to get right from the very beginning.

I. Your Resume Matters!

The first mistake many people make is using their on-campus resume everywhere.

Don’t.

Always create a separate off-campus resume using a standard, ATS-friendly template. Templates like [Matty’s resume](#) are widely used for a reason. They are simple, clean, and optimized for automated screening systems. These tend to perform much better in ATS scans compared to the NITW format.

Think of it this way.

On campus, a human is very likely to see your resume.
Off campus, a machine almost always sees it first.

So optimize accordingly.

II. Apply and Forget (and Be Patient)

This is the most important rule of off-campus applications.

Once you apply, MOVE ON!

Off-campus hiring takes time. Do not sit refreshing the careers page and your inbox every ten minutes. Do not assume silence means rejection or acceptance. If a company wants you, they will mail you. Until then, it is completely out of your hands.

Some companies respond quickly. Others take weeks or even months. Long gaps, delayed replies, and extended processes are all normal. Obsessing over timelines only drains your energy, and patience is not optional here, it is part of the process.

III. Referrals Are Not Cheating

Getting a referral is not unethical. This is the time where you should absolutely use every unfair advantage you have.



A referral doesn't guarantee selection, but it significantly increases the chances of your application being seen by an actual recruiter instead of getting lost in a pile of thousands. And that alone is a massive advantage.

Do not feel awkward about reaching out to seniors, alumni, or connections on LinkedIn. Be polite, be concise, and be respectful of their time. Most people remember how confusing this phase was for them and are genuinely willing to help.

The worst thing that happens? They don't reply. Which changes absolutely nothing.

IV. Your Online Presence Matters

At this stage, LinkedIn becomes extremely important. Follow people working in the field you are interested in, seniors, alumni, recruiters, and HRs from companies you want to apply to. Many off-campus openings are shared directly through LinkedIn posts, reposts, or even comment sections long before they appear on job portals. Staying active here increases what you get to see.

Alongside this, make sure your GitHub profile reflects you well. Keep it organized, and updated. Pin your best repositories, write clear README files, and avoid clutter. Even if a company does not explicitly ask for GitHub, recruiters often check it when evaluating off-campus candidates.

V. Be Careful While Applying

One last thing while applying off campus: be cautious!

If a company asks you to pay a registration fee or any kind of upfront amount, walk away immediately. If a role description is vague, promises unrealistic pay, or asks for excessive personal details early on, treat it as a red flag.



Legitimate companies do not charge you to interview.

Off-campus opportunities reward consistency and patience more than anything else. If you keep applying, keep improving, and keep your head clear, something will eventually click.

And when it does, it will feel completely worth the chaos.

CHAPTER

Advice You Probably Don't Wanna Hear

-specially from our seniors

So you've read through the various sections and are now questioning if you've got it in you to survive the intern drive stress. Maybe you're feeling overwhelmed or that you're late to start preparing. And yes, I'll be honest, the stress does increase ten-fold once the intern drive actually starts. It usually gets a bit worse before it gets better, and then one fine day you bag that internship offer and suddenly life feels better!

If you're reading this with a lot of time still left before the intern drive, this is a great time to start your preparation. Intern prep is much easier when you do it **slowly and consistently**. Starting early, ideally during second year, and putting in small consistent efforts saves you from turning your summer into juggling DSA and projects and falling short of time. You do NOT need to study like you did for JEE. A few solid, focused hours every week is more than enough.

To survive this phase, it's really important to rant and talk about your frustrations with your friends and family. Avoid getting hypercompetitive with your friends, rather lean on each other for support instead. Also, let's be real, DSA is not fun for most of us. Doing it with friends makes it less painful, keeps everyone accountable, and lowers the chances of burning out halfway.

During the intern drive, a lot of factors that affect shortlists are completely out of your control. **LUCK** plays a bigger role than anyone likes to admit. You might prepare everything you can possibly do and still get asked something obscure that trips you up, while someone else gets an interview panel which asks questions you could've easily aced.

You'll spiral thinking, “*what if I had their interviewer?*”, “*what if I had revised this topic before?*”, “*What if I had phrased my answer better?*”. Unfortunately, that's just how this process works, sometimes luck simply isn't on your side.

Sometimes your interview goes amazingly well and you still get rejected. Sometimes you walk out convinced you messed everything up and somehow still get selected. Your ML prediction skills are absolutely useless here. Do not assume anything before the final list is out. Rejections will happen. It sucks. Rant for an evening and then move on. There WILL be more chances.

Let's also talk about the idea of a dream company. Having goals and ambition is great, but the intern drive does not operate on logic. Avoid keeping all of your hopes on getting selected for your “dream” company, and try your best for every opportunity that comes your way. Once the intern drive starts, you'll quickly see how much randomness and luck are involved. Do not take every rejection personally. Sometimes it just wasn't your turn yet.

The drive is chaotic and exhausting, fortunately it's temporary. You have to wait for the day you perform well and luck favours you.

CHAPTER

You Got In!

Now Try Not to Blow It Up

CCPD Notification.

Final Intern Selects.

You.

Damnnn! You got the internship. Congratulations! After all that turmoil and turbulence, you've finally done it. You've reached the phase where the air smells fresher, water feels more refreshing, and life is all sunshine and rainbows. You can click that 'X' on your DSA sheet tabs and pretend like you'll never have to open them again!

Now begins the waiting period for the [offer letter](#). This varies from company to company. For some, it arrives a few weeks after selection, while for others, it comes just a few weeks before the internship starts. CHILL OUT, IT WILL COME!

For a lucky few, there will be an option to choose their preferred location - Bangalore, Hyderabad, Pune, or Gurgaon. These are the usual suspects, and it's advised to pick the location that suits your comfort. We'd recommend choosing a city you're familiar with, or the company's main branch by default if you're indifferent to all of them.



Once you have that shiny paper in your mail, there are a few things you can do. Based on your assigned location, start looking for suitable accommodation. Choose a PG after consulting a few seniors who have previously worked there for the best options. This is optional, but if your company has already assigned you a team or manager, feel free to look them up on LinkedIn to get an idea of the kind of work you might be doing. Don't feel the need to learn anything yet - it's more of a curiosity check than anything else.

While extremely rare, there have been cases of last-minute changes in location or team assignments. Don't panic if this happens to you. This shouldn't change how you approach your work during the internship. Just be prepared for such scenarios as well.

Few Months Later

We've finally reached Day 1 of your internship. You see the company logo on the building, smell the constantly brewing coffee, and hear the click-clack of mechanical keyboards. While your first corporate badge is being printed, take a moment for yourself and think about how far you've come!

Most companies, especially those in the finance domain, usually have a dress code. Be sure to check it and follow it. On the other hand, regardless of what companies say about being "casual" or "chill," it's imperative that you look presentable. Don't forget to cut your hair, trim your nails, and dress to impress. **First impressions do matter**, so make them count!

You've probably met your mentor and your manager by now. How you interact with them, along with your team, will shape your output and overall experience as an intern. It's crucial to set clear objectives and expectations for yourself as early as possible. [Ask them. Don't assume!](#)

This will help you along the way to justify your work or even highlight how you've exceeded expectations. It's advised that you set up some time every week to interact with your manager. Discuss what you worked on during the past week and what you plan to do in the coming one. This shows that you're proactive and, frankly, helps you GET NOTICED, which is paramount in corporate life.



Initially, you might feel like you don't belong, especially when you start hearing about tools and technologies you're encountering for the very first time. Don't panic! Remember that your team knows it's your first time here. [Ask questions!](#) Try to solve problems on your own at first, but don't hesitate to ask for help if you get stuck.

Right then, now you're locked in. You know the tasks ahead, and you're working. While it's great to finish your work, always be open to criticism. Cultivate a positive, growth-oriented mindset, it's what will define you as a true team player. If you believe you have an idea or a better way of doing something, share your thought process, but be ready to be challenged.

There's usually no absolute right or wrong answer in these companies. It's your **thinking** that's being evaluated. This is also where I'd like to remind you, enjoy your time and don't be a robot! At the end of your internship, your gallery won't be filled with screenshots of the work you did, but with the memories you made along the way. BE SURE TO TAKE PHOTOS!

Be open to those coffee chats with people on your team. While it may be tempting, keep the conversations light-hearted, i.e. don't start bragging about your work or comment on how others are slacking. DO NOT GOSSIP.

Participate in company events like team lunches and outings with genuine enthusiasm. Saying "yes" should be a no-brainer! These moments are perfect opportunities to show your team that you're more than just your technical skills and just a chill, easygoing person. PS: If you're invited to a house party or a place where fancy drinks are involved, it's perfectly fine to say no. You won't be judged for it.

During your internship, **time management** is crucial. Set daily, weekly, and overall summer goals. Keep a to-do list for your internship and make the most of your chosen location. Time will fly. Own your time!

Always remember that an internship is a two-way street. While the company is monitoring your performance to determine your future, it's equally important for you to form a clear opinion about your own future with the company.

Due to the rules of our CCPD, which more or less state that *if you receive a PPO from a company, you cannot sit for placements*, it's important to make your decision carefully.

If you feel that you don't see a future with the company for any reason, it's advisable to inform the HR and your team after the end of your internship. However, this is a bold choice as you'll then be relying on the potential of the companies coming for full-time recruitment and your ability to crack them.

In case you're planning for higher studies, you can choose to stay silent. You'll still have the flexibility to accept or decline the PPO later, keeping it as a backup in case your plans for a masters program don't work out.

Exit interview

An exit interview usually takes place in the final days of your internship. The exact format varies from company to company, and sometimes even from team to team within the same organization. Some exit interviews are very casual, while others are more structured. In certain teams, it may even feel like a proper technical round. There is no single standard format.

Depending on the company or the team, the interview could range from simple questions like, "How was your experience working with the team?" to more technical prompts such as, "Given this scenario, design a low-level system for it." So make sure you revise your project, core concepts used in your project, design principles used and basic DSA and CS fundamentals.

Because of the variation in format, it's important NOT to assume the format and to prepare accordingly. Ask your managers, mentors and team members for what you can expect in the interview. It is your final impression before decisions are made.

Skip to the last day of your internship

Looking back, those two months will have felt like a breeze. As you prepare to leave and turn in your badge and laptop, make sure to have those final conversations with as many people as you can, and express your gratitude to everyone who helped you along the way. Ask for their socials (professionally) so you can stay connected in the future. You can also write an acknowledgment post about your experience on LinkedIn. *Don't forget to attach those team photos I told you to take!*



Naturally, after everything you've been through, there might be a quiet thought at the back of your mind: "*Do I really want to go through the entire process again for placements?*". The very idea of repeating it all can feel daunting.

But here's the good news, if you're confident that you want to stay with the company, there are three magical letters that can make sure you never have to walk through the halls of CCPD again, **PPO!**

CHAPTER

Post Intern

You Don't Get To Relax Yet

PPO

Now obviously the first question you might have is, “What even is a PPO?”

A Pre-Placement Offer (PPO), as the name suggests, is a full-time job offer that a company gives you based on your performance during your internship. Instead of going through the entire process again in your final year, your company directly offers you a role essentially rewarding the hard work and impact you showed during the internship.

Now, a PPO is offered only when your managers and teammates believe you can consistently meet their standards and be a valuable addition to the team or the company as a whole. On the surface, it may seem like a simple yes-or-no question: “*Will you be offered a PPO?*” In reality, several factors that influence this decision are within your control.

Start by understanding what your team does, and understand how it fits into the bigger picture of the company, what they are building, the tools they use, and the kinds of problems they solve. Then, understand your own project: what it is for and how it connects to your team’s work. When you can clearly see how your contribution fits in, things begin to make sense, and you will feel more confident to tackle the problem you are helping to solve.



In your first week, make sure to meet with your manager or mentor to understand what your tasks are and what they expect from you. It's important to know exactly what you're responsible for and what you should finish by the end of your internship. You can always try to do more than what's expected, but first make sure you're clear on your goals and deliverables.

As you work through your internship, focus on completing as much of your project as you can and make sure all requirements are met. Check whether your project actually achieves its purpose.

For example, if you were asked to build a tool to automate a task, does it truly make the process faster or easier? If you were supposed to improve an existing feature, does your update solve the problem it was meant to fix? Regularly asking yourself these questions helps you stay on track and ensures that the work delivered is meaningful and aligns with what your team's needs.

Besides the technical aspects of your project, your behaviour also plays an instrumental role. How you interact with your manager, how you behave with your teammates, and even how you treat your fellow interns, all of these contribute significantly to the impression you leave and the confidence they have in offering you a PPO.

Your behaviour with your manager should always be polite and respectful. Do not treat your manager like a professor. Instead, see them as a guide throughout your internship. Make it a habit to check in regularly, either weekly or biweekly, to share your progress, explain your thought process, and ask for feedback.

It is important to remember that managers have many responsibilities, and it is best to give them space and avoid adding to their workload. Your manager probably deals with numerous issues, and they certainly do not want their intern to become another one. Always try to reduce their burden by avoiding unnecessary questions or constant messages, especially when your teammates or co interns can help you. At the same time, ensure that your manager is kept informed about anything important. The key is to find the right balance between not disappearing entirely and not being overly demanding.

While interacting with your colleagues, be approachable, respectful, and easy to work with. Treat everyone with the same level of courtesy regardless of their role or experience, and make an effort to communicate clearly, ask doubts, and show gratitude when someone helps you.

Observe how your team operates, adapt to their work style, and maintain a positive, professional attitude. Make sure you remain confident without becoming cocky or trying too hard to impress people. Participate in team activities whenever possible to bond with your coworkers beyond just work. For informal interactions, understand what kind of jokes are acceptable in your team and what is not, learn to read the room, and if someone makes a harmless joke about you, take it sportingly rather than personally.



Your colleagues can become valuable sources of guidance, learning, and support, so building good relationships with them not only makes the workplace more comfortable but also reflects well on your overall performance during the internship.

It also goes without saying that you should never start begging for a PPO or bring it up from the very first day. Talking about PPOs too early can make you seem impatient and uninterested in actually learning. Instead, focus on being a genuine member of the team, settling into the workflow, and contributing naturally. Ironically, being “nonchalant” about your PPO is key. Even if your manager or colleagues bring up PPO, acknowledge it, but keep the response subtle and indirect rather than overly eager. Remember that the primary purpose of your internship is to learn, grow, and understand how real-world projects function. A PPO should be a byproduct of your performance, not your sole mission.

Most importantly, you’ll have a final presentation toward the end of your internship where you showcase your work to your teammates, your managers, and possibly members of sister teams. It really helps to keep this presentation structured and easy to follow. Walk the attendees through the problem you tackled, the approach you took, and the impact your work actually created. Make sure to thank your teammates and share the positive things you learned from them, without dwelling on anything negative.

In many cases, your manager’s managers or other senior leaders will also be in attendance, and they often end up assessing your manager’s guidance just as much as your own work.

Because of this, it is always a good idea to acknowledge the support, clarity, and direction your manager gave you. Do highlight how your project fits into the larger picture and why it matters, because people seeing your work for the first time, especially upper management, usually care far more about the value it adds to their team than about the specific technologies or implementation details. Lastly, present confidently and without fear. If you ever start feeling nervous, just remind yourself: “*This is my work and story, and no one in the room knows it better than I do.*”

Delayed PPO

Let’s address the elephant in the room, PPO delays. Sometimes they take weeks or sometimes months. But delays are completely normal in some companies because of their business planning and recruitment strategy.

At this stage, you are usually faced with two paths, each with its own uncertainties and outcomes.



Option 1: Wait patiently for PPO

You decide to wait for your PPO result, accepting that this may affect your on-campus placement chances.

This path requires unshakeable optimism and the ability to answer “*Any update?*” from your friends and family constantly. This may look like a risky scenario, but the risk level depends on multiple factors like how your work went, what your manager and mentors have indicated, the company’s current hiring plans, and even their past PPO conversion trends. You should absolutely consult your manager and mentors, as their perspective often gives you the clearest signal. Their feedback will be a major part of the evaluation behind your PPO result.

If your heart is truly set on the company you interned at, and you feel it is worth the wait, then waiting is a perfectly valid decision.

- Best case: You get the PPO and experience pure relief.
- Worst case: You get rejected, but at least you’ll know you gave yourself every chance. No “what ifs,” no regrets.

Option 2: Choose On campus and drop PPO
On the other hand, maybe you’re not emotionally married to your internship company. Maybe you just want to get placed in 7th semester, lock in a good offer, and enjoy your final year. Totally fair.

In that case, skip the PPO suspense, focus on your DSA, projects, mock interviews, and aim for the on-campus opportunities. This route does not allow you to accept your PPO later (unless there are very exceptional circumstances). Usually the first offer is the final offer. However, this option gives you clarity and an early job offer.

So, which is right?

The truth is that there is no universal “correct” choice. There’s only what aligns with your goals, your risk appetite and your mental peace. Whether you wait or you don’t, what matters is that you own the decision.

FTE

Internship preparation already takes you halfway through FTE preparation. The overall structure of the placement drive is largely the same as the intern drive. With that in mind, here are the key points:

1. Practice DSA to the point where you dream about linked lists.
2. Gain a good understanding of OOPs and DBMS.
3. Have an awesome and unique project of your choice (make sure it's unique).

These points do not cover everything, so go through the internship preparation section [\[Chapter 3\]](#) if you haven’t already.

Let's say you've finished your internship and are lying on your bed, wondering, "What now?" Lucky for you, I have outlined the next steps below.

Update your resume

Your resume is an active indicator of your current skills and experience. Never forget to update it. You gained significant experience during your internship and contributed to a solid project, and there is no reason to leave that out. As mentioned before, one of the first things an interviewer will likely do after saying “Hello!” is read your resume.

That is where they realize that the candidate in front of them has done an internship and worked on an interesting project, which naturally leads to follow up questions about it. To be precise, under the Experience section of your resume, include a brief summary of your project, along with the company you worked at and the start and end dates of the internship.

Revise DSA

Unfortunately, you are not done with DSA. This topic will continue to haunt you. DSA is still one of the most asked questions in an FTE interview. So revise important algorithms, practice data structures, and learn how to solve those hard questions you left.

Projects

It's probably been some time since you last worked on the project you showcased during your internship interview. Your interviewer will still expect you to remember it as if you wrote the last line of code two hours ago. Similar to internship preparation, you should understand your project thoroughly and be ready to answer questions about it, not just the tech stack, but the underlying theory and concepts as well.

Interviewers may also ask about your internship project, so be prepared for that too. You do not need to explain every detail, but you should be able to clearly talk through the important points.



Take some time to research the company you are applying to and align your preparation and resume with what they value. For instance, if you are applying to a cybersecurity-focused company, it's smart to revise cryptography and related concepts. If you have built a project in that area, place it higher on your resume to highlight it and make sure to discuss it during interviews. This can significantly strengthen your chances.

Higher Education

When should you consider higher education?

Most people consider higher education in the following cases:

- Switching to a different career path
- Aiming for management (MBA) roles
- To specialize in a certain area of interest and gain deeper knowledge to unlock better opportunities.

When considering higher studies, there are a few things to keep in mind and ensure that higher education is indeed, the path you want to take forward.

- Does the course help you attract work opportunities you would like?
- Are you ready for the rigorous work load that comes with certain courses, specifically, thesis-based ones?
- Is the time and money, i.e. the opportunity cost, it takes to pursue a Masters/PhD/MBA course going to result in an outcome that you will be satisfied with?

Think through each of these questions thoroughly, and do your research.

If your answers to these questions point you towards pursuing higher education, here are a few things you need to prepare for:

- Study for GATE, CAT, GRE (for universities abroad), etc. exams to be eligible for applying to the programs you desire.
- Maintain a good CGPA that demonstrates your knowledge, consistency and hard work.
- Develop projects, participate in research, and do internships that showcase your experience. Having a paper or two under your belt definitely helps, especially when applying to programs abroad.
- Maintain good relations with a few of your favourite professors, who can give you glowing Letters of Recommendations.
- Do thorough research about the universities and courses you're applying to so that it shines through your Statements of Purpose.



And That's a Wrap (Sort Of)

So, you've made it to the end of this ebook.

By now, you hopefully have a clearer picture of what the intern drive actually looks like. You know what to prepare, what to expect, and what life might look like once you've finally landed that offer.

Here's the thing though. No ebook, guide, or senior's advice can fully prepare you for the very specific, very personal rollercoaster that your intern drive will be. There will be moments that feel unfair. There will be shortlists you were sure about that don't come through, and ones you never expected that do. There will be nights where you genuinely question everything, and mornings where it all feels manageable again.

That's normal. That's the process.

What matters is that you show up, keep going, and don't make the mistake of measuring your worth by a shortlist or a rejection. The drive is one chapter of a much longer story, and honestly, it's one of the more character-building ones.

So close this ebook, open your laptop, and get to work.

You'll figure it out. Everyone does, eventually.

And when you're on the other side of it, sitting in your PG room with an offer letter in your inbox, come back and pass this along to someone who needs it.

Good luck. You've got this!

CREDITS

CONCEPT AND CORE CONTENT

QCM Team, CSES

GUIDANCE

Prisha Behera

Dharani Murugan

DESIGN

Design Team, CSES

CONTENT SUPPORT

Final Years. CSES

