

A PRACTICAL GUIDE FOR 2027 BATCH FRESHERS



Get Found. Get Called. Get Hired.

How recruiters actually search for candidates, how to rank higher on Naukri and LinkedIn, and how to prepare for the interview that follows.

Sayan Paul

Full stack developer, Kolkata • sayanpaul.in

Written from my own job hunt, November 2026

The short version

I'm a final year BCA student graduating in 2027, and I've been job hunting for developer roles. Like most freshers, I applied to a lot of jobs and heard back from very few. So I stopped guessing and looked at how the other side works: how recruiters search, which keywords they type, what the job portals reward, and what interviewers are really checking.

This guide is everything I learnt and applied to my own profiles. Some of it comes from recruiters explaining their own tools, some from articles I've cited at the end, and some from a small experiment I ran myself on 100 live job posts. Where I'm sure, I say so. Where something is a snapshot that can change, I say that too.

The one idea to remember

Recruiters don't read profiles first. They **search** first. If your profile doesn't contain the exact words they type, they never see it, no matter how good you are. Most of this guide is about fixing that, honestly.

If you're in the 2027 batch like me, this is written for you: the placement and off campus season you're heading into right now.

What's inside

Part	What you will learn
1. How recruiters search	The filters and keywords that decide who shows up
2. Find your keywords	A simple way to see what recruiters in your field search for
3. Naukri	Every section that affects your ranking, and the limits nobody tells you
4. LinkedIn	The four recruiter filters and how to show up in each
5. Resume and consistency	One page, honest numbers, the same story everywhere
6. Applying smartly	Daily routine, where to look, and how to spot scams
7. The interview	How rounds changed in 2026, the 5 stories, likely questions
8. Skills to learn next	What to study, in what order
9. Checklists	Everything above on one page

If this helps you, I'd love to hear what worked. You can find me at sayanpaul.in. Good luck with the hunt.

How recruiters actually search

Before you change anything, understand the machine. Recruiters work with search tools, filters and keyword matching long before a human reads a single line about you.

On LinkedIn

A recruiter explained in a short video how LinkedIn's candidate search works. It gives about seven filters, and four of them matter most:

Filter	What it checks	What it means for you
1. Job title	The titles in your Experience section, not just your headline	If no position on your profile says "Full Stack Developer", you can be missed in that search even if your headline says it
2. Keywords	Words anywhere in your profile, especially About and descriptions	Use the real industry words: tools, frameworks, skills
3. Years of experience	Worked out from the dates on your positions	Keep dates correct and current
4. Skills	Your Skills section	Add every real skill. LinkedIn allows up to 100

One correction to a lot of older advice: LinkedIn **retired its Skill Assessments**, so don't spend time looking for those badges. Endorsements and skills attached to a specific job or project are what's left.

On Naukri

Naukri's recruiter side works the same way: recruiters search by keywords, then filter by things like location, experience, education, notice period and salary. The pattern is the same as LinkedIn: **your words have to match their words**, and your filters (location, availability, salary) have to fall inside theirs.

What usually pushes a profile up

- **Keyword match** with what the recruiter typed.
- **Recent activity**: profiles that were updated or active recently tend to show up higher.
- **Completeness**: empty sections give the search less to match on.
- **Filters you pass**: preferred location, job type, availability and expected salary.

Quick win

If you remember one thing from this part: **job titles and keywords decide whether you are found**. Everything else decides whether you are chosen.

Find the keywords recruiters use

Don't guess keywords. Collect them from real job posts for the role you want, then use the ones you can honestly back up.

The 30 minute method

- Search your target role on Naukri or LinkedIn with experience set to 0 years.
- Open 20 to 100 posts and note the skill tags or key requirements.
- Count how often each keyword appears. The top 15 are your core keywords.
- Mark which ones you can prove with a project or internship. Use only those.

My snapshot: 100 fresher developer posts on Naukri

I ran this on 100 Naukri posts for React, full stack, software developer and Node.js roles with 0 years of experience (November 2026). These were the most common tags:

Keyword	Posts (of 100)	Keyword	Posts (of 100)
JavaScript	36	Python	16
CSS	33	Software development	15
Front end	22	Ajax	14
jQuery	19	Java	12
Git	19	MySQL	11
HTML	18	Debugging	10
React.js	17	PostgreSQL	10
Backend	17	MongoDB	9
Bootstrap	16	Agile	8

Two things surprised me. First, plain words like **front end**, **backend**, **software development** and **debugging** show up a lot, and most freshers never add them because they feel too obvious. Second, older tools like jQuery and Bootstrap still appear often. I didn't add them, because I don't use them and I'd have to defend them in an interview.

The honesty rule

Only add a keyword if you can talk about where you used it for two minutes. A keyword gets you found; a keyword you can't defend gets you rejected.

Naukri: section by section

Naukri rewards complete, active and keyword-rich profiles. Here is every section that matters, with the limits I ran into myself.

Section	What to do	Watch out for
Key skills	Your most searched real skills, plus broad words like Front End, Backend, Software Development	On my Naukri Campus profile the whole list was capped at about 250 characters, so drop rare or duplicate tags
Profile summary	A short paragraph: who you are, your stack, what you build, what you want. Put extra keywords here	Limit is 1,000 characters. Write it like a person, not a keyword list
Projects	Each project with dates, 2 to 3 lines on what it does and the hard parts, skills tagged, live link	Skill tags must exist in Naukri's list; some (like FastAPI) don't, so mention them in the text
Internships and employment	Add internships with real dates	Adding a full time job can mark you as experienced and drop you from fresher searches
Education	Fill every level	Never enter a wrong percentage
Languages	All languages you speak	Matters for roles that deal with customers or regions
Resume	Upload your latest PDF	Keep it consistent with the profile
Career preferences	Job type, locations, availability, up to 3 preferred job roles, expected salary	Preferred roles are set on the jobs or recommendations page, not the profile page

The settings that changed my recommendations

My profile had 0 preferred job roles, so Naukri kept recommending sales and business development jobs. After adding three developer roles, the preferences tab went from 0 to 29 matching jobs immediately.

- **Preferred roles:** pick the exact titles recruiters use, like Junior Full Stack Developer, Software Developer, React Developer.
- **Expected salary:** it takes one number. Use the low end of your realistic range so you stay inside more recruiter budgets.
- **Locations:** more cities means more searches you appear in. Only add places you would actually move to.

Stay active

- Make a small edit every day or two, even one word. Activity helps ranking.
- Apply to a few relevant jobs daily. Your past applications shape what Naukri recommends next.
- Check "Search appearances" and "Recruiter actions" on your homepage to see if changes are working.
- Never pay for "profile boosting" or placement services. A complete, active profile does the job.

LinkedIn: show up in all four filters

LinkedIn search is mostly about titles, keywords, dates and skills. Here is how to cover each one without exaggerating anything.

1. Job title

This was my biggest mistake. My headline said Full Stack Developer, but my only positions were an AI/ML internship and a design internship. So in a title search, I didn't exist.

If you've built real projects on your own, it's honest and common to add them as a position:

Field	Example
Title	Full Stack Developer
Company	Independent Projects
Employment type	Self-employed
Description	What you built, the stack, real users or results, and one hard problem you solved

2. Keywords

- **Headline:** role plus your main stack, for example "Full Stack Developer | React, Next.js, Node.js, Python | Open to roles".
- **About:** keep it short and human, but include one line that lists your real tools.
- **Descriptions:** every position and project description is searchable text. Use the real names of tools.

3. Experience and dates

Add every internship with exact dates and attach proof as media: certificates and experience letters build trust. When you save a new position, LinkedIn may offer to replace your headline and post an announcement, so check both before you click.

4. Skills

- Add every real skill, up to 100. Use LinkedIn's own suggestion for each (pick from the list, don't type free text).
- Choose your top 5 in the About section; those show first.
- Attach skills to the position or project where you used them.
- Remove skills you can't defend. A listed language you never used will bring interview questions you can't answer.

Open to Work

Add up to five job titles that match recruiter searches exactly (for example Full Stack Developer, Software Developer, Software Engineer, Web Developer). Avoid vague titles like just "Developer". Choose whether everyone sees the green frame or only recruiters.

Bonus

Write posts about what you build: the problem, what you made, one thing you learnt. A post about a real bug you fixed does more for your profile than a list of certificates.

Resume and consistency

Your resume, Naukri, LinkedIn and portfolio should tell one story with the same facts. Recruiters notice mismatches.

Resume rules that work for freshers

- **One page.** Summary, skills, experience, projects, education.
- **Projects carry you.** For each: what it does, the stack, and two or three lines on the hard parts.
- **Action plus result.** "Fixed X by doing Y, which led to Z." Use numbers only if you can prove them, like users counted from a database or tests you wrote.
- **ATS friendly.** Simple layout, real text (not images), standard headings, keywords from the job post that you actually have.
- **Links that work:** GitHub, live demos, portfolio, LinkedIn.

The consistency check

I found my own resume said 15 and 59 tests while my notes said 16 and 61, and I listed WebSockets without ever using them. Small things, but an interviewer who spots one mismatch starts doubting everything else. Before applying anywhere:

- Every number matches across resume, Naukri, LinkedIn and portfolio.
- Every skill listed is one you can explain with an example.
- Project dates and descriptions are the same everywhere.

Quick win

Interviewers will ask about anything you write. If a line on your resume makes you nervous, change it or remove it.

Applying smartly

A strong profile still needs a steady routine. Here is what I do and what I avoid.

Where to look

Platform	Best for
Naukri	Volume of fresher and campus roles, walk ins, services companies
LinkedIn	Startups and product companies, recruiter outreach, referrals
Internshala	Paid internships, many remote
Cutshort and Wellfound	Startups where your GitHub matters more than your marks
Company career pages	Big companies' fresher programs and their eligibility rules

Spot fake or bad listings

- Any request to pay for registration, training, a kit or a certificate.
- A stipend that is far above the market for a simple role.
- The same company posting many near identical roles in different languages or stacks.
- Unpaid "internships" that are really unpaid work with no mentoring.

Search titles that bring more matches

Search each separately, because each brings different results: Full Stack Developer, React Developer, Frontend Developer, Software Developer, Associate Software Engineer, Python Backend Developer, AI Engineer Intern, Technical Support Engineer.

A simple daily routine

When	What
Morning	One focused DSA or skill session
Evening	Apply to 3 to 5 relevant jobs; small Naukri edit
Weekly	One LinkedIn post about something you built or learnt; message 2 people in companies you like
After applying	Note the role, date and resume version. Follow up once after about a week if you have a contact

The interview

Getting the call is half the work. Interviews in 2026 test how you think, explain and check your own work, not just whether you can write code.

What changed in 2026

- **Debug this code rounds:** you get broken code and must find the bug.
- **Explain as you go:** why this data structure, what is the complexity.
- **AI rules vary:** some startups allow AI tools and judge how you check the output; most Indian product companies and IT services still ban them in coding rounds.
- **Take homes come with a walkthrough** where you defend every decision.
- **Behavioural rounds are more structured** and want specific stories you clearly owned.

Rounds by company type

Company type	Typical rounds	What decides it
Early stage startup	Founder chat, take home or live build, culture call	Can you ship and own features
Product company	Online test, DSA rounds, project deep dive, basic design, HR	Problem solving plus project depth
AI first startup	Project talk, practical build, often AI allowed	Building with LLMs and checking their output
IT services	Aptitude, coding without AI, technical, HR	Cutoffs, OOP and DBMS basics, communication
Global capability centres	Online test, DSA, technical rounds	DSA and CS fundamentals

The 5 stories that answer most questions

Prepare five real stories and you can answer almost any behavioural question. Learn the facts and the lesson, then tell it naturally.

Story	Template
Leadership	During ____, I faced ____, so I took ownership by _____. I ____, which resulted in _____. This taught me _____.
Teamwork	While working with ____ on ____, we faced _____. My role was ____, so I _____. As a result, _____. I learned _____.
Success	One achievement I'm proud of is _____. My goal was _____. I achieved it by _____. The result was _____. This taught me _____.
Failure	A situation that didn't go as planned was _____. I made the mistake of _____. I took responsibility by _____. Eventually, _____. Since then I've changed _____.
Struggle	One challenging situation was _____. It was difficult because _____. I handled it by _____. Eventually, _____. This taught me _____.

A failure story that works

"When I first launched my reel summarizer app, I focused on the AI and didn't check who could access what. Auditing my own API, I found anyone could open another user's data by changing a number in the link. I fixed it with ownership checks and signed webhooks, wrote tests that try exactly that attack, and now I think about access from day one."

Likely questions and what they test

Question	What they are really checking
Tell me about yourself	A clear 60 second story ending on why this role
Walk me through your best project	Depth, and whether you can explain it simply
Did you build this yourself or did AI?	Honesty and ownership; walk through one real decision
What would break first at 100 times the users?	Scale thinking and honest weak spots
Here is broken code, find the bug	Calm, careful reading; test edge cases first
What happens when you type a URL?	Web and networking basics
SQL vs NoSQL; what is an index?	Database judgement
Your marks are low, why trust you?	Maturity; let your work answer
Rate yourself in React out of 10	Calibrated confidence: say 6 or 7 and explain why

How to answer anything

- **Shape:** a direct answer, one real example, then how it helps them.
- **Coding:** say the brute force first with its Big O, improve it, then test by hand on a tiny example.
- **Don't know?** Say what you would check and how, instead of guessing.

Questions to ask them: amateur vs pro

Instead of	Ask
What is the company culture?	What do the people who thrive on this team have in common?
What are the company's goals?	Where is the team's focus for the next 12 months, and how would my role feed into it?
How will I progress in this role?	What would I need to prove in one year for you to call this hire a success?

Before and after

- Research the company: product, recent news, tech stack.
- Re read your own resume; be ready to defend every word.
- Check that every live demo loads and every repo is public and tidy.
- Camera at eye level, light in front, join 5 minutes early.
- Send a three line thank you note within 24 hours, and write down every question you were asked.

Skills to learn next

If your stack is already full stack JavaScript and Python, these are the gaps that matter most for fresher roles right now.

Priority	Skill	Why
1	DSA: arrays, strings, hashing, two pointers, sliding window	Most companies run a coding round first
2	Core CS: OOP, DBMS, OS, networking	Asked in campus and services interviews
3	TypeScript	Expected in most React and Node posts; quick win if you know JavaScript
4	AWS basics: EC2, S3, Lambda	One of the most searched keywords
5	RAG and vector databases	The most wanted AI skill right now
6	Docker with docker-compose, Redis	Common in backend interviews
7	Testing (Jest or Vitest), JWT and OAuth, Linux basics	Fill the usual gaps in fresher profiles

DSA: the five topics that matter most

- **Arrays and hashing:** Two Sum, Contains Duplicate, Valid Anagram, Best Time to Buy and Sell Stock.
- **Strings:** reversal, palindromes, character counts.
- **Two pointers and sliding window:** turn $O(n^2)$ answers into $O(n)$.
- **Big O:** asked about every answer you give.
- **Stack and linked list:** Valid Parentheses, Reverse Linked List.

For service company tests, also practise the classics without built in functions: reverse a string, palindrome, prime check, Fibonacci, factorial, second largest in an array, sorting without `sort()`, and pattern printing.

Checklists

Everything above, condensed. Tick these off once and revisit them every month.

Naukri

- ☐ Key skills: top real keywords, including Front End, Backend, Software Development
- ☐ Profile summary with your stack and extra keywords
- ☐ Every project with dates, description, skills and link
- ☐ Internships with real dates; no fake full time job
- ☐ Up to 3 preferred job roles, locations and expected salary set
- ☐ Latest resume uploaded; languages added
- ☐ Small edit every day or two

LinkedIn

- ☐ A position whose title matches the role you want
- ☐ Keyword line in About; real tool names in descriptions
- ☐ All internships with dates and proof attached
- ☐ Every real skill added, top 5 chosen, fake ones removed
- ☐ Open to Work with up to 5 exact job titles
- ☐ One post a week about what you built

Resume and consistency

- ☐ One page, projects first if you are a fresher
- ☐ Numbers only where you can prove them
- ☐ Same facts on resume, Naukri, LinkedIn and portfolio
- ☐ Every link works

Interview

- ☐ 5 stories prepared: leadership, teamwork, success, failure, struggle
- ☐ Can explain each project's architecture in two minutes
- ☐ Daily DSA practice without AI
- ☐ Three pro questions ready to ask
- ☐ Thank you note sent; questions written down after

Sources and further reading

- LinkedIn Help: Skill Assessments, no longer available. [linkedin.com/help/learning/answer/a1690529](https://www.linkedin.com/help/learning/answer/a1690529)
- TechInterview: What actually changed in tech interviews in 2026. techinterview.org
- Business Today: TCS fresher hiring hits 25,000 in FY27 (April 2026). business.today.in
- Outlook Business: Infosys plans to hire 20,000 freshers in FY27. outlookbusiness.com
- OwnYourCareer: Indian tech freshers in 2026 (citing Xpheno and NASSCOM data). ownyourcareer.in
- Keyword counts: my own count of skill tags on 100 Naukri job posts for fresher developer roles, November 2026. A snapshot, not a permanent rule.
- Recruiter search filters and the 5 stories and pro questions frameworks: short videos by working recruiters and placement coaches on LinkedIn and Instagram, summarised in my own words.