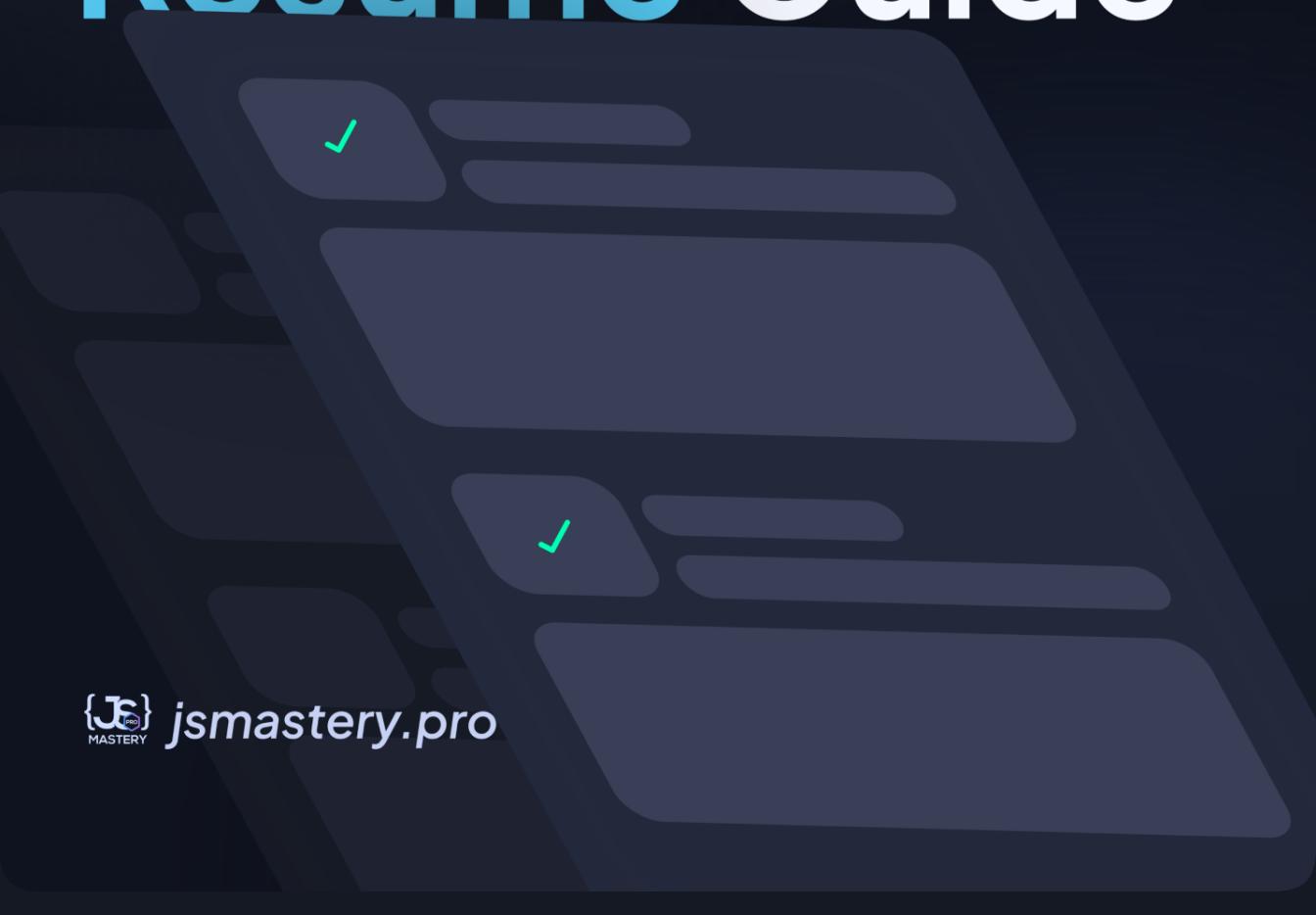




STAND OUT AND GET HIRED

# The Tech Resume Guide

A dark, abstract background graphic consisting of several overlapping, rounded rectangular shapes in varying shades of dark blue and black. Two small, bright green checkmark icons are placed on the left side of the graphic, one near the top and one near the bottom.

# Chapters

- 1 Introduction**
- 2 How do you write a Good Developer Resume?**
- 3 How to stand out from the rest with your work impact?**
- 4 Common Mistakes**
- 5 Different Levels, Different Paths**
- 6 LinkedIn**

## CHAPTER 1

# Introduction

Why do you need a developer resume? What factors that you need to take care of? What are the factors you can't control?

## **Introduction**

---

Software engineers know that there is no one fit-size solution for anything. Be that developing a website, solving an algorithm, or designing an architecture. While this guide promises to deliver great content to help you ace a job interview, it doesn't guarantee a job upfront or a recruiter call every time.

There are plenty of factors that contribute to job hunting. While few are in hands, few aren't. It's that dead simple.

### **So what are the factors that you need to take care of?**

There are plenty of factors that contribute to job hunting. While few are in hands, few aren't. It's that dead simple.

#### **Skills**

You must have relevant skills to apply for a specific job.

#### **Experience**

Some industry experience can make your resume stand out more. (Don't worry if you don't have any experience right now. In the following chapters, I'll discuss how to get experience as a beginner.)

#### **Communication**

You should at least communicate well in English (as its international language), if not a fluent speaker.

## **Introduction**

---

**Then what are the factors you can't control?**

### **Position**

The position might have already been fulfilled—without you knowing about it.

### **Competition**

You might have a strong resume with great expertise, but someone might have more than you to impress the recruiter. Especially after Covid-19, you will have to compete worldwide—if applying for remote work.

### **Location**

Although most of the jobs are remote these days, people who are near or don't need to relocate are preferred by hiring staff.

### **Seniority**

Certain companies only look for specific seniority roles. If you are just starting or have less than a few years of experience, you're likely to be overlooked.

### **Luck**

Luck always plays a role, take it or not. To make it less of a factor, you can go for referrals. (In the following chapters, I will share the tips on how to get a good referral).

## **Introduction**

---

Now you know what it takes to be ready for a job/role. Let's move on to writing a resume than can sell you for the position you have been applying for. A great resume:

- Showcases rare and valuable skills that are exactly what the job needs.
- It's built on years of experience you gathered that makes you stand out from the crowd.
- It shows a solid progression throughout your journey. You take upon new roles/challenges/responsibilities and are not afraid of learning new things.

Remember that preparation and persistence will make a difference. One without the other is an invitation to rejection. To get your dream job, you have to do the hard work. Staying positive, focused, and creative will help you climb the staircase of victory.

I recommend reading this guide to level up yourself.— [How to be a Kickass New Software Engineer guide](#). It's written by a self-taught engineer, Raymond Gan.

(If you get a page not found, log into your linkedin account first)

Keep looking for some inspirations or stories about how people went from ground zero to software heroes. It will not just boost your morale but also guide you like a map. You know, one can learn a lot from other experiences.

Now we understand what it takes before creating a resume; it's time to write that resume.

## CHAPTER 2

# How do you write a Good Developer Resume?

There are some essential points that you must take into consideration while writing your resume. Learn what are the essential points to create an outstanding developer resume in this chapter.

# How do you write a good developer resume?

These are some essential points that you must take into consideration while writing your resume:

## 1. Years of Experience:

- How long have you been working?
- The recruiter will check your education and year of passing to calculate how long it has been for you? They will internally match these years with the years of experience they need. They will also evaluate you with other candidates. The more experience you have, the more are your chances of getting into the \*Yes pile\* of resumes.
- If you make this info hard to find, you might end up in the \*Reject pile\*. Make sure to provide transparent information.

## 2. Relevant Technologies:

- How much relevant experience do you have?
- E.g. If you're applying for a Frontend role and know a little of backend stuff, recruiters will prioritize you over other applicants that only know frontend.

## 3. Work Experience:

- How much relevant work experience do you have?

- Do you come across as someone who has consistently delivered the impact?
- Don't misunderstand the work experience with the year of experience. While the year of experience is the time you have spent so far, the work experience is the impact you have delivered in the entire industry.
- You might have solved a complex problem in your job/internship or even a freelance project. Be that building a highly secure API, catchy frontend website, or marvelous UI/UX architecture. It's all measured in your work experience.
- Another awesome thing you can include in this is your open-source work. Have you contributed to any open-source project which is famous and widely used?

#### **4. Ground Rules:**

We need to follow a few things to write a professional resume. Make sure your resume follows these traditional points:

##### **Good grammar and No typos**

- Your resume represents you. It has to be the best. If you allow typos or poor grammar, it will reflect your personality carelessly—someone who doesn't pay attention to details or doesn't have a good command of the language.
- Use free spell-checking tools and grammar checking tools such as Grammarly.

- Even better is to ask people to re-read your resume.

**Basic contact details:**

- Include your email address and relevant contact information, like phone number, city & country where you live, at the top.
- Keep it short. You don't have to add your full address or other unnecessary details.

**Dates in reverse chronological order:**

- Point your work and education experiences with precise dates.
- List them from the latest on the top to earlier ones underneath it.

**Don't include photos or non-required personal information**

- Many people make this mistake.
- Including photos or personal information, like date of birth, relationship status, and religion, creates bias. It's about your skills and not looks.

**Two Pages or less:**

- It's not a strict rule, but wise to follow.
- If you're a fresh grad or someone with 1-2 years of experience, it's not expected from you to fill two pages. A one-page resume is enough here.

- You can go for a two-page resume if you have ample work experience that counts. Still, the max limit should be two pages. Recruiters don't have time to go through your 3-5 page resume in a pile of 100+ resumes.

## 5. Simplicity and Consistency

Resumes that are simple, concise, and easy to read will be read more. Writing well is an underrated skill in a resume. Read more on how to write better [here](#).

### **Clear, neat, and consistent formatting:**

Use the same formatting, font size, and font family throughout the resume. It helps for a smooth read.

### **Bullet points:**

- Short sentences with bullet points are easier to read over long, overly verbose paragraphs.
- Bullets points are easy to scan for recruiters. They are used to it.
- Avoid sub-bullet points. It clutters your resume, making it harder to read.
- Avoid dashes (-) for bullet points too. It looks out of place and doesn't stand out well at first glance over bullet points.

**Date formats:**

- Format like 08/07 – 20/12 is hard to understand. The reviewer has to think over the date, “Oh, it’s from the 8th of 2007 to the 20th of 2012”. The year is not differentiated from the date.
- Instead, directly use the standard format — August 2007 – December 2012. Easy to read in seconds.
- Don’t forget to stick with the same format throughout the resume.

**Resume format:**

- Do not, I repeat, do not send any format other than PDF.
- Formats like .doc, .rtf display content inconsistently on different machines. What you see might differ from what the recruiter sees. You don’t know reviewers/recruiters’ machine specifications, do you?

**Other Information:**

- Avoid that trivial info that doesn’t bring value to your resume.
- If you have something important that can help you stand out, e.g., your school & achievements, maybe you presented as a leader of your batch in your college, or that university competition—both technical and non-technical (sports), add it at the end of the resume.

After the generalized points of the resume comes its structure. Let’s understand how to structure a resume:

There are two types of software developers: one who had their first job and one looking to land a job. You'll want to structure your resume differently depending on who you are.

For those who had a job, your work experience, professional skills, and the technologies you worked in that company matter on your resume. While for the new grad, it's their internship, open-source contribution, or a personal project that helps them jump out of the crowd.

When you're a student or new grad, you often feel you have little to no experience to show. Even though that's the case, you can always make the most of what you have. Start building personal projects. Go for some freelancing projects. Apply for internships. You can do anything as you're just starting!

**Here are the experiences that catch the eyes of people reviewing your resume the most, in priority order:**

**1. Real-world work software development experience:**

- If you have/had been working as a part-time or full-time developer, list it and your achievements.  
Show how you delivered and progressed in your career.

**2. Internships:**

- You can attract attention with the internships on your resume.  
But don't forget that there will be many people with internships while applying to large tech companies. Specify your contribution and results from the internships.

### **3. Open Source Projects:**

- If you have created a project with many contributors, show it off.
- If you contribute to a certain open-source project frequently, show it off.
- If you have taken part & contributed as an open-source contributor in any organization, show it off.
- If you are not in any of these categories, it's time to do something. Find a project, join an organization or create something on your own.
- You can find open-source programs/competitions [here](#).

### **4. School/College Details:**

- Especially if you have attended a highly prestigious school/college and have high grades, this can be a remarkable addon.

### **5. Projects:**

- Have you built any project which has some great functionalities?
- Do you have something that can show the impact or metrics of your project?
- You can list your creative, complex projects in the projects section of your resume. Link the source code on Github with a good README.

### **5. Tutoring or Leadership::**

- Have you been some part of NGOs where you taught some programming?

- Or you might have taken upon leadership in your college where you led your batch. Anything that you have collaboratively done to do a project or teaching. It shows your team's skills & proficiency in the field. Don't miss adding that.

## A new grad resume structure:

1. Experience
2. Projects
3. Education & Certification
4. Languages & Technologies
5. Interests

Fit all these on one page. As you have more working experience, a second page will make sense.

## How can new grads and interns grab the attention of recruiters?

Sebastian Prieto Tovar and Claire Taylor have recruited hundreds of students and interns for Uber and other tech companies. Here's the advice they have for people starting:

- We always tell students to read the job description, then amend their resume accordingly. And reach out directly to the recruiter when you can.

- We look for your studies, relevant courses, and what you're best at with studies on a resume.
- We also care about extracurricular activities, hackathons, working in teams outside school, apps, websites, and other cool things you created outside school.
- Do spend plenty of time preparing for the interviews themselves.
- Read about the company, watch the videos, learn about the culture, read stories about employees, and understand the job description.

## **Work Experience structure:**

### **1. Work Experience at or near the top of the page**

Your current title, company, and past few years of experience

### **2. Have a Languages and Technologies section on the first page**

List things you are an expert or, at the very least, proficient at in order.

Mention the languages and frameworks that the job description mentions.

Don't bother listing non-relevant technologies like Jira or Trello.

### **3. If you have spent a long time at one workplace**

List key projects you developed & shipped. You would like to mention roles or responsibilities you take upon promotion, if any. It shows progression. Write all these in a new sub-work section.

#### 4. Education details

It becomes less important as you progress in your career. If you have less than 1-3 years of experience, it's good to go with. With more experience, you would like to focus more on work experience than past education details.

#### 5. Certifications

List the relevant certification related to the job industry below the work experience. Make sure these certifications show some quality. Trivial certification, like LinkedIn certification, doesn't add significance.

#### 6. Projects

The more experience you have, the less relevant outside-work projects tend to become. List only those projects that are outstanding in the extra curriculum section.

#### 7. Interests

Point out a few things that interest you to make the resume more "human." Keep it short.

**As a rule of thumb, avoid listing your expertise level.** Instead, list only languages you feel proficient with, and list your most vital languages and technologies first.

There is a special section that has gained different opinions from different people. Guess?

Yes, it's the **summary section!**

## Do you need to have a summary section?

Many resumes start with a summary or profile section at the top. Here is a thing: recruiters or hiring people barely read this section at first scan. They only do so when they have decided to proceed with your application.

For less experienced candidates, it's not wise to have this section unless you customize it for job listing, showing how a great fit you are!

So, where the summary section can be helpful?

### Senior/Standout profile:

Describe your motivation, what you're looking for, and your year of experience.

### Remote position:

If you're looking for a fully remote position, the summary section comes in handy. You can clarify that in the section.

### Role change:

If you are someone from a non-technical background who wants to switch to a technical career, having a summary specifying your motto will be very helpful.

So if you decide to go with it, make sure your summary section isn't too short or too long.

**So far, we have talked about the resume's general part and its structure. Let's focus on the crucial part of a resume, i.e., Work Experience.**

## CHAPTER 3

# **How to stand out from the rest with your work impact?**

Catching the eye of the typical recruiter who is wading through scores of resumes can be quite challenging. Here's how to make it more likely that your resume will get noticed.

# How to stand out from the rest with your work impact?

While listing your work and projects, focus on what you achieved instead of what you did.

You want to convey that you're self-sufficient and that your work has impacted the team & then the company. You're well aware of your valuable contribution. But how to stand out? There is a simple formula that Google recommends:

**Accomplished [X] as measured by [Y] by doing [Z]**

Learn more about it [here](#).

Effective resumes should contain two things: Responsibilities and Accomplishments.

To stand out from the crowd, you must point out how you consciously and proactively added value through your work.

Edit your achievements with these points:

## 1. User Numbers

- Quantify your impact wherever you can
- Instead of saying, *"Built a tool widely adopted by the company,"* say, *"Led a team of 3 developers to build a dependency injection framework adopted by 15 teams and all 50+ developers at the company."*

## **Work Experience**

---

- Numbers can be anything:
  - Number of people on your team
  - lines of code
  - code coverage % before and after
  - number of users
  - number of installs
  - ratings

### **2. Use active language**

- Show what you have done and how you have been proactive.
- Use active verbs like “led,” “managed,” “improved,” “developed,” and “rolled out.”

### **3. Specific languages & technologies**

- Mentioning technologies in your work experience is powerful
- Ensure these technologies overlap with the languages and technologies you mentioned in the standalone section.

### **4. Keyword stuffing**

- Resume filtering is done by discarding resumes that don't have specific keywords.
- A workaround is to read the job description and pick all the necessary keywords.

- An excellent way to keep these keywords without losing professionalism is to structure them in the “languages and technicals” section, where you list technologies you’re familiar with or relevant to a job.

When it comes to technical experiences, some people falsely state things. Some might even shy away. Don’t claim untrue things, but do aim to paint a great picture of yourself.

Here are a few tips for you:

**1. Talk about yourself, not your team. Avoid using “we.”**

- The resume is about you and not your team. The hiring team wants you! Always use the first-person approach. In most cases, you can drop the “I.”

**2. Be concise but not humble.**

- Don’t hide your achievements.

When in doubt, inflating them on the borderline will hurt less than hiding them. You never know how these small, impactful achievements can help you stand out from the competitive crowd.

**3. Side projects & Open Source contribution**

- Mention these in your project section with the impact-result contribution model.

## **Work Experience**

---

Provide GitHub source code links with a proper explanation of how your project works. Create nice READMEs containing project details with the installation process.

### **4. Extra curriculum activites**

- Do talk about these activities at the end of the section.

Maybe you're a technical writer or content creator on social media—mention it. Provide two links to your popular article.

Showing off high-quality or standout activities can make a good resume even better.

### **5. Mention your learnings**

- Learning new and valuable skills proactively is a positive trait that hiring managers appreciate.
- Do mention if you picked a technology to speed up a project, gain knowledge, or stretch yourself.

### **6. Don't highlight negatives.**

- You don't need to list things that don't show you in a good book.
- Avoid mentioning failed projects, low GPA/CGPA scores, etc.
- Remove items that don't convey you're a good fit for the role

We have mostly covered what should go on a resume. Yet people make some trivial mistakes. What are these? Let's check that out!

## CHAPTER 4

# Common Mistakes

In this chapter we will discuss some most common resume mistakes made by software engineers as a fresher and some essential tips to avoid them.

# Common Mistakes

## 1. Creating hard to scan resumes:

- Multi-column resumes are hard to scan for managers.
- Simple formats are better. Go utmost for a two-column layout.

## 2. Too much bolding

- A resume should have minor and consistent bolding for critical parts, such as dates, titles, and companies.
- Don't bold random parts that don't **make sense** at all. Just like this text!

## 3. Flashy resumes

- A developer's resume is 95% about the content and 5% about the style.
- It works if you're applying for a UI/UX role as it's more about style with creativity.

## 4. Inconsistent formatting

- Pay attention to the fine details. Keep everything consistent.

## 5. Sloppy phrases

- Avoid unprofessional language. Don't use "etc.", "so on," etc.

## 6. Using internal acronyms and jargon

- Avoid using internal project names that people outside the company don't understand.
- E.g., "*I did FD in aplha2.0 project*". A recruiter can't guess what your FD is in aplha2.0.

## 7. Not reflecting on the job and not tailoring the resume for the position

- Say if you know both frontend and backend but applying for a specific frontend role. You will want to present yourself as someone who knows the front end enough. Backend is a sidekick here.
- Always keep the job position in mind, then tailor your resume.

## 8. Using cliches over statements backed by evidence

- What does the following sentence tell about you? "I am a team player and a fast learner who can hit the ground running."
- From the point of the recruiter, it doesn't provide any information. It's not backed by any fact or evidence that can state it.
- Let your actions speak for themselves

## 9. Being too verbose

- Avoid blocks of text or long sentences. It doesn't help for an easy read. Keep it short and up to the point

## **10. Unnecessary details**

- Do not add photos or irrelevant contact information.
- If not needed or not in the job's context, do not mention spoken languages. Most companies will only care about your proficiency in English.

## **11. Self-rating your skill level**

- Do not rate your proficiency in any language or technology using percentages or scores. It's not necessary.
- No one knows anything about a programming language/technology.

## **12. Non-clickable or full links**

- Cross verify if, by mistake, you have added non-clickable links.
- Add proper links to GitHub, websites, and other media. Recruiters will not copy-paste the name/username to see it.
- Make clickable links and hide the full URL behind a name that describes what the link is for.

## **13. Stale Linking**

- It's better to leave it empty rather than linking to stale or not-touched-in-decade websites or GitHub profile links.
- Link only if they are up to date.

## Common Mistakes

---

So many guidelines, and you can still commit a mistake or two, no matter what. To recap, consider doing most of these to make your resume better:

- Write two different resumes, then create a third one from the best parts of these two.
- Try to estimate the impacts of your project. A rough estimation is better than no specific.
- Do grammar check, not just spell-check.
- Ask friends or family to proofread
- Seek out feedback on various community
- Do a keyword check for your resume. Manually or with the help of automated tools.

## CHAPTER 5

# Different Levels, Different Paths

Knowing what to focus on based on your current level is necessary. The journey to different career levels starts at different points. Let's figure out what to focus on in your resume respective to your level in the industry.

# Different Levels, Different Paths

Depending on how much experience you carry, you will want to focus on different areas of resumes:

## Resume for Students

If you're still a student, knowing fundamentals is an essence. While learning software development, you would want to focus on these things for a future job search:

### Internships

One of the best experiences you can have as a student is working as an intern.

Apply in natural problem-solving industries.

### Campus Activities

Your university/college might offer plenty of opportunities to work with a professor, lead a laboratory, arrange events, etc. Make the best of these times.

You will get hands-on experience mentoring/leading others or taking the initiative.

### University Projects

You will undoubtedly have class projects or, to the least, the final year project.

Consider putting extra time and effort into these projects.

Publish them so that anyone can download or try them. Using GitHub will be super beneficial.

If possible, buy your domain. This might cost you bucks, but it'll be worth it to show the interviewer what you have. Clearly, it will show you as someone who knows deployment to the least.

## Resume for Bootcamp Grads

As a Bootcamp grad who has likely learned basic software development skills, your time spent studying these will be shorter than college/university students. It would be best to compete with these graduate people and other Bootcamp grads.

But no worries if you have the capstone project on your resume. So what do you need to focus on?

### Projects

It's universal for everyone looking for a job. Develop as many projects as you want, but have that one great project!

To stand out, you will want to showcase better projects. Follow these tips:

- **Good READMEs**

Aim to write a README with a good summary: screenshots, details on how to run & test.

Refer to [this](#) for inspiration.

- **Contribute to other open-source projects**

Find projects where you can contribute. Take part in the open-source programs I mentioned earlier.

If you can merge a pull request, you can claim that you have contributed to a project used by more than a certain number of developers.

- **Turn interview projects/challenges or learnings into public projects**

You can do more than submit projects or do coding challenges for a job interview.

Without caring about the results, continue working on these projects/challenges and submit even better-polished solutions online on GitHub.

If you cannot reach the next round, ask for feedback, and build that feedback into this project. If you do even better, don't hesitate to send a cold email with these latest changes. You might get another chance to do it better. No one knows unless you try.

- **Freelance projects**

Try freelancing once in life while you're in search of a job. A freelance project on the resume can tell a lot.

Landing a good client at first can be challenging. If your goal is to get experience, keep the focus on finding projects that can add value to your portfolio.

Publish screenshots and code snippets of the work on the portfolio page afterward.

You don't need to disclose how much you got paid. The goal is to show your knowledge.

## Resume for Career Changers

When you have changed careers from other professions to software development, most of the advice is the same as for a Bootcamp grad—you might have some training behind you. Here is advice for you:

- **Summary**

Have a resume that explains your motivation/reason for a career change

- **Languages & Projects**

List towards the top of the resume, showing that you have hands-on software development experience. Follow the guidelines for listing a project, i.e., source code link, good READMEs, demo link, etc.

- **Past work experience**

Keep it concise and have a one-sentence summary of the job. Focus on any points or skills transferable to software development, e.g., proactive learner, team leader, etc.

- **One-page resume**

Try to fit on one page. Don't over-share past experiences if they aren't beneficial in the software field.

## Resume for Career Breaks

You might have gone for many months—or years—without a job or work in a software field. It might be why you were pursuing something, or you wanted to take a break. Either way, there's a gap in your resume. How do you present this?

- **Career break years**

You don't need to explain the breaks that have happened more than a few years ago — 4 or 5 years. Recruiters or hiring managers will rarely care about it.

For recent career breaks, use your good judgment on whether you want to add it.

Tell a story of your career break.

e.g., Due to COVID-19, I decided to take a break to spend more time with my family.

Read [this article](#) on how to address career breaks better

## Resume for Senior or above Engineers

The more experienced you are, the more likely you stand out based on your experience and past titles. For someone having experience of over 7-8 years, your challenge will be to keep your resume relevant to the position you're applying for:

- **Career break years**

the experience you carry, the hiring team will look forward to this summary section to know about your motivation behind applying to their firm.

- **Soft Achievements**

Mention soft achievements – mentoring junior developers, leading a team, and other activities where you helped others.

- **Impact & Influence**

On top of company work, you can mention what kind of impression you made with your work.

Mention what teams or organizations you influenced for specific outcomes.

- **Education**

As you continue gaining more experience, education will have less value. You can move the education section to the second page. Only keep standout details in one line or two.

## **Resume for Tech Leads**

You might have gone for many months—or years—without a job or work in a software field. It might be why you were pursuing something, or you wanted to take a break. Either way, there's a gap in your resume. How do you present this?

You can learn more about the manager & tech lead archetypes [here](#).

Follow these guidelines when applying for the tech lead position:

- **Showcase how you helped teams**

Hiring managers hire tech leads when they feel their team is behind on delivery, struggling with high turnover, or delivering poor quality.

As a candidate, you will want to showcase these points where you helped your team to excel, increased speed, improved code quality, or repaired stakeholder relationships.

- **Give examples of how you made your team successful**

You will be expected to have a track record. Make it easier to find.

- **Specifics and Context**

Specify the size & makeup of the team—3 frontend, 2 backend & 1 QA engineer's team.

Mention your major technical choices taken within the team.

- **Outcomes and Activities**

Please talk about the daily scrums and how you managed to make them happen within the deadlines.

Include proof points for delivery; e.g., managed a team of x developers that delivered Project X on time and within budget.

# Resume for Engineering Managers

Most percentages of the guidelines shared for tech lead positions apply to the role of engineering managers. But you would like to provide more details:

- **Tell a story** about how you achieved results, improved your team's performance, & how made your organization better.
- Be **familiar with the values of the company** you're applying for. Do your research on the organization's culture.
- Talk about the type of **challenges you are fit to solve**. Point out what kind of problems made you achieve your best results?
- Get **feedback** from your peers or friends.

## CHAPTER 6

# LinkedIn

Along with your resume, you will want to get ready other things like LinkedIn, GitHub Profile, or cover letter. Let's talk about how to create a better LinkedIn profile in this chapter.

# LinkedIn

Along with your resume, you will want to get ready other things: LinkedIn, GitHub Profile, or cover letter. Let's talk about how to create a better LinkedIn profile:

Resumes and LinkedIn profiles are like the yin and yang. You directly apply with a resume, and opportunities can come and find you with your LinkedIn profile. It's smart to invest equally in both.

## Headline

Set a headline that represents what you want to be found for. It should summarize the message you would want to convey.

If you set nothing on your profile, your current position will appear.

Consider adding the “|” separator character to add more information to your headline. For example, “Frontend developer with 2 years of experience | JavaScript, Express, MongoDB, Next.js”.

A tailored headline will rank your profile higher with keyword searches that grab the recruiters' attention.

## Current Position

Describe your current role properly.

For example, if you're leading a team of frontend developers, you should write “Frontend Lead” or “Frontend Team Lead”. Anything that best describes your role.

If you don't have a current job, LinkedIn will rank your profile lower. Make sure not to leave your headline empty, even if you don't have a position. You can highlight the skills you know in that case.

## **Summary**

Mention keywords, technologies, or languages you're familiar with.

Go for a longer summary that best describes your values or motivation in the field.

You don't have to fit everything on a page on LinkedIn. In fact, with more content, you could rank higher in search results. So it's fine to be verbose here.

## **Professional photo**

A professional photo is one where your face can be seen properly without no one else in the picture.

## **Work Experience**

Omit work experience that doesn't support your professional career.

Your profile should tell a good story of your professional side. Remove positions that don't strengthen this image.

## **Open for work**

Do not add "Looking for opportunities", "open for a new role" or something similar on your headline.

It doesn't add much value. Instead, it will take the space that you might have used to present yourself or highlight your skills.

You can activate the option given by LinkedIn to let recruiters see whether you're open for work or not.

## **Networking**

Connect with more people. React out to the great minds!

The close you are connection-wise with the recruiter, the higher your profile will rank.

## Location

Location is often a filter that recruiters use.

If you're moving to a different location, you can update it. In case you're looking for remote positions, you can specify that.

# **The End**

Reference taken from, Special Offer and Thank you.

## Reference

The Ultimate Tech Resume ebook results from references taken from multiple books, articles, and other Internet stuff.

Check:



### The Tech Resume Inside Out

[Get the book →](#)

A major part of the ebook is from a highly recommended resume guide by a famous author—**Gergely Orosz**. You can find his book with even more in-depth content.

**Thank you!**

# Before you go

This eBook was curated from the insights of the best hiring managers in the industry and taught you how to create the best CV.

Inside of the **JSM Masterclass Experience** we take that a step further and help **YOU** on a personal level. That includes beefing up your portfolio with never-before-seen projects, reviewing your portfolios, CVs, LinkedIn & GitHub profiles, and helping you position yourself as the best person for the job.



**Become a Master Developer With the Project Experience to Prove It.**

Until next time,

Adrian,

JS Mastery