

How to get a front-end job

If you want to get a job at a MANGA company without any degree or not much academic background you'll have to put in the work for a couple of months grinding your ass off and learn all the required skills to even get passed the initial screening.

Or, you could learn the bare minimum, build projects, get a job at a small company and then work your way towards your dream job. The choice is yours.

Required technical skills

HTML, CSS, and JavaScript are the building blocks for learning front-end web development. These three are the most important things in front-end development and you don't need to completely master all of them, knowing enough HTML to make a solid structure, enough CSS to nicely style your sites, and enough JavaScript to add interactivity into your projects will be enough to land you an internship.

Out of all the three JavaScript is the most important and you should spend the most amount of time learning it. There are some core concepts in JavaScript such as scopes, closures, and the DOM which are one of the most important things that you must know when applying for a job.

Pick your framework

After learning the three core concepts you need to learn a few more things like a JavaScript and CSS framework which makes your job a whole lot easier.

- A JavaScript framework combines little things like page routing, styling, and state management into a single boilerplate type project for you to use.
- A CSS framework provides you with pre-built styles, components, and classes which you can just copy-paste and modify them to your liking.

Although frameworks provide a lot of functionality the problem is that there are too many of them. (especially JavaScript ones.) Pick only one that you like the most or the one that is most popular, or the one which is required in most job postings in your area.

There are also other things like version control, pre-processors, search engine optimization, and accessibility which are also good to know and are easy to learn.

If you deep dive into front-end web development you'll learn about design systems, server-less computing, UI/UX design, and progressive web applications. These skills are the ones that will make you stand out from the rest.

Creating your portfolio

Your portfolio is the most important thing that will help you land a job. It will help you stand apart from the rest, and if it's super impressive you won't even need to apply sometimes.

You need two things in your portfolio, one being the portfolio website itself and the second one being your profile on GitHub. GitHub is the most popular code repository platform and having a good, well structured GitHub profile can be a advantage when applying for jobs.

Make sure you have a great introduction section and pin your greatest projects to the front page of your profile.

Another important thing is that the read me file of all of your projects that are pinned or displayed in your portfolio website should be complete and straight forward. A good read me consists of the following:

- A short introduction about your project.
- The technologies that you have used.
- Instructions on how to run this project.
- A link to see the live preview.
- Screenshots and images of the project.

The second thing is going to be the portfolio website itself. Experiment with your portfolio website as much as you can, add fancy animations and graphics but don't overdo it. Make your portfolio website as simple as possible and the main focus should be about displaying the projects that you have worked on to your employer.

There is no rule about which technologies you should choose to create you portfolio website. Choose the stack you're most comfortable with or learn a new technology to implement it into your website.

Picking projects to showcase

You shouldn't put all of your projects on portfolio. Pick only the ones that you are most proud of and make sure you understand the code so that you can explain it in the interview if asked.

Don't make projects that every other person is making. Find a problem and try solving that or making an outdated analog process digital. Build something that you can show off to other people proudly.

The most important thing is that your project should have a nice design, responsiveness, and it should be simple to use because the interviewer doesn't have time to go through every part of your code.

Like the read me file of your project, if you can provide some documentation about your projects that will be a plus point and if have done testing and show it your read me file your project will be better than most of the candidates automatically.

Creating your resume

Apart from your portfolio, your resume is the first thing that the interviewer will look at when you apply for a job. Most companies use an application tracking system which filters out the candidates based on which one is most relevant. Even if they don't use an ATS, the recruiter only spends a maximum of two minutes looking at each resume.

Your resume must have the top keywords that were in the job posting description and be as simple as possible so that it can be read in just a few minutes. This way your resume will be able to pass the ATS machine and impress the person reading it.

These are the things that you can do to make your resume look nice:

- Don't use more than three fonts and colors.
- The flow of the text should be from top to bottom.
- Include the most important information at the top.
- Use a spell checker to remove any spelling or grammar mistakes.
- Don't add any irrelevant information.
- Make your contact information clearly visible.

Creating a cover letter

A cover letter is probably not required most of the time but if the job posting requires it then you can send it along with your resume.

The important thing for cover letters is to make them short. You only need to write what position you are applying for, what you are doing right now, and why you're a good fit for the position. The crucial part is where you explain why you should get the job.

Create an online presence

In this digital age if you don't have an online presence, you don't exist. This is true because the internet helps us to be visible to people that live across the globe and they can see what we are working on and if they like your work they can collaborate with you easily and if you get noticed by a hiring manager you can land a great new role too.

Programmers don't need to be active on every single social media site on the internet. Twitter and LinkedIn are enough for you.

LinkedIn is the social media site where anyone can share their achievements and progress and contact hiring managers or CEOs of companies. You can add your past experiences, awards and certifications, and the skills that you can share with recruiters all over the world.

Twitter is a place for everyone to hangout and most programmers and startup founders are quite active there so it's a great platform to share your thoughts and ask

for guidance from experts. You can also gain better insights about a company and senior developers and how they work and solve their problems.

Twitter is also a great place to learn in public, you can participate in online challenges like thirty days of X and hundred days of X. Sharing your progress with others will help be more confident. Don't just share your achievements, share your failures and solutions to your problems and insights that you gained on your journey as well.

You're very likely to find a group of like-minded people who will help and support you in your learning journey. They may offer a word of encouragement and advice or even connect you with someone who has a job for you.

There are also other ways to create an online presence like starting a blog or contributing to open source projects. If you start a blog, you can learn a lot from writing down the things you learn and can also help other people by sharing your experience with them.

Answering questions on Stack Overflow and Quora can also help you get recognized online. This will also tell you about how much do you really know about a certain topic.

Finding the job

Finding and applying for a job can be scary if you're looking for a first timer looking for a internship or junior role. Most job postings for developers aren't written by developers but are written by managers or human resources which don't know most of the technical aspects for hiring developers so they'll just put every single requirement that comes to mind in the post.

They do this because they need to filter out only the best candidates possible and this in return can be a lot of requirements to meet especially if you're a beginner.

But don't try to get overwhelmed and if you meet at least 50 percent of the requirements then go ahead and apply for the job. Chances are you'll get selected and even land the job.

Apply for as many job offerings as you can because this will increase your chances of getting hired. Even if you don't get selected for the job you'll build up experience and find your weak areas. Improve on them for the next interview and eventually you'll land the perfect interview.

If you see a job posting, contact them via email first. Introduce your self, tell where you found the ad, what are your skills and why do you think you're a perfect candidate for this job.

Attach your resume and provide links to your GitHub and LinkedIn profiles even if they didn't ask for it. Sometimes people post contract jobs or hire freelancers for a single project. If you have built any project like they need include links to it in the email also.

Some companies offer referral bonuses to their employees if they can help in recruiting new talent. Make connections on LinkedIn and ask if they can refer you in their company.

The interview process

The first step in the interview process in some companies will be a simple 15-minute phone call. The recruiter will determine two things from this phone call, how well can you communicate with others and the second one will be to check if you have the basic knowledge of things listed in your resume and portfolio.

prepare for the call a day before, make sure you're in a quiet room and your phone is charged, you are sitting in a comfortable position, make sure you're not hungry or have to go to the bathroom before or during the call, re-read your resume and go through all the conversation you've had before the call with interviewer. Researching about the company before hand is also a good practice.

General questions

When the interview starts, you're likely to get some general questions. They should warm you up and make you comfortable.

The question most people fail to answer is *Tell me about yourself?* This is a simple question but people may not be able to answer it. Your answer should include the following point in order to impress the recruiter.

- Why do want to become a front-end developer.
- Your skills set and the things you want to learn.
- Why do wanna work in this company.

The interviewer can also ask questions like:

- What projects have you worked on recently?
- What's your specialty?
- What is the project you're most proud of?
- What do you know about our company?
- Why do you want to be with us?
- How do you stay up to date with the latest tech trends?

Technical questions

Your recruiter will also ask you technical questions for which you have to be prepared. These questions can range from easy to questions about algorithms and data structures.

Some of the questions that can be asked in a front-end job interview:

- What's the difference between `let`, `var`, and `const`?

- What is *this* in JavaScript?
- What is event delegation
- What is a prototype?
- What is a Worker?
- What is a Stack and a Queue?
- What is a Big O notation?
- What was the last tricky bug you solved?
- What are things you don't like about JavaScript?

When you start answering the questions don't say everything quickly, keep calm and take your time while answering and try to explain as much as possible. Even if you don't know the answer to the question don't say that you don't know or ask for the next question.

Tell them that you haven't used the particular thing you are asking or if you've used a different approach.

The last part of the initial call are your questions. You should have a few of these prepared:

- How long is the recruitment process?
- How many steps it has?
- How big is your team?
- Are there other teams in the company?
- What is the culture like?
- What projects you usually work on?

Asking these questions will make it seem like you're a professional who really cares about the job.

The recruitment task

Some companies might give you a task designed for the role you're applying to and give a time period to complete it. You can take the task home and complete it at your own pace.

Keep in mind this is a great way for companies to outsource some of the work they have and it's not necessary to complete the task. If you really want to get the job and find the task easy make sure the code that you write to complete the task is the best code that you've ever written. You can also add a cherry on top by adding some extra features to show off your skills.

The on-site interview

Companies will invite to at their offices to conduct an on-site interview and they'll ask you questions related to your portfolio, algorithms and data structures, and general

computer science questions.

The main point of an on-site interview is to evaluate you by checking how well are your communication skills, presentation skills, and can you handle pressure or not.

Keep your calm, how you dress and present yourself at the interview matters a lot. First impressions matter a lot in the corporate world so make sure yours is the best.

Take a moment before you answer questions, if you don't understand the question ask for the interviewer to repeat it. If you don't know the answer just say that you haven't come across this yet.

Watch out for red flags

While the companies you apply to will assess that you are perfect candidate for them or not you should try to assess if the company is a perfect fit for you or not.

There are many red flags that you need to watch out for when doing an interview.

- Is the recruiter on time for the interview?
- Are they pressuring you?
- Are they rude to you or dismiss you while you're talking?
- They are not interested and are doing other things while you're presenting yourself.

After the on-site interview find more information about the company. If you do this in the beginning it would be better. Find out what current and ex-employees have to say about the work environment and the companies turnover rate is also an important factor to decide if you actually want to work there or not.

Replying to the job offer

If you get selected then the company will send you a job offer letter. If you want to work for them then great! Accept the offer or negotiate some changes that you want.

If you don't want to work there send them an email saying the reason you can't work for them but write it as professional and polite as you can.

Learn from your rejection

If you get rejected then don't feel sad instead write an email to the company and ask them to provide you the feedback about your interview and then improve on the areas that you are weak in according to the feedback given.

The end

If you have come this far then thank you so much for reading this. please feel free to give your feedback to me on Twitter @SyedUmairCodes. If you like this please share it with your friends and colleagues.