



## General Candidate & Role Info

# YRW Technologies CEO Position Info

YRW Technologies owns and operates over 25 (and counting) software products. Currently, all projects are mobile apps, web apps, and desktop software. We have plans to expand into more diverse areas of software development, including new file formats, vehicular programming, and programming for appliances and other non consumer devices.

As CEO / Product head, you are responsible for **end-to-end development** of complete software products + features and updates to our existing products.

This means, you will be responsible for everything from recruiting a dev team to ensuring features are built to standard, before the deadline.

## I need someone who.....

- doesn't just follow existing procedures and methods,
- breaks the rules and norms to come up with their own ways of doing things.
- can reinvent processes to move faster, smarter, and more efficiently than everyone and anyone else.
- Spots every tiny issue with software products, and knows how to innovate them to improve them or revamp them completely.

## Example Products

Please note that these are not products that we own, but rather some of our competitors and/or products that have similar functionality to products which we own.

1. [Soundcloud.com](https://www.soundcloud.com) - most of our portfolio is music related like this.
2. Stealthgpt
3. Semrush
4. Fragrantica
5. Oxyllabs
6. Fiverr
7. Liquid Web
8. Tiktokdictionary

## Main Responsibilities / Expectations

- Have a genuine product vision for everything you develop
- Shipping features in existing products and handling end to end development of complete software products to standard.

- Ensuring the project(s) you're responsible for is working and live 24/7. If something goes wrong, make sure it is dealt with proactively.
- Handling some miscellaneous in-house operations such as SEO content. This does not mean you will be doing the operation yourself, this means you build SOPs, hire / shortlist people to run the operations and manage their performance.

If there is an issue with any of the above, that responsibility falls onto the CEO.

## What skills you need in order to fulfill the above main responsibilities

- Complete hiring from headhunting to shortlisting to hiring, leveling and managing engineers
- Team management
- Project management
- SOP writing (*falls under project management*)
- UI design
- Full stack development (flexibility is a priority, rather than what frameworks you already use).
- Efficient task management workflow
- Keeping organised and tidy

## What the CEO is not responsible for

- Marketing / distribution (Except for some simple SEO operations)

## The absolute MAX I should be doing + Ideal Candidate Working Norms

My goal for a CTO - Not initially, but eventually, I want to get to the stage where I can simply say something like: *"we need to add a functionality that does xyz to address the xyz concern which users are facing"* or *"we need a way to advertise all the careers we have open, why don't we create a new subdomain + hiring portal on webflow and link it on the footer"*

.....And everything gets done without me having to intervene much. Think outside of the box to address future issues that users or we may face.

When starting a new project, **the most I should be doing is writing up the PRD (project requirements document), designing the UI and also building the frontend with React / Next to use as a starting point.**

The same applies for adding new features to existing products. As part of your responsibility, you may also be required to come up with these features / improvements on your own.

Try to identify the bottlenecks our users are facing when using our products. What is their goal of using our product and what is making it hard for them to achieve this goal? Build / improve new features that address these bottlenecks.

## CTO Skills / Experience & Behavioural Expectations

The list of skills below outlines the main issues I face when running this business on my own which has led me to opening this role.

I'm not looking for someone who can simply match my skills. I want someone who can outperform me. This position is for someone who can guide me rather than me guide them.

### 1 - Team management & Recruiting / Hiring

#### **This solves the problem of:**

- Underperforming team which eventually results in delays, poor quality of work etc....
- Slow and ineffective traditional hiring methods

#### **This means:**

You need to be able to **think creatively** to come up with systems to hire qualified developers and designers much faster. Branch off my personal hiring methods if you want as I already know how to do this quite well, but not well enough!

You need to be able to lead a team and have a solid 'law enforcement' strategy (I like to refer to disciplinary action as law enforcement because that's what it is).

### ~~1.5 - RPA (Optional but nice to have) & Other Forms Of Task Automation~~

#### ~~**This solves the problem of:**~~

~~— Slow scaling / growth caused by wasting time on small, 'automatable' tasks~~

#### ~~**This means:**~~

~~Any task that is either:~~

- ~~a) Extremely simple and needs to be done many times in the future~~
- ~~b) Difficult and needs to be done just once or repeatedly in the future~~
- ~~c) Better suited to be automated, outsourced, or handled by a specialist on our team~~

~~....Needs to be automated, outsourced or done by a specialist in our team.~~

~~There are many ways to automate tasks. Depending on the complexity, you can use cron jobs or simple things or RPA for more complex systems. For very complex workflows / systems that may not be viable to be handled entirely by RPA, can be handled by human workers or hybrid human and RPA.~~

~~For any new position that requires RPA or human workers, you will be required to write the correct documentation, such as an SOP for the particular automation.~~

~~You need to be able to know where the threshold is to determine whether or not you should automate.~~

~~Consider variables like cost, usefulness, time consumption and deadlines. Ask yourself; if we spend a lot of money on this now, will we get our money's worth later on? Is the same process going to be something we will need to be doing 5 months down the line? If not, would it be better for us to just~~

## 2 - Functionality & Competitive Advantage

### **This solves the problem of:**

- Building software products that do not carry the capabilities to do what the user intended to do with it
- Software products that do not function as well as intended
- Missing features that should obviously be there to support the user journey

### **This means:**

All functionality must be developed from the user's point of view. Before building anything, think like the person using the product. What are they trying to do? What might confuse them? What could slow them down? Identify every possible problem, friction point, or missing step in their journey and fix it before it becomes an issue. The product should feel smooth, obvious, and effortless to use, with no guesswork required.

## 3 - Unique And User-Focused UI Design

### **This solves the problem of:**

- Generic, boring, hard to use UI
- Software interface design that does not innovate anywhere beyond what we normally see or what UI component is conventionally used for a specific purpose.

### **This means:**

Get creative. The UI layout must be completely original and different to anything else. You need to INNOVATE. Figure out what's wrong with existing UI components and elements and come up with a new, never seen before solution.

Designs should not look like other software interfaces. They must be designed specifically for the functionality which our unique product has. Put it this way - if someone removes the logo, it should still be obvious that this interface belongs to the specific product because it is built around its exact function without taking inspiration from any others.

Here is a good example of what i'm talking about - <https://imgur.com/a/v3eIMmN>

This design for a music sharing web application considers the user's intention and makes it as easy as possible for the user to do what they want to do on the platform.

### **More examples:**

- [Smart home control dashboard](#)
- [Smart Home Control Dashboard 2](#)
- [Property Search Dashboard](#)
- [Delivery Web App](#)

- [Travel app](#)
- [Heatmap UI](#)
- [Crypto dashboard](#)
- [Interior Design SaaS Web App](#)

**In order of priority:**

- 1 - MUST not be replicatable by AI.
- 2 - Completely unique, user focused design that barely takes inspiration from existing solutions
- 3 - Aesthetically pleasing / creative CSS styling. Unique, never seen before design

## **4 - Innovating, improving & Inventing (Creative Thinking)**

The most important skill is to be able to innovate. You should easily be able to identify problems with existing solutions - for absolutely everything.

Never conform. Never copy others. Never be satisfied with existing solutions.

Instead, come up with your own solutions to any problem you encounter. This applies to any CTO-level skill, from hiring to feature conceptualization.

**Look at how things are currently done, whether it's UI design, hiring, managing development, or anything else, spot what's not working, and find a better way to do it.**

For example, if we need to hire two developers within 12 hours under strict requirements, I don't expect you to rely on traditional hiring methods. I expect creative, unconventional solutions that achieve the outcome despite unrealistic constraints

As part of this role, you need to be able to think outside of the box to come up with your own solution to 'unrealistic expectations'

For example, before modern AI language models were invented, most people would not have believed that a full essay could be written in seconds or that a mobile app prototype could be created in minutes. These tasks would have seemed impossible without the current technology.

At one stage, this would have been considered to be an unrealistic expectation....right?

Obviously that's an exaggerated example, but the point I'm trying to make is that you need to come up with a 'never seen before' solution to a goal that sounds unrealistic at first.

To elaborate, the CTO position I'm looking for requires you to forget every 'standardised' way of doing things, instead, you need to come up with everything on your own.

Think about it like this:

99.99% of people are following pre-existing systems for accomplishing anything. Like what they teach you in uni = Therefore, your goal is deemed unachievable if you say you want to do something that no one has done before.

Come up with your own way = possible.

Get creative and do your own thing. This applies for everything from:

- Conceptualisation of product / features
- Hiring
- Team management
- Anything .....

## 5 - User-Satisfaction

**This solves the problem of:**

- Building software products that do not carry the capabilities to do what the user intended to do with it
- Bugs and other issues that hinder useability. This includes CLS related issues as well as things like UI elements not being positioned optimally - basically anything that can frustrate the user, even the tiniest bit.

**This means:**

You need to be able to oversee QA. You need to come up with an unconventional way to ensure there are ZERO bugs.

*I suggest writing down every single user flow before handling QA.*

## 6 - Genuine Product Vision - Coming Up With New functionality, UI etc.....

**This solves the problem of:**

- Having to rely on me for every single improvement / suggestion in regards to how certain functionality should work, UI and anything in between.
- Our products looking like AI vibe coded slop

**This means:**

To a reasonable extent, you will need to have a creative vision, similar to mine for the projects you're responsible for.

This means you need to be able to identify features / improvements that make our projects more satisfying for the user.

Most importantly though, at the bare minimum you will need to have common sense and think from the user's perspective when building absolutely anything.

**This ties into point number 3 above - creative UI. You need to be able to come up with innovative new ideas for our products, both in terms of design and functionality.**

**In a world where AI can do almost anything, we won't be able to stand out by simply**

doing what AI can already do.

**We need to do what AI can't do. Think outside the box. Innovate by identifying efficiency and effectiveness bottlenecks in existing solutions and finding better ways to solve them.**

**This could mean:**

- **Revolutionising an existing UI component and turning it into something entirely new.**
- **Inventing a completely new UI component.**
- **Creating a totally new functionality that existing solutions or competitors don't have, can't replicate easily, or haven't thought of yet.**

**We're looking for people who can challenge assumptions, rethink how things are done, and create experiences that genuinely differentiate our products from everything else on the market.**

## 7 - Frontend & Backend Development, Security, DevOps & Code Cleanliness

You should be able to adapt with new frameworks, languages and other technologies fairly quickly. The skill alone outweighs how good someone already is at this or that.

Here are some of the things we work with on a daily basis:

- Node JS and all pre-requisite languages (HTML, CSS, Javascript)
- Reactjs alongside Nextjs
- Django (rarely these days)
- C++ and other languages which will aid in the development of future projects.
- Figma
- Webflow
- PHP & Laravel
- Ruby on Rails (Rarely).

## 8 - Keeping Organised & Easy To Work With

**This solves the problem of:**

- Right now it's a bit messy. I have multiple google docs to store project info, seperate monday boards for recruitment, team management, development, feature tickets etc...
- Everything is all over the place which makes it very, very difficult to manage. It would be much easier to have everything in one place and accessible.

**This means:**

A CRITICAL skill already mentioned above that i need to emphasize is the ability to organise staff, operations, workflows etc.... systematically and neatly.

Everything and everyone you are responsible for needs to be Organised to the point where we can actually deal with what needs to be dealt with when needed.

## 9 - Identify and solve bottlenecks

**This solves the problem of:** \*self explanatory

**This means:**

Bottlenecks with every single system in the business as well as those which arise in the user flow of our products. The leader should be able to identify these in advance, and either remove them completely or find a way to work around and/or minimize them.

## 10 - thinking outside of the box

**This solves the problem of:**

**This means:**

How can we make our product better? How can we improve the efficiency of our team? How can we hire qualified devs faster?

## 11 - Guidance - **MOST IMPORTANT**

**This solves the problem of:**

- My lack of competency in all the skills listed above

**This means:**

For each and every single skill listed above, you need to exceed my current level in that particular skill and impress me with whatever you do, from hiring to brainstorming efficient backend functionality, etc.

Put it this way, I should rarely be able to suggest a better way of doing any CTO / Tech lead / product head - level skill.

I'm hiring a CTO / Product head because you're supposed to be better than me at this. That's the whole point.

If I'm the one spotting problems and telling you what to fix, then my limited knowledge becomes the ceiling for the product's quality. That defeats the entire purpose of hiring an 'expert'.

## 12 - Doesn't Take No For An Answer

Instead of saying "This can't be done", instead, say "how can we innovate our own solution to solve this problem?"

# Unsatisfactory Performance Examples

For the sake of example, here are some past instances I've had with senior devs and tech lead positions who have worked with us in the past.

Many of the examples below involve lower-level roles, such as designers and junior developers, but the same principles apply to tech leads, as they are responsible for everything those roles handle.

## **Instance 1 - Failure to recognise an unsalvageable codebase**

A previous tech lead had spent over 5 days refactoring the frontend of a project we acquired from some random offshore dev.

I became frustrated by the lack of progress after all this time. So I decided to look into the repo one night.

Upon a 5 minute inspection into the codebase, I quickly identified that it would be significantly faster to restart from scratch, rather than try to refactor every component one by one.

The 2 things i don't like here are:

1. The lack of experience in the tech lead for him to not be able to realise that the project was not salvageable
2. Even if it was salvageable, he should not have done the refactoring himself. As tech lead, he should rarely be touching code, if ever. Leave this to the developers themselves.

## **Instance 2 - having to explain how to hire developers**

I should not have to explain how to effectively source and hire developers. There is no single right way to do it, but there is definitely a wrong way - and that wrong way is doing what every other tech company is doing to hire their developers. Especially if we are trying to get things done 10x faster and better than others.

As mentioned clearly above, I am offering a generous pay rate for this position because I specifically require someone who can guide me, rather than me guiding them. While my current methods for various tech lead tasks may already be effective, they are not effective enough.

## **Instance 3 - Stupidity and intentionally trying to minimise innovation in our product**

This team member was actually a backend developer who worked for me in June of 2025. I think every single team member should at least have some passion / vision for what we develop and they should be innovative - not just the tech lead(s).

This particular developer repeatedly told me that the product we were building at the time was not like a very well-known competitor, implying that our solution would not be able to compete.

#### **Instance 4 - Unable to think from the perspective of the user**

- Shit features
- Same as everything else
- No innovation from existing solutions
- Nearly identical to competitors without much difference
- Issues that hinder user flow - doesn't matter if competing products have the same issues.

#### **Instance 5 - Pure stupidity with UI/UX**

This happens a lot with fiverr freelancers. There have been numerous instances where the designer / frontend dev will either design something from scratch or redesign something and when you ask them what it's for, how it works or why they designed it that way, they come up with some stupid excuse like 'I don't know' or 'you asked it to be like that' or 'it was in the original design so i just improved it' or anything along those lines.

The worst one yet: I once asked a designer to explain what the functionality of a particular component was in the UI. Instead of explaining it, he just read out the copy of that particular component. Even after repeatedly asking him, he was unable to explain to me what the component even was.

I can't even believe i have to explain this, but if you're a designer or frontend dev and you don't understand how a component works, what it's for or why it's designed like that, then how the *FUCK* is the user supposed to know?

#### **Instance 5 - AI Slop**

I can't count the number of times something like this has happened.

I may require a PRD, prototype, MVP or some type of lightweight task, and they just vibe code some junk by copying and pasting what i asked for into Claude code, emergent or

Absolutely unacceptable. I should not have to explain why.

#### **Instance 6 - surface-level thinking**

There can be many 'correct' solutions to a problem, and the correct answer is never what comes to mind first.

## Ideal Candidate information

Ideal candidate past experience:

*(Each is just optional. Not expecting someone to have all these skills as some contradict others).*

- Previous CTO, senior dev or upper management for a large-scale software company and/or development agency who has plenty of experience building a diverse range of custom software.
- Ex-startup founder who knows how to build and manage multiple operations
- Solo dev who has hiring skills and can keep organised.
- Head of R&D / Innovation Lead
- Scale-Up Specialist (An engineer who has previously joined a company at the Series A–C stage and scaled the engineering org from a handful to dozens/hundreds of devs)
- Preferably someone who does not have a degree or formal qualification - as not having a degree would have forced you to think creatively to solve problems.

**The ideal candidate must be willing to initiate work by taking on part time / contract based work, and should be open to full time work in the near future (strong preference).**

Ideal Candidate current situation:

**Please read, this is important.**

The candidate must be willing to work per contract initially (KPI based), for the first few weeks. After growth picks up, the role will convert to a full time position.

Based on the condition above, the ideal candidate for this role could possibly be someone who has the required skillset and is currently unemployed, or is employed but able to commit additional hours to this role.

Ideal Candidate Examples:

## Pay:

Depending on which project you are assigned & whether you are qualified for a fractional or permanent CTO position

- **Single project build** - will be dependent on the complexity of the project requirements. This will be discussed before starting.
- **Fractional part time CTO - Starting at \$190k AUD / year**
- **Full time dedicated CTO - \$575k AUD / year** with increases as time progresses (full time CTO, permanent. Must meet tier 3 requirements as mentioned above)

Note: If you are applying for the Tier 6 SDM position, the payment structure is different. Instead of receiving a salary, you will be paid a fixed monthly fee for each software product you build and maintain. This payment will continue indefinitely. The monthly budget for each individual project will be agreed upon in advance, based on specific performance KPIs.

**Location:**

USA / AUS based is strongly preferred.

**Problems faced in prior hiring experience:**

When trying to recruit technical positions in the past, we faced just some the following issues:

Problem	Instance	Solution / Change Made
<p>We rapidly hired a team of 6 - 7 backend and frontend developers to build a project for us.</p> <p>Hiring was rushed with no clear system to properly filter out under-qualified candidates.</p> <p>Candidates were from Africa and India.</p> <p>We failed to fire and replace our team even after severe underperformance, because we thought we could teach them over-time. This was not the case, and even if it was, why would we waste time and money teaching them skills they should have already known?</p> <p>\$50k AUD loss and 4+ months wasted.</p>	<p>May 2025 - S***** dev team</p>	<ul style="list-style-type: none"> <li>- Come up with a multi-step hiring system to filter our rubbish candidates.</li> <li>- Avoid working with Indians and African developers.</li> <li>- Speed up the hiring process, but don't skip important safety measures</li> <li>- Quality check their past work thoroughly</li> </ul>
<p>We hired a reputable senior backend / cloud engineer who was already employed at a reputable Australian bank.</p> <p>The candidate was a family member of one of our company directors at the time.</p> <p>The candidate who was a senior dev, had no CTO-level skills whatsoever as he previously only worked as a devops engineer. We were willing to take the risk and bring him onboard anyways as he</p>	<p>Aug 2025 L**** CTO</p>	

<p>had experience building a small startup (chatgpt wrapper) once in the past.</p> <p>The candidate caused a massive stagnation in the development of one of our projects. This severely hindered the progression and the product never launched.</p>		
<p>A reputable Australian recruitment agency sourced a few candidates for us - for the CTO position.</p> <p>The candidates had not worked on projects that were even remotely similar to the projects we had developed in the past, and planned on developing in the future.</p> <p>We chose not to proceed with the candidates, as they wanted high pay immediately with no proof that they could do most of the skills we needed them to do.</p>	<p>Jan 2026 - Agency recruited CTO</p>	<p>No leniency. No second chances.</p> <p>Do not entertain those who have not proven that they have the essential skills of this role.</p> <p>We are here to hire someone who can guide us. Not us to guide them.</p>
<p>Poor quality testing criteria.</p> <p>We gave the opportunity to a few devs who had</p> <p>We were getting lower quality candidates anyways, so these guys's test results stood out, even though they were subpar.</p>	<p>Jan 2026 - Microtask Team</p>	
<p>Plenty of technical ML knowledge but 0 product vision.</p> <p>Lacked knowledge about hiring, team management and user perspective.</p> <p>Had a completely different vision to mine.</p>	<p>April 2026 - Temp. CTO for ML project</p>	

## Hiring Stages

The process we use to filter out sh\*t candidates from the rest, for literally any position in the company from frontend developer to CEO.

### [1 - Past Experience Check](#)

Skills tested (specific to this position)

- Vdav

## 2 - Initial Questioning - IMPORTANT

The most effective filter in this system. This is the same as the questions doc where they will need to identify bottlenecks and come up with solutions for them. Identify where past workers have gone wrong and filter them out here.

- Critical thinking skills
- Innovation

## 3 - Competency Test / Pilot Project If Contractor

Actions speak louder than words.

## 4 - Hired

The candidate is now hired on a full time or part time employment contract.

# Sourcing Plan

# Candidate Sourcing Plan

Normal hiring methods like job boards or LinkedIn may not work for this role because the required skills are very rare and hard to find.

## Criteria:

We need to source those who have the following experience ([as described on the first page](#)).

- project management
- Team management
- Hiring
- Creative project vision
- Full stack development
- UI / UX design for web apps

## Where can we find such people:

Not job recruitment boards etc...

Instead, we will search:

- Manually pick and choose from reliable companies
- Reddit subreddits
- Product hunt

## Hiring Idea 1:

Scout and shortlist new single-founder startups on Product hunt, Reddit and organically. Startups which are well-built. Does not need to have many users, if they have any. Focus on quality.

Ask them via email if they are interested in a CTO position.

## Hiring Idea 2:

Connections and referrals. Ask existing connections.....



## 0 Past Experience Check Pt0

# Past Experience Check - IMPORTANT

The candidate's past experience should clearly demonstrate that they can ship software with:

1. Lack of obvious bugs / user flow bottlenecks
2. Clean, tidy codebase and management system. Nicely organised + plenty of comments left to make it easier to work with when onboarding new dev team members.
3. Creative / original design + more user friendly than alternatives
4. Creative thinking: unique functionality and features to improve UX - ("We added xyz to accommodate for xyz problem which users were facing")
5. Creative thinking to address challenging scenarios
6. Tight deadlines
7. Software projects with complex backends that can't just be vibe coded and advanced functionality (not simple ChatGPT wrappers and other basic beginner level projects)
8. Similar projects to [what we have made in the past](#) (optional but preferred)
9. INNOVATION - most important skill (read below for more info)

The candidate's previous work should clearly demonstrate that they can think creatively to:

- Recruit / headhunt devs
- Manage a team
- Manage projects with tight deadlines
- Address security concerns
- Keep extremely organised

## Examples of what i like to see

### Product functionality & Design

- <https://coolors.co/>
- <https://www.memberstack.com/>
- <https://www.watermarkremover.io/>
- <https://www.productioncrate.com/>
- <https://linear.app/>
- <https://gptzero.me/>
- <https://dribbble.com/shots/26714537-E-commerce-Buying-Selling-of-goods>
- <https://dribbble.com/shots/15521680-UI-Design-For-Artwork-s-Gallery-3D-Models>
- [https://www.behance.net/search/projects/web%20app%20concept?tracking\\_source=typahead\\_search\\_direct](https://www.behance.net/search/projects/web%20app%20concept?tracking_source=typahead_search_direct)
- <https://dribbble.com/shots/25882999-Expandable-Notification-Toast-with-timer-for-SaaS-App-Dashboard>

Other specific CTO / product head behaviour to look for in their resume / past experience check:

- Well-written PRD (project requirements) like this one: <https://keytasker-docs.vercel.app/>
  - Neat and organised. Well presented
  - Easy to understand
  - Outlines bottlenecks
- Well written hiring flows like this: <https://frontend-engineer-eval.vercel.app/>

## Examples of what i don't like to see:

Vibe coded junk, AI slop, poor UI, bugs, lack of functionality, lack of creativity, boring, generic ideas, not even MVP worthy etc.....

- [www.junopm.co](http://www.junopm.co)
- <https://www.autoapplier.com/>
- <https://searchablepdf.org/>
- <https://kaleab-ayenew.github.io/chatvas/>
- <https://apps.apple.com/sg/app/spotvault/id6758209904>

## Innovation

- Identifying issues with existing solutions
- Creating something new
- Never copying others
- Thinking outside of the box
- Never seen before ideas

Read more here -

[https://docs.google.com/document/d/1\\_IG3TC79nFwy9FcVbbI\\_lo7vwwK1HVI2vW4fLZ4ZxtY/edit?tab=t.wiy347a8ggkl](https://docs.google.com/document/d/1_IG3TC79nFwy9FcVbbI_lo7vwwK1HVI2vW4fLZ4ZxtY/edit?tab=t.wiy347a8ggkl)

# 1 Initial Questions - Skills Testing Pt1

# Interview / Questionnaire (*Simple Questions*)

***These questions are all optional, but it would be ideal if candidates can complete as many of these questions as they can - this way, we can get a better understanding of each candidate's thought process.***

*I have written some **brief** sample answers in grey italic text. These responses are incomplete and only include a short summary of my answers, with non-critical information removed.*

## Team management, project management & Recruiting

1. Outline an efficient and reliable method to hire a competent UI designer within an unrealistic 12 hour timeframe.
  - *Reddit - I will make a post to niche reddit design subs, buy upvotes to the post and ask the candidates to send us past experience via email. Then it's just pick and choose from there.*
  - *Fiverr - though fiverr is notorious for only having the bottom 10% of workers in any profession, there appears to be some good designers on the platform.*
  - *Manual outreach on dribbble*
  - *Job posting + ads*
2. What signals do you look for to identify top 1% developers quickly?
3. Assume we hire a freelancer or contractor to build some APIs for a project. If their work falls well below our quality standards, we may choose to terminate the agreement and attempt to recover some of the costs. However, even if some money is recovered, the time lost and project delays caused by the underperformance cannot be recovered.
  - a) Assume we are limited to only relying on freelancers for whatever reason. How can we prevent something like this from happening again when working with freelance developers?
  - b) How do we minimise the repercussions from the situation at hand (assuming we have not decided on anything yet with the freelancer).
    - a) *We may need to use freelancers for quick, one-off tasks, especially when we are working under a tight deadline and cannot hire a reliable long-term developer fast enough.*

*First, we will do a quick portfolio review to assess the freelancer's quality. If the task we need completed closely matches the type of work they have done before, we will be more likely to hire them.*

*Next, we will identify five freelancers whose experience fits our requirements.*

*Using Fiverr's built-in escrow system, we will place the same order with all five freelancers for the exact same task. Clear expectations and strict deadlines will be provided upfront, like [this](#).*

*When the deadline arrives, we will review all submissions, select the best result, and cancel the remaining orders.*

- b) *Since the party is a freelancer, we will simply use a pre-defined criteria to determine whether the work is salvageable or not. If not, we can simply force cancel the order via fiverr support.*

*In situations like this, it may not be worth salvaging the code.*

4. You are managing a team of 10 frontend developers, 10 full stack developers and 2 UI designers. What preventative measures do you enforce to prevent:
  - a) Lack of communication, resulting in messy PRs and overwritten code
  - b) Slow progression
  
5. Running out of developers when we need them most can have serious consequences. How can we ensure we always have a reliable team of developers available at our fingertips?

*First and foremost, ensure each team member has a real, genuine reason to work for us rather than for others. Retaining our team should be the top priority. Things like making sure each team member has a real passion for what they do and for what we're building can go a long way.*

*When we do need to expand the team, we will ensure that we are ALWAYS hiring no matter what. This way, we will always have a surplus of candidates.*

*The best way to approach this is by creating a separate page on our most popular products, or on products that are likely to have users with relevant technical experience.*

*On this page, we will advertise job openings.*

*Here is one example that has worked for us in the past: <https://imgur.com/a/liY0kIW>*

*Users can click on any job they are interested in, which will take them to a portal like this. <https://frontend-engineer-eval.vercel.app/>*

*The hiring process is very streamlined. We don't conduct tedious interviews with every single candidate.*

*Instead, we only review the test results from the top candidates. From that group, we will contact those who appear to be a good fit to work with us, based on their test results.*

*If the job portal page isn't getting many clicks, we can draw more attention to it by adding links to it on relevant pages of the site, running ads to the page, etc.*

1. Give an example of how you've 10x'd development speed on a project.
2. Choose any web / mobile application that you have used in the past. List every single UI flaw that can be improved with it.
3. Choose any web / mobile application that has been a top product on producthunt in the past. List every single UI flaw that can be improved with it.

## Design

1. [This](#) is a software designed to rewrite the input text to create a new, natural-sounding piece of content based on up to three text extracts.
  - It combines the key information from the inputs, restructures the sentences so there are no recognizable patterns from the originals, and produces a human-like output that retains the same meaning while bypassing AI and plagiarism detection tools.
  - The system can also mimic the writing style or structure of one selected extract, allowing users to apply a specific tone or layout to the final result.

### **List every flaw with the UI.**

- *Users are not sure whether this product is a good fit for them. Addressing this alone can increase revenue significantly.*
  - *Confusing UI. Based on this UI, it seems like the user must add at least 3 extracts for it to work*
  - *The UI is vibe coded with little to no change. They literally copied the HTML and CSS straight out of the AI coding tool and pasted it straight into VS code with no manual changes.*
2. What's wrong with [this UI](#). List the top 5 most obvious flaws + describe how you'd improve them.

*There are so many to list.*

*First of all, the UI is so outdated and unresponsive. Why is everything centred so much? Even on smaller breakpoints?*

*The layout is terrible and visual indicators are not the best.*

3. Here is a screenshot of Oxylab's dashboard UI - <https://imgur.com/a/wOQRC9h>  
How can it be improved?

*Oxylabs have failed to acknowledge that while a majority of their users are technical, some users are non-technical.*

*If I were in charge of Oxylabs, the first thing I would do is cater this interface more towards the non-technical users first.*

- *On the homepage, we will add a clear, prominent section to list all the use cases of Oxylabs products. Draw attention to all non-technical use cases as well, because many non-technical users are considering alternatives simply due to not knowing whether oxylabs is for them.*
- *Questionnaire - upon signing up, we will ask users if they know how to operate their proxy, or if they are new to this. If they are new, we will include a short, well produced tutorial video. To simplify it even further, we can also 'grey out' certain unnecessary functions which are not necessary until the users get more experienced.*
- *Simplified interface - The rule of thumb that we follow with every single product we design, is assume the user is extremely unintelligent. Assume the user is an idiot. That being said, the design needs to be simplified so much so to the point where anyone could use it - even a 5 year old.*
- *Besides the 'residential proxies' in the 'my products' area, we will also add a div block which the user can click to simply add other products.*

4. How can we ensure our UI/UX is always user-friendly? By user-friendly, we mean interfaces that are easy to navigate, easy to understand, and allow users to achieve their goals quickly without any bottlenecks.

*We should always be redesigning and improving, even when there is no obvious problem.*

*To do this, we will always have designers working on redesigns. Not complete redesigns, but continuous enhancements and improvements. We will develop a system that allows us to quickly identify user complaints, either through user behaviour or through complaints explicitly submitted by users. Constant testing will also be part of this system.*

*The same concept can also be applied to frontend development, backend development, security measures, and other areas.*

*If we are always improving UIs and A/B testing designs instead of waiting for problems to appear or for users to report complaints, we will be able to move forward much faster.*

5. Why is the design and layout of [Freelancer.com](https://www.freelancer.com) not optimized for UX?  
<https://imgur.com/a/HfPAnbh>
6. Do you approve / disapprove of the UI / UX and current feature set of [petrol spy?](#)
7. Why is Spotify an example of poor UI?  
To answer this question, you must first identify the user's intention. What is the user trying to do? Based on that, what functionality and/or UI element is missing from the interface that makes it difficult for the user to reach their aim? That being said, what functionality is missing from this product that makes it feel incomplete for the user?
  - *Outdated*
  - *Most of the time, users don't know what to listen to. Yes, they introduced DJ, but DJ is still not good most of the time*
  - *No music discoverability apart from discovering mainstream music or songs from small artists that sound horrible.*

## Genuine Product Vision & Functionality / Optimised User Flow

1. What were some challenges you have faced in your past roles, and how did you use innovative thinking to overcome these challenges.
1. Think from the perspective of a real [Soundcloud.com](https://www.soundcloud.com) user. Identify two issues that slow down the user flow, and suggest improvements. These improvements can either be changes to existing features or new features designed to fix the bottlenecks you identified.

*Example answer - Firstly the design is horrendous. It's outdated, not 'functionality first' - the homepage is designed like a landing page which reduces conversions.*

*The pages are all designed like every other music streaming app available. There is no innovation here, other than the fact that soundcloud lets users upload directly to the platform.*
2. Describe just one functionality change that we could add to google docs to make it more user friendly. Think about a common problem many users have and how we could address that?

*Example answer - Currently there is no way to add quotation marks in just one click. So it would be nice to have a button or function where we could highlight text and automatically put it in quotation marks and italics.*

*Another cool feature would be shortcuts. Like mentioned above, what if we could map out keys that automatically format text in some way based on keyboard presets.  
Another cool feature would be integrating docs with the OpenAI API to help write faster somehow.*

3. Here is a [screenshot of a playlist from Spotify](#). It's quite standard, and variations of this same layout can be seen across many other music streaming apps. List 5 things about this layout and design that is frustrating the user, or limiting their experience on Spotify.
4. Totally hypothetical - Imagine we want to use the OpenAI API to make a free, simple, online tool for reddit users. The fundamental aim of this project is to develop a free online web application that in some way, affects (changes) what reddit users were intending to post to reddit (text based posts only).

A simple ChatGPT wrapper won't cut it here. Based on the aim, how can we create a product that is truly unique and useful and will stand out from the rest?

*Here is my idea summarised: Paste your post → tell the tool what you're posting → get back a version optimized to blow up in that subreddit.*

*This idea works because it solves a core problem that reddit users have. What is one thing all reddit posters want? Virality / to be seen, right? What's the solution to that problem? A virality optimizer.*

*To summarise my idea, here are the key points of my web app idea:*

- *The tool rewrites their post to maximize its chances of going viral in that specific subreddit*
- *Once the response is generated, there will be toggle switches which the user can control to remove certain edits which the rewriter has made (see bullet list below)*
- *No loading page, since this is a free tool, the homepage will be the functional dashboard*
- *Put 2 - 3 ads to monetise*

*The tool will help the user's post go viral by:*

- *The response will be tailored to match the tone of viral posts in the subreddit they intended to post to*
- *The response will sound like it was human written, by the user. There won't be any major changes to the original input.*
- *Suggests flair and title to maximise virality potential*
- *Suggests subreddit to post to (if they don't already know)*
- *Tailors post to match subreddit rules and formats*

- *Tells best time to post for that subreddit*
- *Fixes big walls of text*
- *Adds spacing / formatting*
- *Uses bullet points when needed*
- *Reduces density / wordiness*
- *Makes it easier to follow on mobile*
- *Fix intro length (too long / too short)*
- *Fix wrong tone (too formal / too casual)*
- *Fix buried lead*
- *Fix wrong post style (story vs question vs guide)*
- *Improve title if not engaging*
- *Seamlessly injects something funny, interesting, shocking, or controversial to boost engagement*

5. Totally hypothetical - imagine we are building a competitor to ChatGPT. Focus only on the API aspect for this question. What are some unique, problem solving ideas for a new text-generation API similar to Open AI's API?

*To answer this question, we will think from the perspective of the developer who may want to build a product using our API or integrate it into their existing product*

*Here's some features we will add to make our API more appealing than OpenAI's*

- *Tiered pricing = we will offer significantly cheaper discounts for customers with high API consumption. This alone could be our competitive advantage*
- *Our API will bypass AI detection by frequent re-training*
- *Adapt the user's writing style*
- *To be honest, we could literally use the OpenAI API to train our custom model, reap it up again and sell it as a different API for text generation.*

6. User intent is the core goal behind how people use a product, but most platforms fail to provide infrastructure that actually supports what users naturally try to do. Instead, users hack together workarounds (like subreddit formats, reaction trends, or "wanted" listings), and platforms only sometimes catch up.

The opportunity is to build software where the infrastructure is designed around real user behavior from the start—making it easy for users to do what they already want to do, and continuously evolving based on how they use it. This allows us to exploit UI/UX gaps in existing platforms and create more flexible, user-driven systems.

1. Choose a web or mobile app you use regularly (daily or weekly). Functionality-wise, explain what feels missing or incomplete about it, and how you would improve it.

## System Design / Architecture

1. Assume we run a free online dictionary where the key feature is that every definition is written so simply that even a five-year-old could understand it. Explain an efficient method for adding large numbers of new words and clear, high-quality definitions to the site. The process should also be designed to help these pages rank well on Google.

*We will split this operation into 4 segments: Keyword research → Definition writing → indexing*

*Keyword research - We will use semrush to find keywords like*

*Given that the purpose of this dictionary website is to write simplified definitions of supposedly complex words, we will also ask chatgpt for a complete list of words that are hard to explain. These will be good for our site.*

2. Our microtask website works, but we have many bottlenecks. Come up with a creative way to solve this bottleneck

*We will split this operation into 4 segments: Keyword research → Definition writing → indexing*

*Keyword research - We will use semrush to find keywords like*

*Given that the purpose of this dictionary website is to write simplified definitions of supposedly complex words, we will also ask chatgpt for a complete list of words that are hard to explain. These will be good for our site.*

3. Give an example of how you've 10x'd development speed on a project.

4. How would you bake growth into the product itself instead of relying on ads?

- Viral / problem solving
- Referral
- reliance

## Frontend & Backend Development

- 1.

## General thought process

The following questions can be applied to all different segments of CTO-level tasks.

1. Instinct is basically compressed survival knowledge from ancestors
  - Individuals with useful automatic behaviors survived more
  - Those behaviors got passed down genetically
  - Over millions of years → they became default behaviors

The thing i like most about instinct and ‘

The common denominator here is: automatically change something depending on something.

If you

How can this same concept be applied to developing software that adapts to whatever the users choose to do with it?

1. Instead of building new features to adapt to users, there needs to be something automated.

## Quick questions (July 2025)

Problem solving

Share past experience, if not, make

Fast learning

Design -

problem solving -

User acquisition -

Out of the box thinking -

hiring -

managin multipl project -

## 2 Competency Test - Skills Testing Pt2

# (Paid) Competency Test + Well-Written PRD Doc

We will test the candidate with a paid competency test.

This competency test will require the candidate to handle start to finish development of a project we provide to them.

With this test, we can test some of crucial the skills of the candidate such as:

- Creative thinking
- Product vision
- Innovation
- Technical abilities
- [etc....](#)

Once the candidate reaches this stage, we need to initiate a video call with them.

In the call, we need to explain what this test is for. Provide them with [this contract](#) for them to read and sign.

## **After the call provide the following information to the candidate:**

1. project aim
2. Sample PRD
3. contract (share again):  
<https://docs.google.com/document/d/1nq7eYwvnqs8eUFb-TzD0zPyJiyiClmWr/edit?usp=sharing&oid=113740647537304075652&rtpof=true&sd=true>
4. due date. The due date for any project will be 2 weeks from the starting date. Ensure the candidate provides milestone updates along the way.
5. budget which we will pay them in milestones (see contract)

Here is an example message which you can send to the candidate to initiate the competency test. This message includes all the relevant info they need to know. The contract includes more critical info.

## Quality expectations.

The candidates output in each milestone should reach or exceed the quality expectations described in the contract -

<https://docs.google.com/document/d/1nq7eYwvnqs8eUFb-TzD0zPyJiyiClmWr/edit?usp=sharing&oid=113740647537304075652&rtpof=true&sd=true>

**To summarise:**

- Innovate solution. The product must be designed differently to competitors (if they exist).  
Never seen before UI / layout and unique user flows.
- Real UI, not vibe coded
- Unique feature set and creative thinking used to come up with the features.

## 3 Probation Period - Skills Testing Pt3

# Hired!

## Leniency:

### Up to 4 major warnings -

- Obvious bugs and efficiency bottlenecks resulting in poor UX
- Unoriginal design / features as a result of lack of creativity and innovation
- I am repeatedly able to come up with a better solution to different solutions
- Lack of knowledge that should be known / Requires guidance
- Unable to learn quickly
- Lack of creativity and innovation in any form ([see more here - scroll near midsection of page](#)). Does things by the books and goes with the flow.
- Poor management of github repos resulting in issues such as overriding PRs, messy code, lack of communications between devs etc...

# Onboarding

# Onboarding

## **For existing project:**

1. Access to all Github repos
2. hosting providers etc...
3. + meeting with the existing team.
4. GET STARTED!

## **For new project:**

1. Provide PRD
2. Forget everything other than the aim
3. *Plan*
4. Build!