**MOCK INTERVIEW QUESTIONS**

1. Tell me about yourself.
2. What interested you about this position?
3. Review your work history,

* 1) How/why you got each job,  
   2) Favorite part of each job,  
   3) Why you left each job

1. Flash forward to a year from now, if you were in this position, what would we have achieved and be celebrating?



1. Tell me about a work/career project or accomplishment that you are particularly proud of.



1. Share with me a specific example of when miscommunication caused an error. What happened?

* OR

Talk about team conflict and how you resolved/handled.

1. Tell us about how you failed or underdelivered and how you responded?



1. Tell me how you are most misunderstood.



1. Walk me through the first five things you would do if you got this job.



1. What was the most difficult part of working on your Final Group Project?
2. What do you look for and what are some important things to consider when looking over a teammate’s code?



1. How would you approach learning a new framework or language?



1. Do you have examples of projects that you worked on with vague requirements? How did you go about working through that challenge?



1. Do you consider yourself a frontend or backend developer? And Why did you decide to pursue it (one of them)?



1. **PROJECT/Demo/Presentation (Links?):**

* Why did you use this particular Language/Framework (Angular, React, Ionic, etc.) for this project  
  you built?

1. Why did you decide to study this (JS/Java/C#...) Tech Stack?



1. Are you familiar with our product? If so, are there any improvements or features that you would consider adding? If not, do you have suggestions for a product/project that you’ve worked on previously?

* [**Question by Product Manager**, trying to understand candidate’s product decision sense & what type of solutions they offer if any at all]
* 17. Can you give me an overview of the architecture of a system you’ve worked on and what technologies you chose to build it and why?   
    
  (if the interviewer isn’t technical, try to rephrase the question as an overview to understand what the candidate has worked on and see if they can communicate complexity in layman’s terms to a non-technical teammate)  
    
  [**Question by Engineer or EM**, similar to **#15]**

18. When you have multiple tasks but you don’t have time to work on all of them, how do you decide what to work on and prioritize?

## **Additional Questions:**

## Background

### Tell me about yourself.

### Can you provide a high-level summary of what you learned in your coding bootcamp?

### Tell me about a project you’ve created that you’re proud of

### What about Software Development excites you?

## Problem Solving/Teamwork

### Think of a recent program you have worked on. Describe a technical problem you’ve faced, and what you did to come up with a solution

### Describe a time you faced conflict with a teammate or a colleague, and what you did to work through it

### What would you do in a scenario where you feel like a team leader (or teammate) is making a wrong technology decision?

### Do you have any experience using SCRUM? What did that look like?

### Why is it important to have regular stakeholder demos and team retrospectives?

**Questions to ask the employer:**

* **What’s your favorite thing about the company culture? (NOT - What’s the company culture like? - bc then it sounds like you haven’t researched)**
* **How does the company intentionally allow personal and professional growth?**
* **(Ask questions that are valuable to you in a workplace)**
* **What do you expect for someone hired on within the first 30-90 days?**
* What do YOU see as the biggest opportunities for this company and also the biggest risks?
* Will it be around to stay?
* When you were interviewing here, what would you have liked to know before you joined?
* What is the biggest problem I can solve in my first six months?
* What were some qualities the previous employee in this role brought to the job that you would like to see carried forward?
* If I asked your direct reports about your management style, what do you think they’d tell me?
* What are some possible career trajectories within the company that could stem from this position?
* Why is this position open?
* How Often Do You Work Evenings and Weekends?
* What will be my responsibilities for the next 6-12 months?
* What’s the Best and Worst Thing About Working Here?
* How are decisions made? Is architecture dictated top down? Are ideas from anyone welcomed? If so, in what scope/context?
* Is there room for growth?
* How Have You Supported Developer Professional Development in the Past?
* Ask Them to Describe Their Software Development Lifecycle (How Often Do They Release Code?)
* Ask About the Company’s Long Term Vision (What Are Your Plans for the Next Five Years?)
* What’s the Average Turnover Rate?
* How will my performance be measured?
* Do you have any feedback for me?
* Why do you enjoy working here?
* What’s a Typical Day Like in This Position?
* How Have You Supported Developer Professional Development in the Past?
* Ask for a Tour Around the Office at the End
* Ask Them to Describe Their Software Development Lifecycle (How Often Do They Release Code?)
* Ask About the Company’s Long Term Vision (What Are Your Plans for the Next Five Years?)
* What does Company Culture look like & how is it different then other companies?
* What's the growth plan for me?
* Tell me about the other teammates that I will join?
* What are the biggest challenges facing the team right now?
* How do your customers benefit from using your product or service?
* Why are you hiring for this position? or Why is this job open?
* What have past employees done to succeed in this position?
* Does this Job require travels? How often?

**Ask Other Good Questions** like:

* **What’s your favorite thing about the company culture? (NOT - What’s the company culture like? - bc then it sounds like you haven’t researched)**
* **How does the company intentionally allow personal and professional growth?**
* **(Ask questions that are valuable to you in a workplace)**
* **What do you expect for someone hired within the first 30-90 days?**
* What’s your favorite thing about working with the company?
* Why did you choose to work here?
* If I were you, what concerns would you have about me?
* What do you like about the company values?
* Do you give back to the community?
* Other sources in asking good questions to a future employer:
  + <https://github.com/viraptor/reverse-interview/blob/master/README.md>
  + <https://github.com/lkostrowski/job-interview-questions-to-ask-companies>
  + <https://www.glassdoor.com/blog/common-interview-questions/>
* **Top 50 Questions asked in Interviews:**
  + <https://www.glassdoor.com/blog/common-interview-questions/>
* HR/Culture Fit Interviews:  
  <https://www.glassdoor.com/blog/common-interview-questions/><https://www.themuse.com/advice/interview-questions-and-answers>

**TECH INTERVIEW QUESTIONS**

**FULL STACK RESOURCES**

* [Leetcode.com](https://leetcode.com/) \*check this out!
* [HackerRank](https://www.hackerrank.com/domains/algorithms/implementation/difficulty:easy/page:1)
* [Codewars.com](https://www.codewars.com/)
* [Interviewcake.com](https://www.interviewcake.com/)
* [Algoexpert.io](https://www.algoexpert.io/product)
* [CoderByte](https://coderbyte.com/) \*check this out!
* [Free Code Camp](https://www.freecodecamp.org/)
* [Pramp](https://www.pramp.com/dashboard#/schedule)
* [Top Coder](https://www.topcoder.com/community/development/)
* [Amazon’s Recommended Software Dev Topics](https://www.amazon.jobs/en-gb/landing_pages/software-development-topics)
* [GeeksforGeeks](https://www.geeksforgeeks.org/top-25-interview-questions/)

Books:

* [Cracking the Interview](https://www.amazon.com/Cracking-Coding-Interview-Programming-Questions/dp/0984782850)
* [Programming Interviews Exposed: Secrets to Landing Your Next Job](https://www.amazon.com/Programming-Interviews-Exposed-Secrets-Landing/dp/1118261364)
* [Programming Pearls](https://www.amazon.com/Programming-Pearls-2nd-Jon-Bentley/dp/0201657880/ref=sr_1_1?dchild=1&keywords=programming+pearls&qid=1629405266&s=books&sr=1-1)
* [Introduction to Algorithms](https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844/ref=sr_1_1?dchild=1&keywords=Introduction+to+Algorithms&qid=1629405286&s=books&sr=1-1)
* [The Algorithm Design Manual](https://www.amazon.com/gp/product/1848000693/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1848000693&linkCode=as2&tag=thecodebytes-20&linkId=c999793349b1b9b4ad6591a006517ea4)

YouTube Videos List:

[Pramp](https://www.pramp.com/dashboard#/schedule) and [Cracking the Code](https://www.youtube.com/playlist?list=PLX6IKgS15Ue02WDPRCmYKuZicQHit9kFt)

Front-End:

<https://github.com/dexteryy/spellbook-of-modern-webdev>

<https://github.com/yangshun/front-end-interview-handbook>

React:

<https://github.com/sudheerj/reactjs-interview-questions>

System Design:

<https://gist.github.com/vasanthk/485d1c25737e8e72759f>

Algorithms:

<https://github.com/trekhleb/javascript-algorithms>

<https://github.com/TheAlgorithms/Python/blob/master/DIRECTORY.md>

Here's the powerpoint to follow along! <https://docs.google.com/presentation/d/1fKNEa_2XIpFtWWR4vDX5Kgt-dwbvoFbXP0sTSq_8taA/edit?usp=sharing>

When you're ready.... solutions: <https://docs.google.com/document/d/1BtGN_y5HgW0izm9AzYLNrcRzKAYsQ_lhifettASoOMk/edit?usp=sharing>

**DESIGN CHALLENGE**

**to demonstrate how the designer approaches problem-solving and identify their design process**.

<https://www.youtube.com/watch?v=ccQJZcTaRyE>

<https://www.youtube.com/watch?v=KPv1UnivSBo>

<https://www.youtube.com/watch?v=MpO5j2m6gS4>

<https://www.youtube.com/watch?v=3blyJn6htLU>

<https://www.uxbeginner.com/how-to-prepare-ux-interview-design-challenge/>

<https://blog.prototypr.io/design-challenge-in-5-hours-fd2080ae8db>

<https://uxplanet.org/a-concise-guide-to-onsite-design-challenges-7e67293390e1>

[https://sharpen.design](https://sharpen.design/)

* [DesignBuddies!](https://www.designbuddies.community/) - community support and resources (my favorite!)
* [Cofolios](https://cofolios.com/) - community support and resources
* [ADPList](https://www.adplist.org/) - community support and resources
* [MilaNote](https://milanote.com/) - tool for design practice
* [Sharpen](https://sharpen.design/) - design prompts
* [Designercize](https://designercize.com/) - whiteboard practice
* [Prototype](https://www.youtube.com/watch?v=8MQmYa_zq0g) - whiteboard walkthrough

**Ramsey Solution** - Design Challenge - see what other candidates have done

* Capstone project (at Bethel Tech) helped with the Design Challenge
* Battle of the bands design challenge
* Product manager was impressed - point was to show your thought process - want to get your point across quickly (you want to stand out - come across your solution in first 4 minutes)
* Visual component - make sure you state how many hours you spent on this project (instead of stating it, she did a visual chart on how much she spent on visual architecture, prototype, wireframe, etc.)
  + Wireframe - get it onto 1 page
  + Get the Branding and colors as similar to your challenge result as much as possible
  + Can google “Battle of the bands” Ramsey Design Challenge as an example

**Resume Website Portfolio example** - https://www.nialee.com/work-2

**DATA SCIENCE INTERVIEW RESEARCH**

Can expect: Portfolio presentation and/or Whiteboard challenge

* [MLPro](https://mlpro.io/) - Practice problems
* [Data Engineering Cookbook](https://github.com/andkret/Cookbook) - practice problems
* [Algoexpert.io](https://www.algoexpert.io/product) - Practice problems and resources
* [TowardsDataScience](https://towardsdatascience.com/how-to-prepare-for-your-data-engineering-interview-d245519da45c) - How to Prepare \*great read!\*
* [TowardsDataScience](https://towardsdatascience.com/over-100-data-scientist-interview-questions-and-answers-c5a66186769a) - additional sample questions
* [365DataScience](https://365datascience.com/career-advice/job-interview-tips/data-science-interview-questions/) - resources
* [Kdnuggets](https://www.kdnuggets.com/2020/01/data-science-interview-study-guide.html) - overall study guide and resources
* [Leetcode](https://leetcode.com/) - stick to level easy questions

Additional Resources:

**TECHNICAL INTERVIEW PREP:**

*https://www.interviewcake.com/*

[*https://www.algoexpert.io/*](https://www.algoexpert.io/)

[*https://www.codewars.com/*](https://www.codewars.com/)

[*https://coderbyte.com/*](https://coderbyte.com/)

[*https://www.freecodecamp.org/*](https://www.freecodecamp.org/)

[*https://leetcode.com/*](https://leetcode.com/)

***Youtube & This Youtube playlist:***

*https://www.youtube.com/playlist?list=PLX6IKgS15Ue02WDPRCmYKuZicQHit9kFt*

[*https://www.pramp.com/dashboard#/schedule*](https://www.pramp.com/dashboard#/schedule)

# Algorithms & Data Structures

Students are required to learn about and practice the following concepts

* Array
* Binary
* Dynamic Programming
* Geometry
* Graph
* Hash Table
* Heap
* Interval
* Linked List
* Math
* Matrix
* OOP
* Permutation
* Queue
* Recursion
* Sorting and Searching
* Stack
* String
* Tree
* Trie

Ref: <https://yangshun.github.io/tech-interview-handbook/algorithms/algorithms-introduction>

These concepts will be given in the form of workshops and an assignment.

This training is 5 weeks long, and it mainly focuses on solving coding challenges.

Week 1 will cover Sequences and basic structures

Week 2 will cover Data Structures

Week 3 will cover Non-Linear Data Structures

Week 4 will cover Advanced Data Structures

Week 5 will cover Dynamic Programming

Reference

<https://yangshun.github.io/tech-interview-handbook/best-practice-questions>

## Practice Platforms

* <https://leetcode.com/> (Recommended)
  + Solutions <https://github.com/fishercoder1534/Leetcode>
* <https://binarysearch.com/>
* <https://www.hackerrank.com/>

## Resources

* <https://github.com/kdn251/interviews>
* <https://github.com/yangshun/lago>
* <https://github.com/jwasham/coding-interview-university>
* <https://www.hiredintech.com/>

**Other**

* <https://30secondsofinterviews.org/>
* <https://github.com/learning-zone/html-interview-questions>
* <https://github.com/learning-zone/java-interview-questions>
* <https://github.com/learning-zone/nodejs-interview-questions>
* <https://github.com/learning-zone/react-interview-questions>
* <https://github.com/learning-zone/php-interview-questions>
* <https://github.com/learning-zone/mongodb-interview-questions>
* <https://github.com/learning-zone/css-interview-questions>
* <https://github.com/learning-zone/spring-interview-questions>

# System Design and Architecture

* Network & Protocols (IP, DNS, HTTP, TCP…)
* Storage, Latency & Throughput
* Availability
* Caching
* Proxies
* Load Balancing
* Consistent Hashing
* Databases
* Leader Election
* Polling, Streaming, Sockets
* Endpoint Protection
* Messages & Pub-Sub
* Logging, Monitoring, Alerting

Resources:

* <https://github.com/donnemartin/system-design-primer>
* <https://www.freecodecamp.org/news/systems-design-for-interviews/>

Other resources

<https://www.hiredintech.com/classrooms/system-design/lesson/52>

<https://www.freecodecamp.org/news/systems-design-for-interviews/#section-3-system-availability>

# **Additional Mock Interview Questions:**

## **Background**

### Tell me a little about yourself

### Can you provide a high-level summary of what you learned in your coding bootcamp?

### Tell me about a project you’ve created that you’re proud of

### What about Software Development excites you?

## **Problem Solving/Teamwork**

### Think of a recent program you have worked on. Describe a technical problem you’ve faced, and what you did to come up with a solution

### Describe a time you faced conflict with a teammate or a colleague, and what you did to work through it

### What would you do in a scenario where you feel like a team leader (or teammate) is making a wrong technology decision?

### Do you have any experience using SCRUM? What did that look like?

### Why is it important to have regular stakeholder demos and team retrospectives?

## **OOP Principles \*\*\* Four Pillars of OOP**

### What is encapsulation? Why might you use different forms of encapsulation? (E.g. Angular)

### What is polymorphism? Can you give an example?

### What is the difference between value and reference data types?

### What is a Singleton and when might you use it?

## **JavaScript**

### Difference between var, let and const?

### Difference between “==” and “===”?

### Can you describe what a Lambda function is?

### What is a scenario when you might use a Lambda function?

### Can you explain what a JavaScript Promise is?

### Difference between JavaScript and TypeScript (“use strict”)? Benefits?

## **Angular**

### How does the \*ngFor directive work?

### What is the difference between 1-way and 2-way data binding?

### Can you reference styles from one component in another?

### Where would you write CSS in an Angular app if you wanted the styles to apply to multiple components?

### Describe some differences between Services and Components.

### Differences between Components and Directives?

### Describe the advantages of using Modules in Angular.

### What is a Route Guard?

### Differences between Reactive forms and Template-driven forms?

### What is an Observable? How is this different from a Promise?

## **React**

### What is JSX?

### How do you loop inside of JSX?

### Differences between a Class Component and a Functional Component?

### What is State in React? Is it accessible to other components?

### Are there ways to share state between multiple components?

### Why should you not update the state directly? (e.g. this.state = 123 vs. this.setState(123))

### How would you pass data into a Component from another component?

### Can you describe what the Virtual DOM is and how it works?

### How do you use styles in React?

### What is the React Router?

### How to programmatically navigate using React Router?

## **Server Application/API Development**

### What is the MVC Architecture?

### Can you describe how server-side rendering works in C#/Java/Node? (Looking for Razor/Thymeleaf/Handlebars)

### What type of data does a REST API return?

### Can you describe how a client application interacts with a server app?

### What are the primary responsibilities of a client app vs a server?

### What are the different HTTP Request Method types (GET, POST, PUT) and when do you use one vs the other?

### How would you go about organizing the routes in an API?

### What is an example of an API change that would break its contract with client apps?

### Can you explain what middleware is and why it's used?

### Are you familiar with the microservices architecture?

### How does it differ from traditional applications (monolithic architecture)? Benefits?

## **Databases**

### Difference between SQL and NoSQL?

### (SQL) What is a Primary Key? Foreign Key?

### (SQL) What is a Constraint?

### (SQL) Different types of relationships between entities in SQL?

### (SQL) How do you represent a Many-to-Many relationship?

### (SQL) Why would you JOIN tables? Can you describe different types of joins?

### (SQL) If querying a table is taking a long time, what can you do to improve the performance? (Looking for Index)

### (SQL) Advantages and disadvantages of using Indexes?

### (NoSQL) What are some of the ways data is stored in a NoSQL database?

### (NoSQL) When should you embed one document within another in MongoDB?

### (NoSQL) Advantages of using NoSQL over RDBMS? Disadvantages?

### Why do we use databases over in-memory data storage?

## **Programming Experience \*\*\* SOLID, Design Patterns, Testing**

### What is the Single Responsibility Principle?

### Are you familiar with the concept of Dependency Injection, or Inversion of Control? Why is it useful?

### Tell me about your experience with version control systems (like git)

### In what ways could you improve the way you use it?

### What are different types of testing?

### Do you have any experience with automated testing?

### Why would you use one type over another?

### What is TDD and what benefits does it offer?

### Can you tell me about CI/CD? What are its benefits?

## **Programming Exercises**

### Write a function that prints out all of the multiples of 3 and 5 between 1 and 1000.

### Write a function that, given a number, prints out the sum of all of the digits of that number. E.g., 1234 would return 10.

### Write a function that determines if a given string is a palindrome.

C# Editor: <https://dotnetfiddle.net/>

JS Editor: <https://jsfiddle.net/>

Java Editor: <https://www.tutorialspoint.com/compile_java8_online.php>

**Communication 101 (Conflict Resolution):**

(Bethel + Simon Sinek source)

•**What’s the best way to confront someone?**

1. Say how they made you feel - “I felt xyz when...”

2. Share the specific action that made you feel that way

3. Explain the impact of their action

•**Seek to Understand**

1. Never Assume

2. Seek to Understand

3. Seek to Be Understood

•**Listener** - Summarize & Repeat it back to them:

“So what I hear you saying is...”

When one is aggressive or off putting and carries a spirit

1. Never assume
2. Seek to Understand

* ASK: “hey can I provide feedback?”
* ASK: “hey I’m experiencing you this way. Is there something going on where I’m experiencing you with aggression?”

**7 Steps of Confrontation:**

1. You never assume

2. You seek to understand

3. Reflect to them what they are saying (“What I’m hearing you say is…“) They feel understood and loved by that

4. You allow yourself to be understood.

5. Invite them to find the solution with you. Ask how they feel like we can move forward?

6. Find a win-win.

7. Ask when can we implement this?

**DAVE RAMSEY SOLUTIONS**

<https://www.daveramsey.com/careers/opportunities?category=Technology&p=job%2FoKvScfw3>

(careers page continually updating)

You'll want to study for the **coderbyte TEST** with the specific coding language you want to be tested over.

**Here's info I received in the past on Dave Ramsey's culture and hiring process when students were interested:**

During the application process with Ramsey Solutions, you may have a series of written questions.

Be thoughtful in your written answers, **not just a 1 word** answer.

These are things they're looking for:

* Are you a "**Fit**" for our company and culture?
* **Values** + **Mission** **Statement** (Are you behind this?)
* Are you a **team** **player** and forward thinking?
* **Communication** - How do you handle **Conflict**?
* **Hungry**, **Humble**, **Smart** (You'll eventually read the "The Ideal team player" book by Patrick Lencioni)

**CORE VALUES:**

* self-employed mentality
* Shoot sacred cows
* Share the profits
* Momentum theorem
* Righteous living
* Fear not
* No gossip
* Team
* Crusade
* Excellence in the ordinary
* Family
* Never give up
* QBQ
* Colossians 3:23
* Marketplace service

## **1. People Matter**

If you were only allowed one core value to follow, it should be this: Put other people first. And that’s everyone—your family, your team, your customers. Heck, even your competition. Every single person deserves to be treated with respect, dignity and a caring heart. At Dave’s company, everything he and his team do is based on this simple idea.

## **2. Find the Why**

As best-selling business author [Simon Sinek](http://www.ted.com/talks/view/lang/en/id/848) says, every organization knows what they do, but very few can say why. That why—your purpose—is the foundation of your culture, and it is what will make your team passionate about their jobs. Let them know that it’s not all about the money and teach them that their work matters. It makes all the difference in the world. Find your why and share it with your team as often as possible.

## **3. Hire Smart**

Creating a team of talented rock stars who deeply care for each other starts at the hire. From leadership to an administrative assistant, take your time filling every position. And hold out—no matter how much you need the job filled—until you find the absolutely perfect person.

“We hire and keep only people who are sold-out passionate about our cause,” Dave says. “You cannot work on my team if you are simply looking for a J-O-B.”

## **4. Stay Out of the Ivory Tower**

At Dave’s company, it’s not unusual to see one of his leaders pick up a piece of trash, set up tables, or do whatever is needed to get a job done. They look for every chance to let their team know that although they’re in charge, everyone is in it together.

## **5. Recognize and Reward**

Part of letting people know they matter is to brag on them. Why? Because every individual around you yearns to be appreciated. Recognize your team early and often, and watch them strive to achieve more and more.

## **6. Schedule Some Play Time**

Ugly-Christmas-sweater contests, a battle of the bands, a few potlucks, some victory celebrations, a softball tournament … It doesn’t matter what activities you schedule—it’s worth the extra effort to set aside the time. Why? Because those who play together will bond. There is something magical about meals too. When you break bread with your team, you get to know them on a different level.

As a business owner, it’s up to you to decide if you want a building full of mindless drones or happy, fulfilled team members who will do anything to help you win. And it all begins with culture. When you create a culture that cares, anything is possible. So what are you waiting for? Start yours today.

Josh’s Resources:

<https://www.youtube.com/results?search_query=ux+design+challenge+interview>

<https://uxtools.co/challenges/>

<https://www.uxchallenge.co/>

<https://uxplanet.org/how-to-solve-a-product-design-challenge-22c4daa27117>

<https://sharpen.design/>

# **AFTER AN INTERVIEW -- IF YOU'RE ACCEPTED**

* Write a **Thank you Letter**
* Let them bring up the salary and pay information.

**NEGOTIATION**

* Are you wanting to **negotiate Salary**?
  + If you’re wanting a higher starting pay, ask in a way that’s not offensive. You can say, **“I’d like to ask if there’s room for negotiating the starting pay”** (not this is too little).
  + Focus on the market value so you know what the market salary is for your job position and range.
  + Focus on your **value** and specific things that make you valuable to the company whether it's measurable things you've done for companies in saving cost, increasing revenue, or changing company culture into positive work team environments.
  + When asking, Don't use a range like saying you want something "between" $50,000 and $70,000. Instead, ask for an **exact** or **specific** #number like $69,000.
  + Studies show you're more likely to get a raise if you ask on a **Thursday**, especially for "performance reviews" while you're already in a job.
  + Don't mention personal needs like "I need childcare".
  + In general, you can ask for anywhere between 10-30% more than the job offer without being offensive.
  + Don't be afraid to counter.
  + If they aren't able to give you what you ask and you still want this position, ask what performance reviews look like and when the company does raises.
* **Other areas you can negotiate:**
  + Job Title
  + Health Benefits (As a woman, it's okay to make sure to calculate in -- short term or maternity leave)
  + Time Off (If you need certain chunks of time off)
  + 4 Day Work Week (Work 10hrs/day in 4 days to make it a 40-hour work week and take off Fridays)
  + Learning Budget for education (For ex. $500 for additional courses in Udemy, etc.)