To get up to speed with respect to how and what to prepare, please find below certain materials/links which will be extremely useful to you. Do have a thorough look at them.

Step 1. Refresh yourself with CS fundamentals

- We expect you to have pragmatic programming experience. We look for candidates who can write modular, scalable and clean code with the ability to handle edge cases, debug errors and think of multiple solutions to a give problem.
- Coding Style Guides (Python,GoLang)
- Geek for Geeks - Study CS undergraduate topics
- Platform Engineering resources:
  - School of SRE
  - Compilation of SRE Questions
- System Design resources:
  - System design interview tip!
  - Hired in Tech

Step 2. Coding Practice + CS review

When you practice, do not use an IDE. You need to be able to write legible, compilable code without help with regards to layout, or spelling of standard library class/method names. We suggest solving similar style algorithmic/ DS problems on a google document or on paper to simulate a real interview. Several sites that provide similar problems to those typically asked in the interview are:

- Leetcode
- Codeforces
- HackerRank
- Topcoder

Step 3. Docker and Kuberenetes

- Kubernetes basics
- Kubernetes tutorial
- Docker basics
- Docker tutorial

Step 4. OS and Networking

- Introduction to Computer Networking
  - Latency numbers
- Istio
- Introduction to Linux
- Linux Performance Analysis

Step 5. Observability

- Monitoring basics
- Prometheus basics

Additional Notes/ Optional Studies:

- **Study Materials:**
    - **Courses: can choose one as per your requirements**
        - Coursera - Algorithms, Part 1
        - Coursera - Algorithms, Part 2
        - GoLang for DevOps
        - System Design Interviews
    - **Youtube Channels:**
        - Introduction to Memory Management
        - Platform Engineering Playlists
        - System Design Playlist
        - System Design Playlist - Arpit Bhayani
    - **Books:**
        - Beautiful code
        - Elements of Programing Interview
        - Coding Interview University
        - Cracking the Coding Interview

Interview behaviors to note

- Talk through your thought process about the questions you are asked. In all of Mercari's interviews, our engineers are evaluating not only your technical abilities but also how you approach problems and how you try to solve them.

- Ask clarifying questions if you do not understand the problem or need more information. Many of the questions asked in Mercari interviews are deliberately underspecified because our engineers are looking to see how you engage the problem. In particular, they are looking to see which areas leap to your mind as the most important piece of the technological puzzle you've presented.

- Think about ways to improve the solution you'll present. In many cases, the first answer that springs to mind isn't the most elegant solution and may need some refining. It's definitely worthwhile to talk about your initial thoughts to a question, but jumping immediately into presenting a brute force solution will be received less well than taking time to compose a more efficient solution.

- Access to a computer at the time of interview is required as you will be requested to write and share code. Interviewers may use Google Docs to facilitate coding in real time