Engineering Onboarding

Mission, Vision and Values

[Refer here, here and here for what Mission, Vision and Values mean]

Mission

Empower developers to build amazing experiences.

Vision

We want to be The testing infrastructure for the internet.

Values

- **Solve real problems**
  Companies that take over the world don't chase ideas and trends. Their every action is about solving a real problem.
- **Be aggressive**
  Companies that thrive are the ones that continually reinvent themselves. Move fast, make decisions, be aggressive, and iterate quickly. Push yourself to limits to deliver the best.
- **Be with the best**
  We surround ourselves with the best people, technology, advisors and tools so that we can achieve something special together.
- **Stay lean**
  Keep it simple and minimalist. Automate as much as possible, more people is rarely the answer. Less processes and less friction help do things better.
- **Be open**
  The best idea always wins no matter where it comes from. Be transparent, treat everyone equally and share feedback openly.

Engineering @ BrowserStack

There are certain expectations from every engineer at BrowserStack, and certain attributes and behaviour patterns that enable the engineer to be successful in the organization.

- **Solve the Problem**
  - The nature of the business is such that there is a lot of R&D involved.
  - A successful engineer will be able to pick up new technologies fast, to deliver on the work at hand.
  - You are likely to hit walls for many tasks. Push yourself to figure out how to get around them, or through them.
  - Ask around your team and seniors when you are stuck.
    - If you ask the obvious, you are not doing your job.
    - Your team is there to support you, not to spoon-feed you.

- **High Ownership**
  - Ownership means a lot of things.
  - Show responsibility for your work.
    - Get things unblocked with other teams, follow up where required. Figure out who to talk to in other teams, sit with them and fix your problems.
    - Support other teams when they need help on your components.
    - Be proactive about your areas of ownership and the team's areas of ownership.
    - Keep track of things in production, understand importance of production issues and treat them accordingly.
  - Be a responsible member of the org.
    - When things happen that you disagree with, or feel something is not right for the organization, raise your voice.
  - As you grow in experience, ownership means that you keep track of your systems, raise concern about issues that impact your systems, and drive the vision and development of your systems.
  - Have a good understanding of the product and business that your areas of ownership impact.
    - What does your service mean to the product. Understand the impact of your systems (and their downtimes) on the product, business and customer.
    - Understand why you are doing what you are doing.

- **Team before self**
  - Individual ownership, collective responsibility
  - If something breaks, the whole team is responsible.

    - So:
        - Support your team members in their work.
        - Expect support from your team.
        - Mentor your team members

- **Seize the opportunity**
    - We are a startup and growing.
    - There are a lot of opportunities to learn. But. No-one will give those to you on a platter.
    - Step up to the plate - there is no hierarchy in the team. If you have higher impact, and show the capability to take up some responsibility, you will be given that.
    - Do not be afraid to fail. It is more important to try than to play it safe.
        - But when you do fail, step back and figure out why.

- **Follow good engineering hygiene**
    - Write test cases around your code. In general, write testable code, and write test cases for ensuring that core functionality is correct.
    - Refactor your code as a part of your work. When you see code cruft, fix it.
    - Keep track of health of your systems.
        - Keep track of the daily health reports of your systems.
        - Keep on top of the real time metrics in your systems.
        - Stay on top of all alerts of your systems.
        - Apply required fixes, and discuss health of your systems in standups.
    - Document your work - document decisions taken and why, problems fixed, issues seen, things tried.

Onboarding Process

Timelines

SNo.	Action	Owner	Timeline
1	HR Induction	HR	Day 1
2	Product demo	Divyesh	Day 1
3	Engineering Induction	Srijon	Day 1
4	Product Overview	Srijon Nakul Yeshwant	Max closure by 2 weeks
5	Status check on Product Overview	Srijon / Nakul	Every 2 days
6	Decision on Buddy	Srijon	max by Day 2
7	Define Assignment for an Individual	Srijon	max by Week 2
8	Assignment to deliver	EM	Week 4
9	Starting Common Engineering Training	EM	2 weeks
10	Starting Team specific Engineering Training	EM	2 weeks

Relevant Documents

1) Coding guidelines here.
2) Things to do to understand the product here.
3) Scores for individuals for Engineering training here.
4) Developer training here.
5) Onboarding Engineer calendar here.
6) Oncalls @ BrowserStack here.
7) Engineering components and repos here (has links to product flows and mailing lists as well).

Responsibility of the buddy

- Act as source of information to the joinee.
- Go for lunch with the joinee.
- Go for lunch with the joinee along with other teams.
- Get the joinee added in all relevant mailing lists for the team.

Responsibility of the Team Lead

- Refer here