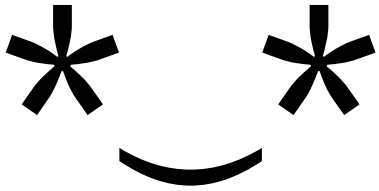
AstroSonic

A provider of leading-edge open-source solutions to modern problems

CONTRIBUTOR'S MANUAL

v0.0.3
21st May 2020



Welcome to AstroSonic

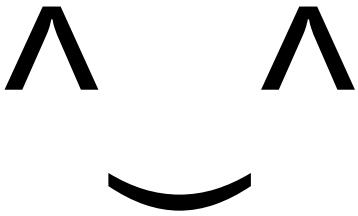
We are a budding community of open-source developers with nothing but computing devices, coding competency and big dreams in our minds. The reason why we all unite here is to leverage technology and automation to build cool applications and services as a solution to modern day problems all while learning from the experiences and having fun while doing so. No matter how old you are or how experienced you are at developing sofwares – we do realize that there is a first time for everything.

Many times, you might have dreamed of an awesome project and have gotten to work on it. Thousands of lines of code down with multiple assets built and bugs to fix, there is certainly a long way to go. But alas, you are quite unable to give your hundred percent of potential to it – be it because of monotonous assignments, meeting deadlines or just maintaining relationship with the partner of your dreams. It is okay. We all have been there and understand how sad it is to give up on a dream project.

Then we realized something great. Building cool applications and services is all about collaboration and sharing. Not every one can drive a project from the start till the end all alone. Adding to that, building projects all by yourself means to be in a constant lack of proper testing and shortage of valuable insights, thus leading to creation of something broken and with limited support. Not to mention, the constant headache of handling every aspect of building assets, writing codes and doing testcases.

There are millions of people across the globe who are interested in something you are passionate about. Being in the same community is a win-win situation for both as it would provide that person something to have fun with while you might just get your big idea executed. Once you are done, you can also prototype applications, write code and provide your insights to some other idea while working towards executing it as well. Believe it when we say, it is an endless stream of joyful contributions.

Being a community with no boundaries as to who can participate and contribute, we are very clear with the fact that you can always find yourself at home with us. Experience live projects, high stakes, real responsibility, artistic people and quirky developers. This is an opportunity to start from where you are and to rise high by building cool projects with your friends from all across the community. The team stands by with you in all the stages so you can finally get a big picture from your thumbnail.



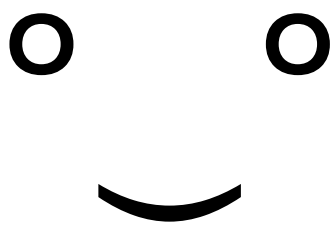
What is the community all about?

We at AstroSonic wish to extend this opportunity to all students and professionals – no matter what kind of expertise they have to get one-on-one with their status quo of leveling up several levels of contribution hierarchy before they reach a stage worth bragging about. Join today and let us know what you are good at. We are certain that the community is indeed looking for what you are good at and aching to see how you can contribute. Remember that a journey of thousand miles begins with one step.

We have been active parts of the technology industry and multiple open-source projects to know how it is like to contribute parts of code and assets. We absolutely frown upon how companies maintain Kanban boards to prioritize tasks and how issues are used to track what is needed to be done. There is absolutely no reason why task assignments should be so vaguely expressed. Every member has a different approach to solve problems so they should never be flocked in a herd.

The way we work at AstroSonic is all intuitive. The community will have its own forum for discussions and announcements. A separate room will be provided where members can brainstorm new ideas and discuss their validity. New members would not find themselves overwhelmed as a dedicated section is provided for onboarding and explaining basics of how things work. Of course for hardened professionals who want to get to business as soon as possible, we have operations.

We understand that professionals designing machine learning models might not be inclined to work on cryptography while those working with web development might not want to get their hands on data analytics. It is alright and that is why, we have special designated teams to handle tasks that we like to call task force. At times, the task forces might need to converge together in a joint operation – like for instance when a movie recommendation system requires a frontend to let end users interact.



Why is open-source a big deal?

You can find open-source almost everywhere and it is not just limited to communities and developers. It influences and welcomes participation of individuals even from the non-technical arenas, big corporates and engineering schools.

- With open-source contributions, you can build applications and services which can then be released along with their source code so that literally anyone can change them at their will. They can at least have an option to utilize the services of a vendor of their choice.

- The fact that any other type of software exists is itself strange as there is limited control over what you can control and what you cannot. Any required maintenance would need some other person to fix, re-package and assist in providing even basic upgrades.

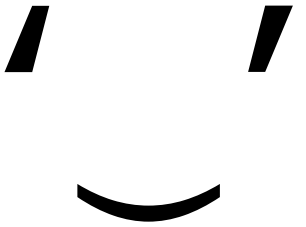
- For big corporations, open-source has become to most reliable and cost-efficient way to build great software applications. They are fine with their status quo but they cannot look away from the insights and reviews that open-source communities tend to provide.

- Ever wondered why big industries organize hackathons and ideathons? Employees seldom provide good contribution out of will and it is only a good old competition that brings the best out of people. It is win-win as even corporations get good ideas to bank on.

- Individual developers should seriously consider open-source projects as their hobby. It is fun hanging out with a bunch of like-minded folks on forums discussing how a piece of code can be optimized. You can can hone your skills by getting on such an open platform.

- There are always new ideas to explore. Members are willing to discuss problems with you way more than your fellow classmate or your peer in the next cubicle. Contributions are welcome – be it a tiny bit of code, a discovery of bug or simply a typo correction in docs.

- People from non-technical areas do not even know that they are contributing to open-source already. Almost every web application hosted on a GNU/Linux server traces issues and understands when things go wrong. They are actually hunting bugs this way.



Why should I contribute to open-source?

Contributing to open-source describes a lot about you as a community member.

- **You are passionate**
  You are writing thousands of lines of code, drawing diagrams, building assets and running tests because you are passionate about it and not because you want to get paid for it. You absolutely love what you are doing and in the long run, your future employer expects the same sort of interest while working with them.

- **You are confident**
  You are confident about your creation and ready to face public criticism about it. With those reviews and comments taken on a positive note, you are willing to work tirelessly to make all changes to build better stuff. An average developer breaks a sweat to get an upstream while you know how good you can be.

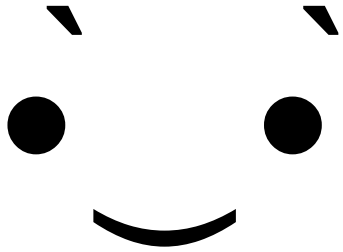
- **You are comfortable**
  You are not overwhelmed when your project scales up rapidly with multiple dependencies and concurrent contributors. It clearly tells a lot about how comfortable you are in reading a foreign code and how you can adapt to the stated requirements. Better yet, you can make people match your requirements too.

- **You are interactive**
  You are aware about how you should interact with the fellow contributors and how you should ask for assistance. Working as a player in a multi-talented team is important and soon you would find yourself giving your best to bring together the best works of every teammate into a single magical project.

- **You are informed**
  You are skilled about a stack and have garnered tons of experience with many projects. To move on to newer a stack would require you to go through documentation – which is cool. But getting to learn new technologies from your fellow community members while teaching them what you know is definitely cooler.

- **You are helpful**
  You are going to start from where you are. If it is zero that you are starting from, you will most likely get lots of help for your task. If you have previous experience in your field, you can pretty much help other newbies to start off. Open-source contributions are done as a hobby and you get also get to manage your time.

- **You are collective**
  You are a part of a collective. Open-source movement is all about helping people through software applications and services with open codebases. There is a palpable sense of freedom in choosing how you would tackle problems. You are free to make fork a project and add stuff that you think would make it better.



What are we looking for?

We are looking for people with some of the following modest requirements

- You should have at least 6 months or longer experience in any programming language
- You should have been involved in the development process of at least 1 project
- You should have great verbal and written communication skills
- You should have a friendly demeanor and collaborative in your tasks
- You should have an inclination to learn new things while building cool stuff
- You should have an expertise to find solutions from popular search engines
- You should have an ambition to be an open-source contributor
- You should have an experience in the prominent office enterprise applications
- You should have an experience in the prominent graphic design suites

We are looking for only some of the above stated requirements. We understand that it would be very unfair if we ask a cybersecurity analyst to have an experience in design suites and a frontend designer to have an experience in C++. We would not do that. You are in for a quick chat about your interests and objectives if you satisfy any three (or more) of the above stated requirements before we welcome you in our community.

We will try our best to avoid unpleasantness so kindly note

- You should not look for any kind of monetary advantage from any contribution
- You should not possess the know-it-all attitude
- You should not be selective about the tasks that have been assigned to you
- You should not be always looking for a smooth-sailing experience and certainty
- You should not avoid risks and dangers that may come during contribution
- You should not be irregular with meeting with your stated commitments
- You should not have a desire to own projects made together by the community

How can I contribute?

This is how you can contribute to our open-source community. We have a total of seven divisions with their own distinct sets of skills and expertise. If you think you fall in it, you are more than welcome to fulfill those tasks. These divisions would not divide you into teams though as teams are purely based on what project you are up to. It is certainly possible to have people from multiple divisions to be in the same team just as much as it is possible to build a team entirely with members from a single division.

1. **Architecture Division**
   You have a strong sense about how things should work and why they should work the way you specify them to. You are good at making charts and seeing the bigger picture. We can expect valuable insights and idea reviews from you for you can pretty much visualize them before they are even implemented.

2. **Development Division**
   You are good with programming languages and your logic hits bullseye most of the time. You like to spend countless hours in front of black screens of IDEs and nothing brings you more joy than a snippet working just as expected in the very first run. We can expect rich codes and optimal logic from you.

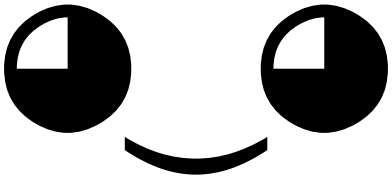
3. **Design Division**
   You feel there is something wrong with how things work right now and you can definitely walk the path of innovation to make it better. You love designing creative and efficient solutions. We can expect massive brainstorming and new ideas from you for you know what should be worked on.

4. **Frontend Division**
   You understand the fact that a good project would not make a great impact among the end-user and other friendly communities if it fails to look and feel good. You have an affinity to web development and we expect you to spin a rather joyless piece of code into an eyecandy with your magical skills.

5. **Service Division**
   You realize that there is a lot of code and systems that perform shabbily and someone has to get their hands dirty to make things right. You feel right at home when you run those benchmarks and lint evaluation your code. We expect you to test, report and fix bugs as and when you come across them.

6. **Documentation Division**
   You see that there are a lot of people around who cannot quite get into development for things are not quite explained to them. You take it as your responsibility to bridge the gap and make things lucid. We expect you to write project documentation in a way people would clearly understand them

7. **Outreach Division**
   You know that a community is better off with more people and friends to goof around with. You get it that it is fun to build things together and have fun while doing it but the more people you have, the merrier it will be. We expect you to speak for our community and reach out to interested people.



What is the community workflow?

The community follows two distinct kinds of workflow which makes us different from other open-source communities.

- **Release Workflow**
  Members participating in this workflow are completely dedicated onto building a single project at any point of time. They are in complete control of the direction of the project but cannot decide how successful it can be down the line. This requires a minimum of five effective contributions per week. This workflow is the most focused and hugely challenging.

- **Rolling Workflow**
  Members participating in this workflow are completely dedicated onto maintaining multiple projects at any point of time. They do not have a say in the direction of the project but can pretty much decide how successful it can be down the line. This requires a minimum of three effective contributions per week. The workflow is moderately focused and greatly challenging.

How do we work?

For all the tasks that are performed in our open-source community, we follow the time-tested and widely trusted pull request model. This simplifies the way of creating workflows around task reviews while quantifying progress and giving insights as to what was contributed. This model was widely popularized by GitHub and it slowly found its way in all kinds of workflow – open-source and proprietary alike.

Not aware how it works? It is okay. Read on the following points.

- All the project assets and codes are stored in a remote location. This remote location is called a repository. Every project has their stuffs collected and stored in their own repository. Keep in mind that a regular change is needed in it to add new features or fix existing bugs.
- The event of making changes in a remote repository is called pushing. You can imagine how chaotic it can be if that repository is kept open to pushes made by everyone. Thus, owners of the repository are the only ones that are allowed to make pushes to the repository.
- You might begin to question the openness of open-source if only a selected group of people are allowed to make changes. You are allowed to make contributions but like every stuff out there – your work must be monitored and evaluated before being added.
- To begin making changes, you need to have your own copy of the remote repository. The event of making a copy of a repositories to your version control account is called forking. This would create a fork of the repository with all the changes made till the date of forking.
- This fork gives you complete authority to make any change you want. Keep in mind that all the changes you make would stay in your fork and none of it would be reflected in the source repository. We will talk further about how you can do so on the source repository.
- Your fork of the source repository is stored remotely and in order to make changes to it, you need to download it to your working device. This event of downloading a copy of your fork is called cloning. Now navigate to the folder where you have cloned your fork to start.
- There might be an uncertainty as to if your additions would actually be helpful. It is normal for developers to end up breaking their project while making new additions so you might want to isolate your changes. This event of isolating all changes is called branching.
- Make all the changes you wish to see in the assets and codebase of your newly created branch. You must save those changes in order for them to persist. This event of saving changes is called committing and it helps you track changes and rollback to a previous state.
- All commits that you have made so far would stay only in the cloned copy of your fork. In order to make sure that they make it to the remote fork of your version control account, you need to upload them. This event of uploading commits is called pushing.

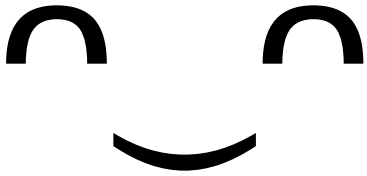
- Now that all the made changes are in your remote forked repository, it is time to make these changes on the actual source repository as well. You need to ask the repository owner to take a look at your changes and add them. This event is called making a pull request.
- This pull request is for bringing together the changes made in your branch to the changes made in the primary branch of the source repository. The primary branch is generally known as master and the event of bringing together two branches is called merging.
- Give the source owners some time. Let them review your changes and they will come back to you with more change requests. If all goes good, your changes might just get merged to the master and you would get labeled as a project contributor. Keep going for more now.

You know what to do right now. But here is when you should do this all.

- When you are assigned a task over a collaborative chat, you can follow the aforesaid method to add your part of the contributions.
- When you see an active issue on a bugtracker, you can ask fixing to be assigned to you before you start adding your fixes to code.

To protect the contributions made so far, the following must be kept in mind.

- All commits must be made in a branch, which is not master
- Fork master to source master merging is disallowed
- There can be at most one pull request per issue fixing
- Avoid merging branches if there are active conflicts
- There can be at most one source fork per VCS account
- One commit per file is suggested to ease tracking changes
- Single push for multiple commits is suggested
- Concurrent tasks on multiple branches is not recommended
- Branches should be named appropriately
- Commits should have concise message under 50 chars



What do I take away from this?

There is a lot to benefit from being in our community.

The quantifiable ones are listed below.

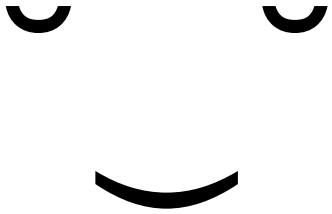
- You obtain footprinted contributions on popular VCS hosting websites
- You end up having multiple projects developed in your sleeve
- You see an insurmountable growth in your professional network
- You acquire multiple skills from a variety of division domains
- You build projects that reach out as articles in Medium, Dev.to etc.

The unquantifiable ones are listed below

- You learn the art of self-sustainability with a sense of ownership
- You become comfortable with uncertainty and unpredictability
- You find a surge in your team-building and leadership skills
- You feel the satisfaction of having shipped a project built from zero
- You get to have fun hanging out with like-minded people

The following are definitely not the takeaways we offer

- You will not be given any monetary compensation for your work
- You will not be recommended to any corporation from us
- You will not be offered any certificate as a proof your experience
- You will not be provided any measurable evaluation for your performance
- You will not be presented any perks or gifts of any kind



What are the regulations I must follow?

This documentation is constantly worked upon. We are certain to add content to this section down the line.