

OPEN-SOURCE CONTRIBUTION AS A WAY TO BUILD YOUR PORTFOLIO

Code. Collaborate. Contribute. Create Impact.

How contributing to open-source projects can enhance your skills, showcase your work, and open doors to exciting opportunities.



Vishwakarma Institute of Technology, Pune

Welcome to the June 2026 Edition of the IT Bulletin

Open-Source Contributions: The Smartest Way to Build Your Portfolio

This edition explores Open-Source Contributions, one of the most effective ways for students and developers to gain real-world experience while building a strong professional portfolio. Open-source software allows anyone to contribute to projects by writing code, fixing bugs, improving documentation, testing features, and collaborating with developers from around the world.

By participating in open-source communities, contributors develop technical expertise, strengthen problem-solving and teamwork skills, gain hands-on experience with industry-standard tools like Git and GitHub, and create a public portfolio that showcases their abilities to recruiters and employers. From web development and artificial intelligence to cybersecurity and cloud computing, open-source projects provide endless opportunities to learn, innovate, and make a meaningful impact.

Discover how contributing to open-source projects can accelerate your learning, expand your professional network, and help you build a portfolio that stands out in today's competitive technology industry.

Introduction to Open Source Contribution as a Way to Build Your Portfolio



Participating in open source initiatives has become one of the key means for young talents such as students and beginners in the industry to get hands-on experience and to work on their portfolios. Unlike studying theory and working on projects done on the desk in one's room, open source work enables one to interact with live system code used by thousands, if not millions, of people as well as other seasoned developers. This helps gain thorough knowledge of how the industry works in terms of workflows, code version control processes, etc. For students, open source participation provides much more than career experience. It gives them a chance to acquire certain skills while contributing, such as learning programming languages, fixing bugs, etc. Open source technology offers opportunities to develop competencies hard to obtain elsewhere: that is communicating with maintainers, writing understandable code, and reacting to criticism appropriately; in other words, the skills that engineering teams are looking for. A GitHub account with a lot of useful commits and pull requests will have a bigger impact than a line in the CV, as anyone can confirm the work done by self. Contribution does not lead only to technical progress, but also connects students to the worldwide community of developers contributing people to the world of networking and future job opportunities — many employers find a strong leader in passionate open-source contributions.

This article will tell you what open source is, its advantages, the difference it makes to your CV, and how to contribute your first code.

What is Open Source?

What is Open Source?

Understanding the Basics in Simple Words At its core, Open Source means the internal "source code" of the software (the recipe) is made public. Anyone can look at it, change it, and share it. This is built on community collaboration.

What Actually Counts as a Contribution?

A common mistake is thinking you must write complex code. This is far from true. Open Source thrives on diverse skills. A "contribution" is any effort that improves the project. For example:

Writing Code: Fixing bugs or adding features.

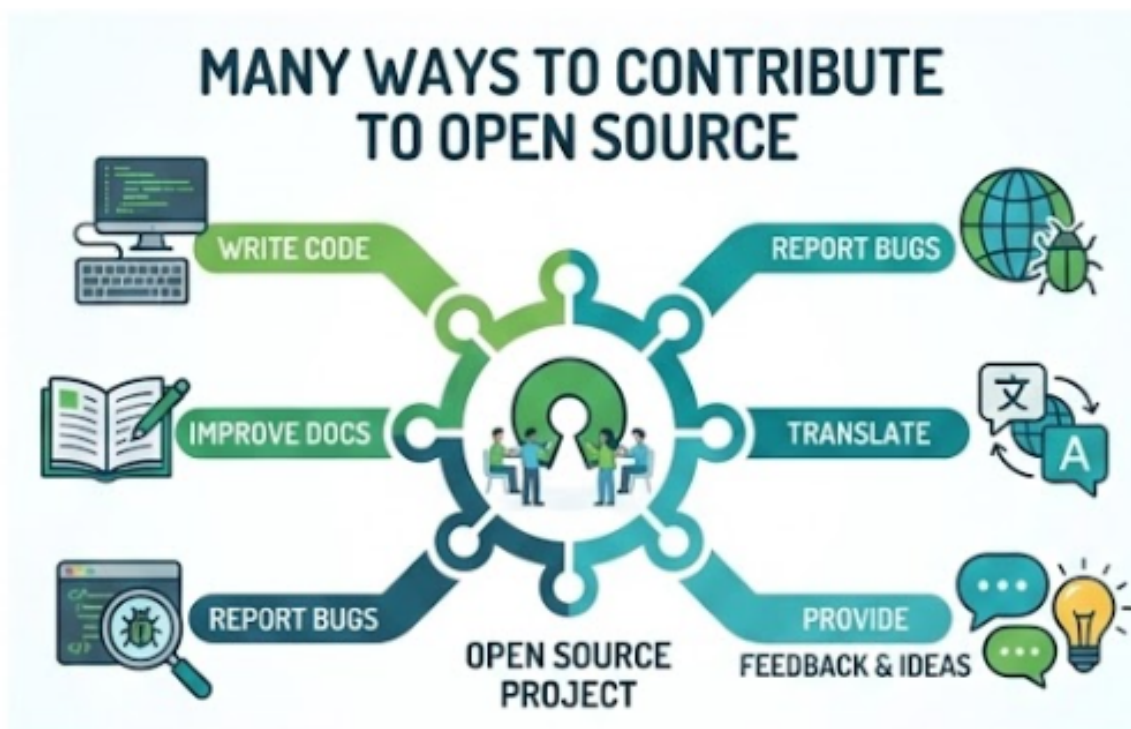
Improving Documentation: Writing clearer guides and manuals.

Reporting Bugs: Finding errors and detailing how to fix them.

Testing: Beta-testing new software versions.

Design: Creating icons or user interface elements.

Translation: Making the software available in new languages.



How Open-Source Contributions Benefit Your Portfolio

In today's competitive job market, recruiters have started looking beyond academic grades and degrees. Open source contributions allow students to demonstrate technical skills, collaborate on real world projects, and build a portfolio that showcases their abilities to employers.

Open-source contributions provide students with an opportunity to apply their knowledge in practical, real-world scenarios while building a credible professional portfolio. By contributing to publicly accessible projects, individuals demonstrate their ability to write, review, and maintain quality code, solve complex problems, and work with industry-standard tools such as Git and GitHub. These contributions also reflect consistency, initiative, and a willingness to learn new qualities that employers highly value. Furthermore, collaborating with experienced developers across diverse projects enhances technical proficiency, communication, and teamwork, while exposing contributors to professional software development practices. A well-maintained record of meaningful contributions serves as verifiable evidence of a candidate's skills, making their portfolio more authentic, competitive, and impactful during internship and placement opportunities.

Unlike personal projects or classroom assignments, open-source contributions expose students to real-world software development practices. Contributors learn to work with existing codebases, use version control systems, follow coding standards, and collaborate through pull requests, code reviews, and issue discussions. This hands-on experience develops technical proficiency while fostering essential professional skills such as communication, teamwork, and adaptability. As contributors gain experience, their GitHub activity and accepted contributions become credible evidence of their capabilities, demonstrating to recruiters that they can effectively collaborate, learn new technologies, and contribute to large-scale software projects.

Why contribute to open source?

One of the most effective ways of building up a real and verifiable portfolio is to take part in open-source projects. Unlike individual projects, your contributions show that you are able to work within existing codebases, stick to team conventions, and have your code reviewed and accepted for merging by other people-this is something that goes well beyond what can be stated in a resume.

As you go along, you pick up skills more quickly by reading the code written by experienced developers, gain a clear record of your performance, and develop a network of collaborators who might later help you get new opportunities. Most advantageous of all is that it is not necessary for your contributions to be large in order for them to be important-even correcting a typo, fixing a minor bug, or enhancing the documentation all qualify as genuine and significant work.

Also, contributing gives you something that a classroom project could not-real feedback from actual developers. When your code is reviewed, you learn how experienced programmers think, what mistakes you should avoid, and how to write code so that other people can understand it and it can be maintained. Gradually, these contributions amount to something valuable-evidence that you can work together, take feedback, and deliver working code, skills which are just as important as technical knowledge when you're beginning your career in IT.

Popular Open-Source Platforms

The Software Running the Modern Web:

Open-source software isn't a niche concept; it is the foundation of the modern digital world. Entire ecosystems, from the servers hosting your favorite websites to the tools developers use daily, rely heavily on open-source platforms.

Core Examples of Open-Source Power:

The open-source ecosystem is vast. Here are some of the most influential platforms that demonstrate its scale and versatility:

Linux (Operating System): The backbone of the internet, powering the majority of the world's web servers, supercomputers, and even Android smartphones.

Git (Version Control): The industry standard tool for developers to track code changes, collaborate seamlessly, and manage software versions.

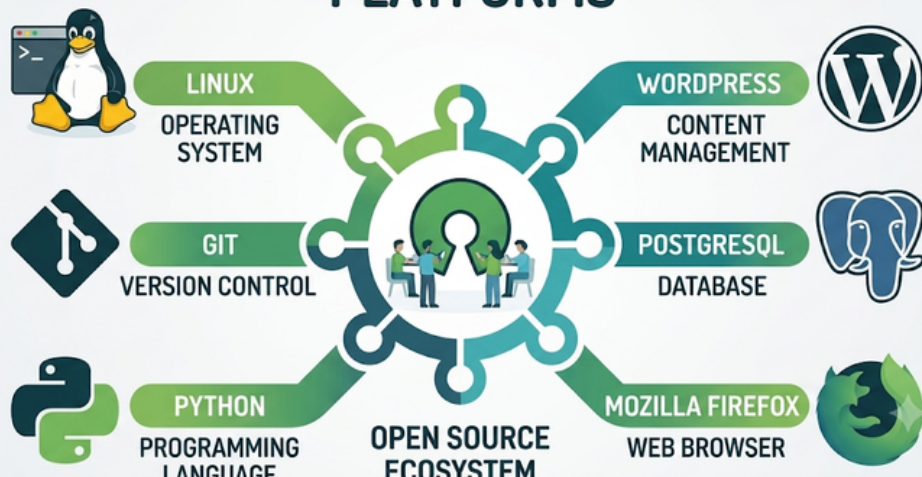
Python (Programming Language): A highly popular, beginner-friendly language dominating fields like artificial intelligence, machine learning, and data science.

WordPress (Content Management): The driving force behind content creation, powering over 40% of all websites globally.

PostgreSQL (Database): A remarkably powerful, reliable, and secure database system trusted by massive enterprise applications.

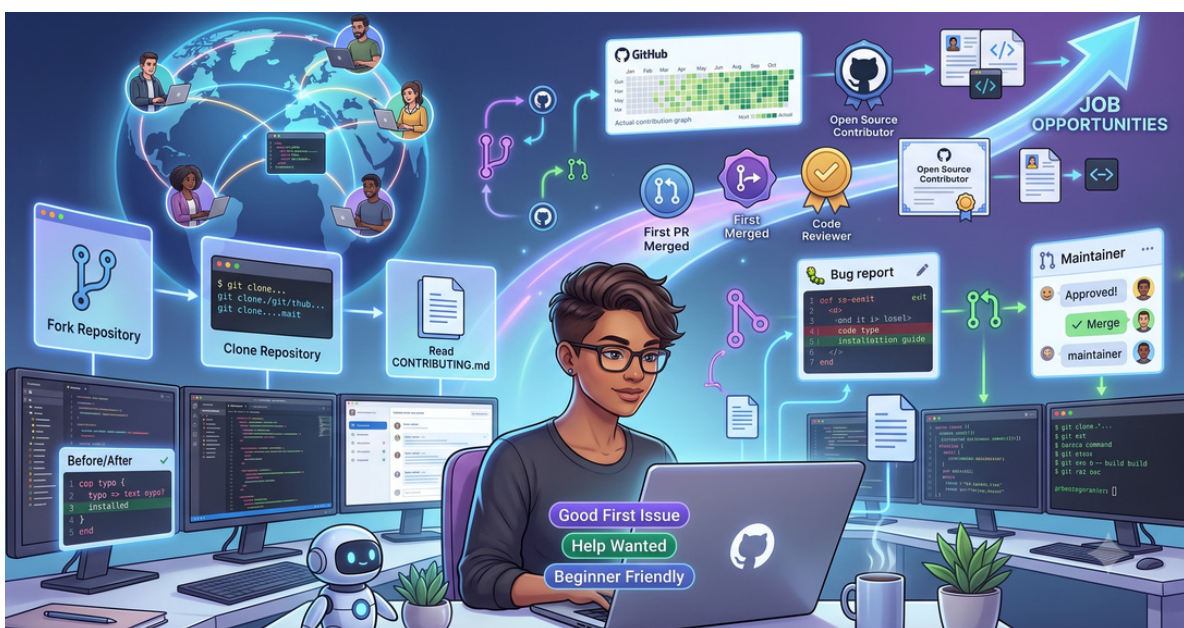
Mozilla Firefox (Web Browser): A privacy-focused browser built and continuously improved by a global community of contributors.

POPULAR OPEN-SOURCE PLATFORMS



How to Get Started

Although starting to contribute to open-source projects may sound difficult to most people, it is, in reality, simpler than they think - the only problem is that those who are new to it do not know where to begin. However, the brightest idea is that any big project welcomes beginners, provided they do it properly. Check the platforms such as GitHub or GitLab and find the repositories that have tags for beginners such as good first issue, help wanted, or beginner-friendly. The issues are listed specifically for new developers so that they can contribute to the project without any special knowledge about the code. Before you start coding, you should read the file CONTRIBUTING.md since it will explain the coding style, the installation process, and the rules for making a pull request. Forking and cloning as well as launching the project on your own computer is one of the best ways to gain insight into the project. A lot of contributors often begin with writing or improving the documentation by correcting typos or explaining the installation process, which is a good way to get acquainted with the workflow of the project while still being useful. After that, it's better to work on small but tested bug fixes rather than major features since it's much more effective for making the maintainers trust you. As time passes, this becomes a documented performance record that is recognized as more valuable than what other students can present.



Challenges & Tips

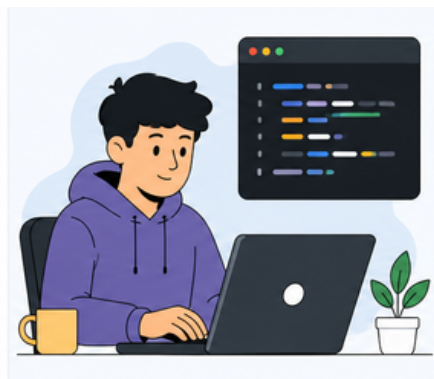
Every open-source contributor begins as a beginner. While the journey may seem challenging at first, overcoming these obstacles is an essential part of developing technical expertise and collaborative skills. Understanding the common difficulties and adopting the right approach can make the contribution process both rewarding and enjoyable

While open-source contributions provide valuable learning opportunities, beginners often encounter several challenges during their initial journey.

Selecting a project that matches one's skill level, understanding unfamiliar codebases, and learning Git and GitHub workflows can seem overwhelming at first. Many first-time contributors also hesitate due to the fear of making mistakes or receiving feedback during code reviews.

However, these challenges are a natural part of the learning process and help develop both technical and professional skills. Beginners should start with repositories that label issues as "Good First Issue" or "Help Wanted," carefully read the project's documentation and contribution guidelines, and begin with small tasks such as fixing documentation or resolving minor bugs. Seeking guidance from maintainers, accepting feedback positively, and contributing consistently are key to building confidence and improving technical proficiency.

With patience and persistence, every contribution becomes a meaningful step towards becoming a skilled developer while strengthening one's portfolio and collaborative experience.



Conclusion

Open-source contribution is no longer something developers can choose to do. It has become one of the real and easy ways for students to show what they are truly capable of. From writing your line of code to seeing a pull request accepted, every part of the journey creates not just a better resume, but a better developer.

As this edition has shown, contributing does not need skills or many years of experience. It needs curiosity, effort and the ability to start with steps. Whether it is fixing a spelling mistake, reporting a problem or eventually adding a feature, each contribution leaves a clear and lasting sign of your progress as a technologist. Something that hiring managers, teachers and future team members can all see.

As you continue your path through the Department of Information Technology, we suggest you look into sites like GitHub and GitLab, find a project that interests you and take that step. The open-source world is ready. Your collection of work will appreciate you for it.

Happy Contributing!

References

- GitHub Docs, “Contributing to open source,” GitHub Documentation, 2026. [Online]. Available: <https://docs.github.com/en/contributing-to-open-source/github>
- Open Source Guides, “How to contribute to open source,” Open Source Guides, Jul. 2026. [Online]. Available: <https://opensource.guide/how-to-contribute/>
- GeeksforGeeks, “How to contribute to open source?,” GeeksforGeeks, Sep. 2020. [Online]. Available: <https://www.geeksforgeeks.org/how-to-contribute-to-open-source/>
- GeeksforGeeks, “Contributing to open source: Getting started,” GeeksforGeeks, Oct. 2016. [Online]. Available: <https://www.geeksforgeeks.org/contributing-to-open-source-getting-started/>
- DEV Community, “Open source contribution on GitHub,” DEV Community, Sep. 2023. [Online]. Available: <https://dev.to/crescentparth/open-source-contribution-on-github-4p4g> (use the actual slug from your browser if different).
- GitHub Docs, “Finding ways to contribute to open source on GitHub,” GitHub Documentation, 2026. [Online]. Available: <https://docs.github.com/en/get-started/exploring-projects-on-github/finding-ways-to-contribute-to-open-source-on-githubopenresource>
- GitHub, “github/docs: The open-source repo for docs.github.com,” GitHub Repository, 2019. [Online]. Available: <https://github.com/github/docs>

Student Editors



Bhavesh Patil
SY IT-D



Srushti Mule
SY IT-D



Arya Bhaiya
SY IT-A



Shivam Gunjal
SY IT-D