GSoC 2023 Proposal

Strengthening Underserved Segments of the Open Source Pipeline

Nandini Saagar

Strengthening Underserved Segments of the Open Source Pipeline

OSPO

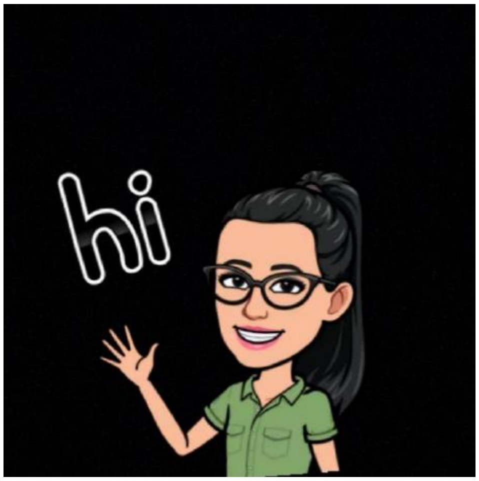
Nandini Saagar

Name: Nandini Saagar
Major: Ceramic Sciences and Engineering
Degree: Integrated Dual Degree (IDD)
Year: Sophomore
University: Indian Institute of Technology, BHU (Varanasi)
GitHub Handle: @Nandinisaagar
Email: saagar.nandinis.cer21@itbhu.ac.in
Phone: (+91) 7620412672
LinkedIn: Profile
Country of origin: India
Timezone: Indian Standard Time (UTC +5:30)

About Me

Hi! I am Nandini Saagar, a Ceramic Sciences and Engineering sophomore at the prestigious Indian Institute of Technology BHU (Varanasi). I am a regular open-source contributor always running after the latest tech trends! I am a passionate, hardworking, and dedicated individual who will achieve everything she sets her mind to!

I plan to become an integral part of the UC OSPO community this summer and continue the relationship with them long after the same is over.



Project Abstract

Contributing to an open-source project offers novices the opportunity to join a community of practitioners, build a technical portfolio, gain experience with industry tools and technologies, and have a real-world impact. This project seeks to invite and support broader, more diverse participation in open source by supporting early contributors – especially those who have been historically minoritized within tech.

The project will aim to accomplish the following tasks:

1. **Creating content** that anyone with some Open Source experience can use to help and guide other completely new students to the journey of OpenSource, GitHub, and all the relevant technologies. This step would involve creating content in the form of transcripts, markdown guides, blogs, and some video content, which would serve as instruction manuals or workshop notes.

2. Providing a medium and a **platform for all contributors to share their various OpenSource experiences** and testimonials. The same can be done in textual or multimedia formats like videos and photographs.

3. Conducting an **Open Source Themed Hackathon/Scavenger Hunt:** We can organize a simple scavenger hunt for students on GitHub, where they can be provided simple tasks like creating issues and commits! GitHub workflows are an excellent tool for auto-evaluation of the same. Technical documentation will also be provided on how other users can create their own hunts.

4. **Social Media Engagement**: Social media is where most youngsters are these days. Thus, we should be leveraging the power of the same to get young minds acquainted with the technical and open-source world at a young age so they can consciously choose to explore it further.

Workflow and

Detailed Execution Strategy

As explained above, this project will be divided into 4 major parts. I will explain each of them in greater detail and how I plan on achieving the same:

1. *Creating Technical Content*

Creating content that helps new students navigate the world of Open Source and related technologies is incredibly important for several reasons:

1. **Increased Accessibility:** Open Source technologies have the potential to revolutionize the way we work and collaborate, but this can only happen if they are accessible to everyone. Creating content that is easy to understand and follow can help lower entry barriers and make Open Source accessible to a broader range of people.
2. **Building Community:** Open Source is built on a foundation of collaboration and community, and by creating content, you can help build and strengthen that community.
3. **Sharing Knowledge:** Sharing your knowledge and experience is one of the core tenets of Open Source. Creating content can help spread that knowledge and empower others to contribute to the community.

Taking motivation from the above reasons, I want to create content on the following topics:

1. Introduction to Open Source
2. Getting started with Git and GitHub
3. Contributing to Open Source projects
4. Finding and joining Open Source communities
5. Building Open Source projects from scratch,

This content would be created in the format of blog posts under the banter of OSPO, providing a starting point for newbies! More technical content on language and tech-specific topics can be created later on the basis of the interests and reach of these blog posts. We could get other members from the OSPO community themselves to provide feedback to the said blog posts and act as co-authors so that the reading community can be assured that the content and advice provided in the same is vetted by experienced industry experts.

When creating your content, I would use clear and concise language and provide plenty of examples and hands-on exercises. While creating the content I’ll also consider using various media, such as diagrams, screenshots, and code snippets, to make your content more engaging and easier to understand.

2. *Platform to Share Experiences*

Sharing experiences in Open Source is arguably the most essential part of the Open Source pipeline for several reasons:

1. It aids in **learning and improvement** as by sharing your experiences in Open Source, you can help others learn from your successes and failures. This can help the community as

a whole to improve and grow as people learn from each other's experiences and build upon each other's work.

2. It fosters a **sense of community and lays a strong foundation** for collaboration. Sharing your experiences can help build and strengthen that community by fostering connections and encouraging people to work together.
3. When people see others sharing their experiences in Open Source, it **can encourage them to get involved and start contributing themselves**. This can help grow the community and bring in new perspectives and ideas.
4. By sharing your experiences in Open Source, you can **build trust** with other members of the community. This can be especially important when working on collaborative projects, as trust and open communication are essential for success.

While the necessity of the same can be seen clearly from the above points, doing the same, especially reaching out to beginners, is often not easy. To make the same more accessible and doable, I propose the creation of a new repository, “**AWESOME OPEN SOURCE!**”

Awesome-Project Name type repositories in GitHub are curated lists of resources related to a particular topic or technology. These repositories are created and maintained by community members, and they typically contain links to tools, libraries, tutorials, blog posts, videos, and other resources related to the topic.

For example, an Awesome-Python repository might contain links to popular Python libraries, tutorials on using Python for data analysis, and resources for building web applications with Python. These repositories can be incredibly useful for anyone looking to learn more about a particular topic or technology, as they provide a curated list of high-quality resources that have been vetted by the community.

The "Awesome" prefix has become a convention for these repositories. Hundreds of Awesome-Project Name repositories cover various topics, from machine learning and data science to blockchain and game development. Anyone can create an Awesome repository on GitHub, and they are a great way to contribute to the community by curating and sharing knowledge and resources.

In our **Awesome OpenSource** repository, all the Open Source contributors could provide links to their blogs, testimonials, and experiences in OpenSource. This would provide a centralized place for a curated list of resources and allow people to reach out to other developers! People can use the **Issue** tab to post about different problems they face, where other community members can try to help them. **Pull Requests** can be used to allow other contributors to add their own experiences to the repository.

Next, we can also build a web portal to enable better communication. This step aims to bridge the gap between those who want to learn about Open Source and those who want to teach about it. In this step, we can create a new portal (or integrate it with the already existing OSPO website), so that any individual/organization which wants to teach about Open Source to their communities (schools and colleges would be a great example of the same) could sign up and express their interest. Similarly, other experienced Open Source contributors who think they want to conduct such sessions and interact with these communities can also sign up on the site. Our portal would connect these two, thus allowing indiscriminate access to learning opportunities everywhere. The contributors can use the above-created guides to plan their sessions and upload pictures, and testimonials of their experience on the same portal, which can then be used by OSPO to raise awareness. While **this is a future prospect** for now, **I would like to contribute to the same, even after the GSoC timeline is over, as I am genuinely passionate about the same.**

3. Hackathon cum Scavenger Hunt on GitHub

A treasure hunt on GitHub would be an exciting and engaging way to introduce beginners to the various features of GitHub. By creating a fun and challenging game, we can encourage beginners to explore different areas of GitHub and learn about its features in a hands-on way.

To implement the treasure hunt, we could use GitHub's built-in features, such as issues, pull requests, and branches, to create a series of challenges that participants need to complete. Each challenge could be designed to showcase a different aspect of GitHub, such as:

1. Creating a repository
2. Forking a repository
3. Creating a branch
4. Creating a pull request
5. Adding collaborators to a repository
6. Using GitHub Actions
7. Using GitHub Pages
8. Using GitHub CLI

Participants would need to solve each challenge to progress to the next one, and they could earn points or badges for completing challenges or solving them quickly. Evaluation of some of the parts of the above challenges can be done with **GitHub workflows.** They can be easily used to automate some behaviors, like showing custom messages to the user when they create an issue or a pull request or running some actions when they push a change! The same would also be a great way to acquaint them with the power of GitHub.

The scavenger hunt can also **promote reading documentation** of the OpenSource projects. The simplest way to integrate the same is in the form of QnA. One sample scenario can be, when creating an issue, we can use the **issue template** to ask a question to the user taking the hunt, and provide them the documentation link which they would need to read to answer the question.

After the creation of the issue, we can run a script to get the user's answer from the issue description, evaluate it, and use GitHub bot to guide them to the answer! (The technical details of the same would require research on the workflows, but the same should be achievable nonetheless!)

Other parts require human interaction, which can be achieved with the help of organizers. I would also write a **technical guide** on how to organize the same for other use cases and highlight the general principles of the same so that anyone can conduct their own GitHub treasure hunt.

I would also create similar technical documentation to host the treasure hunt on GitLab as well. GitLab and GitHub have great interoperability, and many features. This would allow the conductors of the hackathon to choose the platform that they are most comfortable with to conduct the same.

I also plan to conduct at least one such treasure hunt during the GSoC period, which would be open to all! To promote the treasure hunt, we could use social media, email newsletters, and other channels to reach a wide audience of beginners who are interested in learning about GitHub. We could also partner with schools, coding boot camps, and other organizations to encourage participation and offer prizes for top performers.

Overall, a treasure hunt on GitHub would be a fun and engaging way to introduce beginners to the various features of GitHub and help them learn hands-on. Using GitHub's built-in features to create challenges, we can create a game that is both educational and entertaining and encourages participants to explore and learn about different aspects of the platform.

4. Social Media Engagement

Open-source software and technologies are becoming **increasingly popular among the 14-18 age group**. Many young people are interested in technology, programming, and software development. They see open source as a way to learn new skills, contribute to a community, and positively impact the world. According to a survey by GitHub in 2020, **94% of developers between the ages of 15 and 24** use open-source software, and 81% contribute to open-source projects. Additionally, a **report by Red Hat found that 87% of Gen Z developers** (those born between 1997 and 2012) believe that open source is important for their careers.

Furthermore, many educational institutions are incorporating open-source technologies into their curriculum, and there are numerous initiatives that aim to introduce young people to open-source and encourage them to get involved. For example, Google Summer of Code, Outreachy, and OpenHatch are programs that offer mentorship and stipends to students who contribute to open-source projects.

Several social media platforms can **be used to promote the open source community and introduce it to the 14-18 age group.**

1. **GitHub:** This platform is designed to host and collaborate on open-source projects. It is a great place to showcase open-source projects, encourage collaboration, and connect with like-minded individuals.
2. **Reddit:** The website has various subreddits dedicated to open-source and technology topics. You can share information about open-source projects, discuss ideas, and engage with the community.
3. **Twitter:** Twitter can be a powerful tool for promoting open-source projects and connecting with the community. You can share updates, interact with followers, and use hashtags to reach a wider audience.
4. **YouTube:** You can create educational videos on open-source technologies or share presentations from open-source conferences. YouTube has a vast and engaged audience, and your videos can reach many people, especially the younger generation.
5. **Discord:** Discord is a chat platform that has become popular among the gaming community but can also be used for other communities. You can create a server for the open-source community where people can discuss, collaborate, and share their knowledge.
6. **Instagram** is the most widely used social media platform worldwide by the 14-18 age group**.** According to a 2021 survey by Pew Research Center, 71% of U.S. teens between the ages of 13 and 17 use Instagram, making it the second most popular social media platform after YouTube.

We should keep Instagram and Twitter as our primary social media platforms targeting the younger generation, Instagram for primarily video content, and Twitter for most textual and GIF-based content.

Our social media strategy would have several parts:

1. We could use different catchphrases and sayings about open source, like **“purposely lost”** and **“Life would be much easier if I had the source code,”** to get the youngsters acquainted with the actual philosophies of OpenSource and to pique their interest in the same. Due to these short one-liners, the target audience would be intrigued by the open-source concept and thus would take their time googling and researching about the same online (which is the most critical part of open-source in itself).

2. Several **popular content uploaders** on YouTube and Instagram post about beginner-level technical content. Some popular ones are:-

1. *freeCodeCamp* (YouTube)
2. *Traversy Media* (YouTube and Instagram)

3. *Tech with Tim* (YouTube and Instagram)
4. *The Net Ninja* (YouTube)
5. *Programming with Mosh* (YouTube and Instagram)
6. *Corey Schafer* (YouTube)

All of these creators are specifically geared towards only creating content for beginners and have helped to reduce the barriers to tech by a great amount. We can take inspiration from them, and try to create our short versions of GIFs (say creative coding showcasing beautiful illustrations) or short videos (under 30s videos that basically explain a framework and why you should explore it) and post them on our socials. We can even use our socials to boost their engagement and content, which in turn would not only increase our engagement but also help in achieving our goal of promoting OpenSource:

1. **Share their content on your social media channels:** Share their videos or posts on your social media channels, with a brief introduction or summary of the content. This can help expose their content to a broader audience.
2. **Collaborate with them:** Using our social media following or blog, you can collaborate with these content creators by featuring them in your posts or sharing your experiences with their content. This can help build a relationship with the creators and expose their content to your audience.
3. **Tag them in your posts:** When you share their content on your social media channels, make sure to tag them in your posts. This can help them see your support and encourage others to check out your content.

   Overall, supporting content creators who produce beginner-level technical content is a great way to help more people learn and grow their technical skills. By sharing their content and engaging with their channels, you can boost their outreach and build a strong and supportive community of learners.

3. Providing Reminders for Open Source Programs

In the last decade or so, the number of Open Source programs that one can apply to has **increased exponentially**, and keeping track of the same is no easy job. The task is so daunting that even complete repositories like this have been created! But the problem with them is simple: they are static and thus do not provide reminders to the contributors interested in applying to them. To tackle the same, I propose to create a mailing list to which open source enthusiasts can subscribe, and we can issue them regular reminders about the important deadlines of the same programs! We can even use social media handles to share links to these programs when appropriate.

We can also search for data pertaining to the selections, diversity, pay and timelines of all the open source programs that are approaching in the coming months, and create simple visualization for them using projects like RAWGraphs. We can then share them via emails, and also host all of them (along with the data used to generate them) in a separate repository.

Creating a mailing list is an easy task and can be achieved in the following ways:

1. **Choose a platform**: There are several free email marketing platforms available, such as Mailchimp, Sendinblue, or Constant Contact. I have used Mailchimp before and would strongly advocate for it.
2. **Create a list:** Once signed up, we must create a list for mailing subscribers. We can add their email addresses manually (which can be collected by circulating forms and banters in colleges and online) or provide a sign-up form for people to subscribe to the list.
3. **Set up email templates:** I would create a simple email template that can be used for the mailings. This template would include the details of the approaching open-source programs and a brief introduction about them.
4. **Schedule the mailings:** I would set up a schedule depending on the released dates of the upcoming open source programs. This scheduling would be updated on a monthly basis.
5. **Promoting the mailing list:** We can promote it on our social media channels, website, or blog and encourage people to sign up for it. I personally am sure that I can make a list available in all the IITs in India, which should be a good enough starting target audience so that the mailing list can grow organically later.

We would use a third-party service as our mailing service provider, but if the interest in the same grows exponentially (which I hope it really does), we can move on to setting up a local mailing SMTP server on our machines that would do this job for us.

Why am I the ideal contributor?

I am excited to apply for the Google Summer of Code program. It is an excellent opportunity to contribute to the open-source community and gain valuable experience in the field. I have been a member of multiple Open Source communities in the last year, and I can personally vouch for the drastic positive change that the same has brought in my life. Being a woman in India is such a barrier in tech, but it was only due to the open-source community that I could make my own journey and identity as an individual here.

I am passionate about programming and have a strong foundation in various programming languages such as **Javascript, HTML, CSS, and technical communication** in general. I have also completed several projects that involved web development, data analysis, and machine learning.

I am also an active member of **COPS (Club of ProgrammerS)** here in my institute. The whole point of this club is to create a student-led community that helps other students and novices in the technical aspects and provides a shared ground for everyone to upskill themselves. These ideas of shared collaboration and mutual learning later inspired me to pursue OpenSource.

I am interested in contributing to this project under the mentorship of the allotted mentors. I am an avid learner, and I believe that, given the opportunity to be a part of this project, I can help the community positively! I am sure I can complete this project successfully **within the given timeframe**. I am willing to dedicate my time and effort to ensure the project is completed on time and meets the expected standards. I am also open to learning new skills and collaborating with other developers to ensure the project's success.

Thank you for considering my application. I look forward to contributing and working with the wondrous community here. **I have no other commitments for the next 12 weeks** of the GSoC application timeline. I may have institute exams after the semester completion, but the same is over in under a week. I am well versed in academics and can ensure that the same would not hinder my dedication and work schedule.

Detailed Project Scheduling

Week 1-3 **(May 29 - June 18)**	● Creating at least 2 articles/blogs (Technical Content) ● Create the AwesomeFOSS repository and begin to aggregate content for the same ● Create the relevant social media handles ● Work on the creating the reminder list for Open Source programs
Week 4-6 **(June 19 - July 9)**	● Devise a loose posting plan for all these resources on Social Media ● Create a blog post to ask contributors and Open Source enthusiasts to become a part of the AwesomeFOSS repository by sharing their experiences ● Start working on the Scavenger Hunt and creating the required workflows and automation for the same ● Collect data about various Open Source programs, create visualizations from the same and host them on a repository on GitHub
Week 7-9 **(July 10 - July 30)**	● Creating at least 2 articles/blogs (Technical Content) ● Write technical guides as to how I created the scavenger hunt and urge other members to use the same to create their own ● Share about creative coding and visualization projects (like p5.js) from our social media handles

Week 10-12 (July 31 - August 20)	● Work on integrating the blogs and write ups of OSPO contributors with my technical content hosting solution (probably a site on Wordpress or a GitHub repository) ● Conduct a Scavenger Hunt for the new beginners to Open Source
Submission for Final Evaluation of the Standard Coding Period	
Week 13-17 (September 4 - October 4)	● Aggregate content that can be shared on the social media platform (gifs, facts, articles and resources) ● Work on a pipeline/database (drive) of resources that can be used to accumulate all the resources and then quickly post the same to various platforms ● Write an article on creative coding and a technical guide on how beginners can get started with the same
Week 18-20 (October 5 - October 26)	● Work on promoting the content of other Technical Content creators via our social media handles ● Write a technical guide for hosting a hackathon on GitLab as well ● As GSoC timeline would be coming near, I would spend most of the time on finding additional data that can be represented and conveyed with the help of graphics about Open Source programs ● Use the above data to figure out the bottlenecks, as well as the strong points of the Open Source pipeline and ideate on how they can be improved
Week 21-22 (October 27 - November 06)	● Try to gauge the impact this project has had on the reach of OpenSource by analyzing our social media statistics and the various repositories that we created during the GSoC period ● Integrate the blogs of the contributors with long projects into the blogging platform created earlier ● Try to compare the experiences of contributors in Short and Long projects by personally interacting with some of them, and try to create an overview of the same that can be used by contributors next year while applying to GSoC and other related programs ● Write a blog about my complete experience with this project and the impact we had on the community
Project Completion and Final Submission	