GSoC 2023 Proposal

Mentors:

- Mr. Willem Van Iseghem (@canihavesomecoffee)
- Mr. Shivam Jha (@thealphadollar)

Sample Platform

Tarun Arora

Table of contents

GSoC 2023 Proposal

Personal Information

Name:	Tarun Arora
University:	Indian Institute of Technology (BHU), Varanasi
Field of Study:	Computer Science & Engineering
Date of Enrollment:	November 2020
Expected Graduation date:	July 2024
Degree:	Bachelor of Technology
Year:	Junior
Github:	https://github.com/Tarun-Arora
LinkedIn:	https://www.linkedin.com/in/03-Tarun/
Slack:	Tarun Arora
Email Address:	tarun.arora.cse20@itbhu.ac.in
	tarun.arora.030402@gmail.com
Phone number:	(+91) 6307964780
Timezone:	Indian Standard Time (UTC +5:30)

Commitments

- ★ **How many hours will you work per week on your GSoC project?**

  I plan to spend 30 - 40 hours on the project per week.
- ★ **Do you have access to both Windows and Linux for development?**

  Yes.
- ★ **Other Commitments, if any?**

  Yes, I have my summer internship during the GSoC period; June to Mid-July is the overlapping period.
- ★ **Do you plan to apply for any other organization for GSoC'23?**

  I am only applying to CCExtractor for GSoC'23 and have no plans to contribute to any other organization for GSoC'23.
- ★ **If you get selected as a GSoC student, would you like to work on other tasks besides the projects of your choice?**

  I would love to work on other tasks unrelated to my GSoC project.
- ★ **If you are not selected as a GSoC student, would you like to work on the projects as a general contributor?**

  Even if I'm not selected as a GSoC participant, I would happily continue working as a general contributor.
- ★ **Would you like to contribute to CCExtractor long-term after the GSoC program ends?**

  Yes, I would like to contribute to CCExtractor even after GSoC ends.
- ★ **What motivated you the most towards applying for GSoC?**

  Google Summer of Code (GSoC) allows us to gain real-world software development experience and contribute to open-source projects. It allows us to work on challenging projects with experienced mentors, develop their technical skills, and build their professional network. I participated in GSoC'22, with this project only, and had a wonderful learning experience with the mentors.

Introduction To The Project

Sample Platform is CCExtractor's platform for managing tests, uploading samples, running regression tests, and much more. The platform's primary purpose is to test every pull request of the CCExtractor repository with the uploaded samples (which currently have 185GB) and report status.

Technologies used at present

- Sample Platform is built over Python and uses Bash Scripting for its installation on Linux.
- For testing the platform, Python Unit Testing Framework is used.
- The Platform is triggered via GitHub webhooks whenever a PR is created on CCExtractor.
- Rendering of its templates follows the basic HTML, CSS, and JavaScript architecture.
- The sample platform is currently hosted on GCP compute instances.

Scopes of Improvement

The migration of the platform from KVM's to GCP instances, which was carried out during GSoC 2022, has resulted in the resolution of a significant number of bugs in the sample-platform, while also reducing costs.

However, there remain certain bugs and improvements that could be addressed in the current year. These include but are not limited to enhancing the platform's user interface, optimizing its performance, and addressing any remaining technical issues that may arise during usage. Efforts to address these concerns will help to improve the overall functionality and user experience of the platform, and should be undertaken with due diligence:

- Regularly update code with latest stable packages to ensure compatibility, security and minimize bugs. Prioritize regularly maintained packages with large user base.
- Address bugs, including status reporting to GitHub, for improved functionality.
- Implement effective CI/CD to ensure timely code updates on master branch, separate updater from main app for reliable automation.
- Enhance Artifact Download process by addressing issues with expiring Download URLs, such as implementing longer expiration times for improved user experience.
- Introduce a "regression finder" feature that utilizes and functions similarly to git bisect, allowing identification of commits that changed specific outputs in samples.

Why need Sample-Platform?

The Platform helps to manage tests, upload samples, run regression tests, and much more, and report its status for every new version of CCExtractor.
The platform ensures that changes proposed in the pull request are safe and secure. This helps to maintain the stability of the CCExtractor. Sample Platform aims to ensure that CCExtractor and other organizations can use it in their development process.

GSoC 2022 Experience

Participating in GSoC 2022 was a great experience for me. It was my first professional experience and I felt a great sense of responsibility. One of my favorite parts of participating was the opportunity to work on a real-world project and to learn about the industry from my mentor. I also found the emphasis on documentation to be very important and valuable.
The most challenging aspect of participating in 2022 was balancing the workload with my academic responsibilities. It required a lot of discipline and time management, but I was able to stay on top of my tasks with support from my mentor and the community.
Overall, I would rate my experience with GSoC 2022 as excellent. I learned a lot and gained valuable experience that will help me in my future career.
I am also excited to apply for GSoC 2023 as I want to continue to grow as a software developer and contribute to open-source projects. I believe that participating in GSoC again will give me the opportunity to build on the skills and knowledge I gained in 2022, and to work on more challenging projects.
Additionally, I am looking forward to collaborating with the open-source community and to make meaningful contributions to the project I work on.

Implementation of the Project Goals

➔ **Deprecate support for python 3.7 and support python 3.10, 3.11**

Majorly this part of the project is already implemented via this pull request:

https://github.com/CCExtractor/sample-platform/pull/761

This PR implements the following requirements:

- Replace githubpy as it is no longer being maintained
- Replace nose as it is no longer being maintained
- Update unit tests as per the latest implementation
- Update GitHub workflows

I will handle this large PR by dividing them into smaller PRs. This strategy will ensure that the PR is easy to review and that changes can be merged into the codebase quickly.

To prevent breaking changes, I will conduct comprehensive end-to-end testing at each stage of the process. This will help me identify issues early on and correct them before proceeding to the next task.

➔ **Fix windows VM testing**

As of now, windows testing is broken, failing all the tests

For instance see the following test results:

https://sampleplatform.ccextractor.org/test/4325

After debugging I found that the CCExtractor windows build artifact is missing a few .dll files which used to be present earlier.

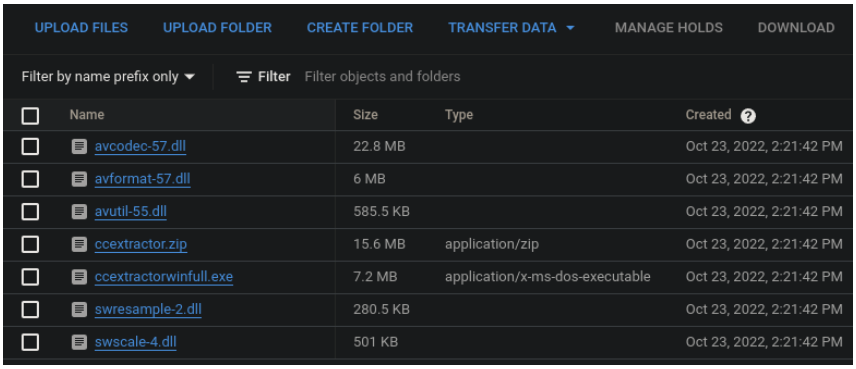
The following DLL files are missing in the artifact:

ucrtbased.dll, vcruntime140_1d.dll, msvcp140.dll, vcruntime140.dll

All these files are from Microsoft Visual C++

I plan to look at CCExtractor commit history and figure out how those missing DLL files can be included in the build artifact again.

Strangely, instead of these dll files, the following files were present in previous commit build artifacts.



➔ **Report errors to test cases whose VM exceeds the time limit**

Currently just before a new test request is triggered, the platform looks at the currently active test VMs, and shuts them down if they have crossed the GCP_INSTANCE_MAX_RUNTIME.
However it doesn't update the test status to Error.

```
def delete_expired_instances(compute, max_runtime, project, zone) -> None:
    for instance in get_running_instances(compute, project, zone):
        vm_name = instance['name']
        if is_instance_testing(vm_name):
            creationTimestamp = datetime.datetime.strptime(
                                    instance['creationTimestamp'], '%Y-%m-%dT%H:%M:%S.%f%z')
            currentTimestamp = datetime.datetime.now(datetime.timezone.utc)
            if currentTimestamp - creationTimestamp >= datetime.timedelta(minutes=max_runtime):
                # TODO: After deleting update the test status to error
                # and remove from gcp_instance table
                operation = delete_instance(compute, project, zone, vm_name)
                wait_for_operation(compute, project, zone, operation['name'])
```

Also if we can configure the VMs to automatically shut down after the GCP_INSTANCE_MAX_RUNTIME after reporting an error that would be better.

➔ **Debug and Fix sample-platform Bot Comments**

Related Issues:

https://github.com/CCExtractor/sample-platform/issues/535
https://github.com/CCExtractor/sample-platform/issues/768

Observations:

★ For the 2nd issue, this is observed to be persistent in Linux tests.
Example:
The following test is completed: https://sampleplatform.ccextractor.org/test/4324
But the GitHub Actions on CCExtractor shows the following:

★ And windows tests are broken at the moment, so for the first issue fixing these subsequent issues is required.

➔ **Fix stuck test runs on the CCExtractor release**

Related Issue:

https://github.com/CCExtractor/sample-platform/issues/574

As the issue description says:

"Whenever CCExtractor releases a new version, a test entry gets added with commit 0000(...) which is invalid, and blocks any test after that."

I have triggered release of my forked CCExtractor, I didn't face any such issue.

But I would try to regenerate this issue again, and update the status of this issue.

➔ **Implement CI/CD for the sample-platform**

To implement CI/CD for the platform we can trigger the following GitHub Actions workflow upon any update on the "master" branch:

1. Generate SSH keys pair and set a private key as an input param.
2. Set Google Cloud credentials that are authorized ssh connection to the VM via GitHub secrets.

We can use this tool to run our own deployment script after ssh into our machine:

https://github.com/google-github-actions/ssh-compute

The script would include the following tasks:

- Force pull from the latest "master" branch
- Copying "runCI", "runCI.bat" and other related files to the bucket
- Writing build commit to "build_commit.py" file
- Restart the platform server via "sudo systemctl reload platform"

With backwards compatibility, if required, we can deprecate the mod_deploy module.

➔ **Improve GitHub artifacts download**

Related Issues:

https://github.com/CCExtractor/sample-platform/issues/683

Implementation:

As I mentioned in the issue, we can upload the arifacts via google-github-actions

Using this would allow us the following capabilities:

1. Fetch the artifacts via signed URLs whose expiry limit can be longer that 1 minute
2. Fetch the artifacts whose GitHub Actions was older than 90 days if required (currently artifacts older that 90 days are deleted)

➔ **Upgrade windows test instance**

For the current windows testing, the VM we use is "Windows Server 2019 Datacenter" whose cost estimates are nearly $63.04 monthly, i.e. $0.08 hourly
We can switch to "Windows Server 2022" whose cost estimates are $29.46 monthly, i.e. $0.04 hourly
This would just require a change in the config.py file, and ensuring that the current startup script does not fail anywhere.
Since the windows testing takes nearly 100 minutes to complete, this would save ~$0.07 per windows test.

➔ **Upgrade to Flask 2**

Related Pull Request:

https://github.com/CCExtractor/sample-platform/pull/682

This would require the migration of the current codebase to Flask 2.

Few common issues I faced while trying to migrate to Flask 2 were:

1. Async Await introduced in Flask 2.0
   Migrating from Flask 1 to Flask 2 can pose a challenge due to the introduction of async/await. Existing code that uses Flask's asynchronous features may need to be modified to accommodate this change, potentially causing issues and requiring significant updates to the codebase.
2. Flask 2.0 added route decorators for common HTTP methods
   Another potential problem when migrating to Flask 2 is that the new version has added route decorators for common HTTP methods. This means that existing code that relies on custom functions for handling these methods may need to be updated to use the new decorators instead.

Timeline

As I mentioned I have my summer internship during the GSoC period, whose June to Mid-July is the overlapping period, I request the mentors to allow me to work upon the following modified timeline:

SHIFTED TIMELINE	OFFICIAL TIMELINE	ASSOCIATED TASKS
April 5 - May 15, 2023	**Pre-GSoC Period** **Community Bonding Period**	Since I have an overlapping internship during the official coding period, I plan to continue contributing during the pre-gsoc period and accomplish the following tasks: • Deprecate support for python 3.7 and support python 3.10, 3.11 • Fix windows VM testing • Report errors to test cases whose VM exceeds the time limit • Implement CI/CD for the sample-platform • Upgrade windows test instance
May 15, 2023 - Jul 14, 2023	**Community Bonding Period** **Official Coding Period**	This is the exact duration of my internship, which coincides with both the community bonding period and official coding period of the program. At present, I am uncertain about the workload during this timeframe. Hence, I presume that there will be minimal work during this period. As a result, I intend to finalize my work during the pre-GSoC period and the additional 15-day extension that I have requested.
Jul 29, 2023	**Mid Evaluations** **(July 14, 2023)**	With the 15-days extension from the official mid-evaluation deadline, I plan to complete the following listed tasks before mid-evaluation:
Jul 29 - Sep 10. 2023	**Work Period**	Post my internship period, I would continue working normally and complete the remaining listed tasks in this proposal
Sep 11 - Sep 17, 2023	**Final week**	I plan to complete the following tasks and submit the final evaluation report: • Debug and Fix sample-platform Bot Comments • Fix stuck test runs on the CCExtractor release • Improve GitHub artifacts download • Upgrade to Flask 2
Sep 18, 2023	**Mentor Evaluations**	Final GSoC contributor evaluations

My Past Contributions to CCExtractor

In late December 2021, I commenced contributing to CCExtractor and subsequently submitted a proposal for GSoC 2022. I was fortunate enough to be selected as a student mentee and have since remained actively engaged with the organization.

Issues Involved:

-
- Bugs Related To Updated Dependencies
    - https://github.com/CCExtractor/sample-platform/issues/606
- Proposal to Implement Dark Theme for the Platform
    - https://github.com/CCExtractor/sample-platform/issues/273
- Bugs Related to foundation up-gradation
    - https://github.com/CCExtractor/sample-platform/issues/620
- Question-Related to setting up the local environment for platform installation.
    - https://github.com/CCExtractor/sample-platform/issues/508

All the issues opened by me are listed here.

PRs created:

Merged:

- Migrate to GCP (GSoC '22)
    - https://github.com/CCExtractor/sample-platform/commit/af14400
- Drop support for python3.7, upgrade to latest python versions
    - https://github.com/CCExtractor/sample-platform/pull/761
- Update GH Actions, remove python 3.7 support
    - https://github.com/CCExtractor/sample-platform/pull/764
- Bug Fixes Related to Installation
    - https://github.com/CCExtractor/sample-platform/pull/612
- Implemented Dark Theme with Toggling
    - https://github.com/CCExtractor/sample-platform/pull/613
- Fixed Foundation Upgrade Issues
    - https://github.com/CCExtractor/sample-platform/pull/621

Others Closed:

- https://github.com/CCExtractor/sample-platform/pull/608
- https://github.com/CCExtractor/sample-platform/pull/624
- https://github.com/CCExtractor/sample-platform/pull/627

All the PRs created by me are listed here.

Why are you the best person to execute this proposal?

Based on the skills and experience you have mentioned, I believe I would be a good fit to execute this proposal. I have a fair amount of experience building applications using Python, working with APIs, and using Git, GitHub, and GCP. This background has given me a solid foundation for executing projects that require these skills. Additionally, my experience over the past year with CCExtractor has allowed me to gain additional knowledge and expertise that I can bring to bear on this proposal. Overall, I feel confident that I could bring a valuable perspective and set of skills to execute this proposal.

Prior Experience

- My participation in GSoC 2022 resulted in the successful completion of the sample-platform project with CCExtractor.
- Over the past two years, I have been actively engaged in web development, initially focusing on frontend frameworks, such as VueJS, NuxtJS, ReactJS, and NextJS.
- During my tenure as a ReactJS Developer Intern with Techlious Network, I had the opportunity to expand my skill set and gain practical experience with AWS services.
- As a full-stack developer with TaggedWeb Inc., I managed concurrent responsibilities for various projects, gaining valuable industry experience.
- In my role as a full-stack developer with Tangli, I developed a Google Meet bot that seamlessly integrates with the Tangli meeting management system and provides analytics for meetings.
- I developed an ASGI web application called Commutify, utilizing frameworks such as Django, DRF, and Django WebSockets to create a real-time communication platform. I am also proficient in NodeJS and have experience working with Firebase.
- I actively contribute to the development of my Institute's Software Development Group's website and have participated in Hacktoberfest for both 2021 and 2022.

Post-GSoC plans

It is highly unlikely that any aspects of the project will remain unimplemented within the 12-week timeframe allotted for GSoC. However, in the rare event that this does occur, I am fully committed to completing any remaining tasks during the extended timeline period. Throughout my participation in the GSoC program, I will strive to make meaningful contributions to CCExtractor and actively engage in community discussions to the best of my abilities. Furthermore, I am eager to continue my involvement with the CCExtractor community beyond the scope of GSoC and remain up-to-date with the latest technologies and ideas. I am enthusiastic about the prospect of providing assistance to the community in any way possible and look forward to building lasting connections with its members.