



Sugarizer Write Activity

Google Summer of Code 2019 - Project Proposal

Mohit Sharma

msharma2@ph.iitr.ac.in

mmatty26@gmail.com

Table of Contents

About me	3
Project Details	4
Why me?	8
Timeline	9
Motivation	11

Personal Information

Basic information

Full name	Mohit Sharma
Github	Scar26
Email	mmatty26@gmail.com
Contact number	7419140386
First language	English
Location	Roorkee, India
Time Zone	UTC+5:30
University	Indian Institute of Technology, Roorkee
Major	Engineering Physics
Current	B.Tech I year (Expected Graduation:2022)

About me

I'm a freshman at IIT Roorkee. I fell in love with programming as a middle-schooler, when I realized that a few lines scribbled on notepad could display my name in a browser.

Over the years, I've been fascinated by python and javascript for the elegance and ubiquity that offer amidst the simplicity. I'm quite passionate about Web Development, Node.JS and PHP being my favorite backend frameworks, and also hold a deep interest in Machine learning and Network security.

I have learnt programming almost entirely by myself. The experience has made me a strong autodidact which would probably prove useful in GSoC. I would love to dig deeper into the codebase and contribute as much as I can. It wasn't until recently that I was exposed to open source, Sugarizer being the first project that I contributed to and I've since been captivated by the beauty of open source.

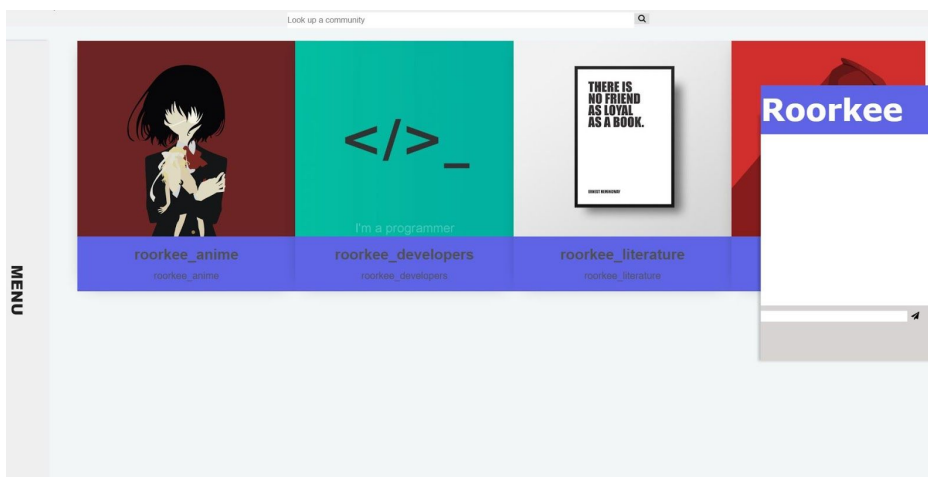
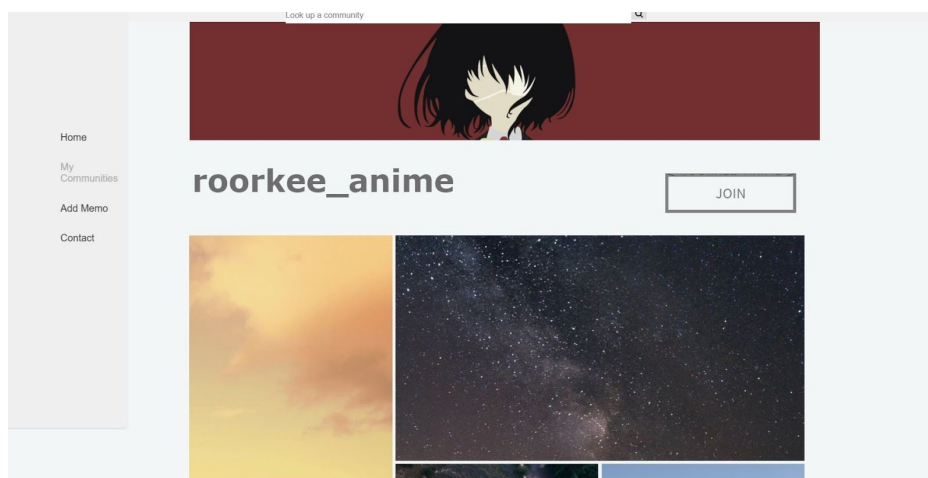
Share links, if any, of your previous work on open source projects ?

I have reported and solved a number of issues on <https://github.com/llaske/sugarizer> as well as added a few enhancements, such as:

- [Pull331](#) (Enabled media exchange in chat prototype via journal)
- [Pull317](#) (Fixed Log redundancy issue in QR code Activity)
- [Pull334](#) (Fixed a code error in documentation)
- [Pull319](#) (Fixed focus issue in Calculator Activity)
- [Pull323](#) (Fixed a front end bug in Chat prototype Activity for older browsers)
- [Issue333](#) (Reported a bug in Shared Notes Activity)

You can have a look at the complete list of my contributions [here](#)

I have also worked on a few projects of my own, such as [Hoodly Desktop](#), a community based social app that focuses on real world interactions. It's written in node.js, using electron js for the GUI, the same frameworks that sugarizer works in.



PROJECT DETAILS

Name of the project

Sugarizer Write activity

Overview

The write activity will be a web implementation of **Write**, one of the major applications provided by the Sugar Learning Platform. It will be written in javascript and included in sugarizer as part of the activity roster. Write activity would aim to provide a concise, web based collaborative text editor for Sugarizer users.

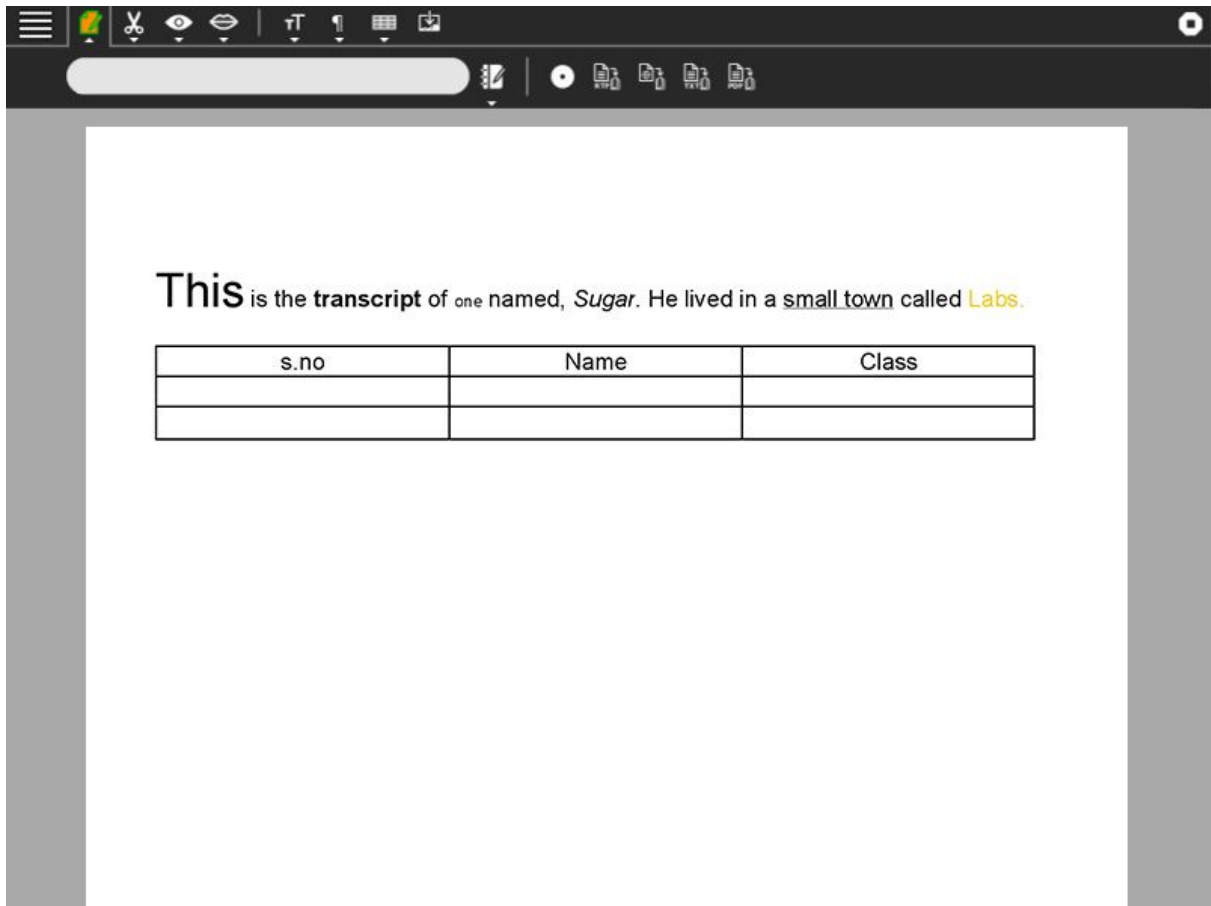
The deliverables of the project will be:

- Text formatting on paragraphs and characters (font change, bold, italic, underline, foreground/background color, alignment ...)
- Extended functionalities like embedded images, hyperlinks, tables and the use of arrays.
- Content will be easily exportable to popular file formats (RTF, PDF, TXT e.t.c)
- Well structured collaborative editing on a document by multiple users.
- Integration of Sugarizer journal to allow the reuse of text and media from other activities
- Real-time Spell check and word suggestions.
- Proper sanitation rules applied on all user input to prevent XSS and Injection attacks.
- Interactive UX while retaining the look and feel of the sugarizer app

Implementation details

- The text editor will be based on QuillJS.
 - I've reviewed a number of API based text editors. My report on the same can be found here:
https://docs.google.com/document/d/1E1n-h0ImBEYnRAiPZIFA62Tqc_xso6EzHjBBDY9Dj8/edit
The main reason that I chose Quill is because of the degree of customization that it offers. I can tailor it to perfectly match the Sugarizer environment. Its functionalities also greatly resemble those of the original python version of Write. Since it has no dependencies, it will be easier to implement and integrate with the rest of the platform.
- Write offers a multi-language text to speech function under the name Alice, the robot. It uses the GST plugin-espeak for this. But since it's a C implementation, it would be hard to include it in Sugarizer. So for this feature I will be using the Speak module that is used in the already existing Speak activity.
 - It offers multiple language support and does not rely on an online backend, meaning users that don't have access to the internet can still use it. This decision was made in consultation with the project mentors.
- Quill's pasteHTML function allows the execution of javascript. Also the content is unfiltered and unescaped. So to prevent XSS attacks, I will be using [Sanitize-html](#) as a dependency to Sanitize all user input.
- For implementing collaboration, I will be using the same model as the Python Write activity, where you can either send an invitation or share it with the neighbourhood. This will be achieved with the presence module that is widely used across Sugarizer activities.
- When the activity is closed, current progress will automatically be saved in the journal. And previous drafts can be accessed from the journal as well. JournalChooser and datastore modules will be used for this.

- Here is the Mockup UI design for the activity. I have tried to keep it as close to the original python activity as possible while adapting it to a browser environment.



I will be working under the guidance of mentors to improve, finalize and implement the UI. For example, I would like to have a keyboard shortcut and context menu item for the speak aloud feature and keyboard shortcuts for a number of other features.

I will also be properly documenting the entire project to make it easy for developers who might be willing to add to the activity.

What technologies will you be using?

I will mainly be using HTML5, CSS and Vanilla Javascript. I will also be using the javascript modules Quill, sanitize-HTML and Talkify for the aforementioned tasks in addition to the standard sugarizer libraries, Presencepalette, Journalchooser and Datastore. I will also be using require.js for the library inclusion so as to match the workflow with other sugarizer activities.

Why Me?

- I have been contributing to Sugarizer for a while and I'm thoroughly familiar with the codebase.
- I have worked with Sugarizer Libraries and integrated modules into activities. [Example](#)
- I have worked with Spell Check algorithms and libraries [here](#)
- I have some experience with other API based web text editors. I used quill for the **Add Memo** feature in [Hoodly Desktop](#)
- I have surveyed and compiled together, the pros and cons of a number of online text editors in order to come up with the best possible approach for the development of Write activity. This list can be found here:
https://drive.google.com/open?id=1ki69wf2GI_Y6HYs7MiNm7NWSs2YecH7MWAvmkJya2pU
- I have worked with the python version of the Write activity before and I'm familiar with its functionalities.
- I'm also used to working as part of a team thanks to Hackathons and other competitions.

Work hours

My summer vacations span from 5th of may to 20th of july. During which I am willing to dedicate 8-10 hours a day to this project. Preferably between 5:30 to 17:30 UTC but my work timings are flexible. Once vacations end, I can still put in 45-50 hours a week but I will be willing to go beyond that if it seems crucial to the project.

Timeline

pre-GSoC period | community bonding

(May 7th - May 27th)

- Interact extensively with the mentors of the project and set up feedback loops.
- Continue to refine the plans and work out the small details of the project in consultation with the mentors.
- Get more experience with the APIs and Libraries that I will be using.
- Get more involved with the community which, after all, is what this period is for.

Week 1-2 | May 27 - June 10

- Refine and finalize the UI/UX design with the mentor's feedback.
- Develop the basic skeleton of the activity.
- I will be starting with the basic template of a sugarizer activity, selecting and integrating the relevant sugar libraries like userSettings and palette.
- Start API integration. QuillJS and Talkify will be implemented in this period.

Week 3-4 | June 11 - June 24

- Integrate journal to work alongside the QuillJS text editor.
- Implement the datastore and Presence modules so backend integration can be started.
- Now that the Text editor and text to speech in place, the UI can also be completed.
- The basic structure is now ready.

Phase one evaluation

Week 5-6 | June 25 - July 8

- Refine the structure further. Document the changes made so far.
- Implement the Collaboration functionality.
- With the Collaboration functionality done, backend integration is now largely completed. UI is also mostly finished.

Week 7-8 | July 9 - July 22

- Minor enhancements, such as the keyboard shortcuts and context menu items that I suggested in the implementation details.
- Complete the UI with feedback and suggestions from the mentors.
- The Activity is now in working condition.

Phase two evaluation

Week 9-13 | July 23 - August 26

- Time for final touches and refinements.
- Add any small features that might've been overlooked.
- Extensive testing and bug fixing period.
- Document in detail, the entire project and add a guide for future developers willing to bring something new to the activity.

post-GSoC period | Finishing touches

Improve the codebase in aspects that will only come to light when integrations are getting merged. Maintain the activity and solve any issues that might be raised by users later on.

Motivation

I was introduced to Summer of Code shortly after I entered University. The idea of working on technology that has a tangible effect on people all around the globe greatly fascinated me.

Skimming through the GSoC archives, I stumbled upon SugarLabs around the same time that I was really getting into Node.js and it was also the first time I got to study the codebase of a real-world application. Every PR I opened, every change I committed taught me something new. Node.js is now my favorite backend framework (as mentioned in the introduction).

Apart from the learning opportunity, what SugarLabs stands for is also something I really admire. I've been actively involved with the education of underprivileged children as part of the JEE cell under the NSS initiative in my college. So the impact SugarLabs aims to make on the world is something I really appreciate.

Overall, I feel like Google Summer of Code, especially with SugarLabs is an unparalleled platform for me to make my summer worthwhile, and utilize it to learn and grow.