


Google Summer of Code 2020

Sugar Labs

Resolve 100 issues in Music Blocks

BASIC INFORMATION:

1. Personal Information

- Name: Aviral
- Github: [aviral243](#)
- Email: aviralgangwar24@gmail.com
- Resume: <https://bit.ly/aviralGangwar>
- Phone No. +91-8958539517
- IRC Nickname: aviral_243
- Location: Roorkee, India 
- Time Zone: India (UTC +5:30)
- Linkedin: [Aviral Gangwar](#)
- First Language: My first language is **Hindi** and I am proficient in Speaking and Understanding **English**.

2. University Information

- University: Indian Institute of Technology, Roorkee
- Majors: Electrical Engineering
- Current: II Year (expected graduation in 2022)
- Degree: Bachelor of Technology (4 Year Program)

3. Contact and Working hours

- Reachable anytime through **email** or **contact number**.

- Typical working hours include:-
 - i. UTC 0330 - 0730 hrs (IST 0900 - 1300 hrs)
 - ii. UTC 1030 - 1330 hrs (IST 1600 - 2000 hrs)
 - iii. UTC 1630 - 2030 hrs (IST 2200 - 0200 hrs)

CODING SKILLS:

Programming Languages

- Sound Knowledge of Javascript and it's frameworks (React, Node), Typescript, PHP, C++, HTML/CSS
- Moderate knowledge in Python, Ruby, and Solidity.
- **Web Frameworks:** React, Redux, Node, Django, Django REST, Flask
- **Libraries:** Socket.IO, JQuery, Bootstrap, Semantic UI
- **Databases:** MySQL, PostgreSQL, MongoDB
- **Utilities:** Docker+Compose, CI+CD Pipeline, Postman, Jupyter, Firebase, MongoDB Atlas

Development Environment

- Ubuntu 19.10
- VSCode as IDE supported by a range of extensions.
- Linux Shell
- Chrome Dev Tools
- Git for version control

Apart from the technologies listed above I have sound knowledge of **Object-Oriented Programming**, architectures like **MVC**, **MTV** and **MERN/MEAN** stack.

My other interests include **Information Security**, **Blockchain**, **DevOps** and **Competitive Programming**.

ABOUT ME:

I am a 20-year-old sophomore, currently enrolled under the **Electrical Engineering department at IIT Roorkee**. I was introduced to programming concepts during High school. I developed a passion for web development in my first semester and since then most of my time went into learning new technologies and developing software using them. I have **been contributing to open-source regularly since about five-six months** now.

I have experience working closely with a team as I have been an active member of the Web Development team of [E-Cell IIT Roorkee](#) and [Students' Club IIT Roorkee](#).

My projects:

- [Chat-App](#)
 - A real-time chat app. Users enter their username and the room they want to join. Users in the same room can **view and send messages to each other**.
 - It uses [Socket.IO](#) along with a [Node.js](#) backend to enable real-time, bi-directional communication.
- [DevConnector](#)
 - A social app to connect Developers. **Users can register and view other users, post online, like and comment**, and have a look at other users' public Github repositories.
 - Uses the **MERN[Mongo-Express-React-Node]** stack for implementation.
- [Utthan Foundation](#)

- Utthan Foundation is a **trust for rural downtrodden women**. It empowers them by providing self-employment options.
- The website's **frontend comprises** [ReactJS](#) and [Sass](#) for styling. It also features a custom [Webpack](#) written from scratch and [Google Sheets API](#).

I am not listing all the projects here. My other projects can be viewed on my [Github profile](#).

Open Source Contributions:

- Sugar Labs

I have been **working with Sugar Labs for the past 2 months**. The statistics of my contribution (at the time of writing) to **Music Blocks** are as follows:

- **Pull Requests:** 19 (**16 merged**, 1 closed and 2 open)
- **Commits:** 50 (1633++, 770--)
- **Issues:** 8 (7 closed and 1 open)

Pull requests:

- ❖ [#2061\(merged\)](#): Not setting the display to none on duplicating
- ❖ [#2071\(merged\)](#): Music Blocks doesn't go into infinite loading animation on dropping files
- ❖ [#2078\(merged\)](#) : Removed ESLint file
- ❖ [#2079\(closed\)](#): Documentation update after block addition code refactoring
- ❖ [#2080\(merged\)](#): Documentation update after block addition code refactoring
- ❖ [#2083\(merged\)](#): Migrating help strings from turtledef.js
- ❖ [#2088\(merged\)](#): Disable Keyboard input while MB loads
- ❖ [#2094\(open\)](#): Change mode to beginner/advanced without refresh
- ❖ [#2107\(open\)](#): Redo button implemented
- ❖ [#2109\(merged\)](#): Error message displaying on decreasing note value below 0

- ❖ [#2111\(merged\)](#): Fix help modal position
- ❖ [#2120\(merged\)](#): Implemented browser check to prevent MB from stopping on Firefox
- ❖ [#2122\(merged\)](#): Hardwired position parameters for help div
- ❖ [#2138\(merged\)](#): fix official app repository link in plugin readme
- ❖ [#2146\(merged\)](#): subtract one block added
- ❖ [#2149\(merged\)](#): Fix subtract block SVG
- ❖ [#2158\(merged\)](#): Conditional rendering of like button tooltip in planet
- ❖ [#2163\(merged\)](#): Feature: Cycling through block help after introductory help
- ❖ [#2171\(status is closed but changes were merged\)](#): Added find button and advanced block indication

Issues:

- ❖ [#2077](#): Browser shows audio play symbol even after hitting the stop button.
- ❖ [#2086](#): Documentation changes after #2082 got resolved
- ❖ [#2087](#): Music Blocks not opening due to null value in voicename block.
- ❖ [#2091](#): Weird on hover animation of buttons
- ❖ [#2110](#): Help modal gets covered by Aux menu
- ❖ [#2121](#): Help div not getting centered on first time use
- ❖ [#2137](#): Official App Repository link is broken inside plugins/README
- ❖ [#2157](#): Conditional rendering of like button tooltip in planet

Commits:



([Music Blocks](#) contributors statistics for the time frame Jan 19, 2020–Mar 24, 2020)

- **Rocket.Chat**

Rocket.chat is an **open-source team chat platform**.

Pull Requests:

- ❖ [#16122\(merged\)](#): Add Cloud Info to translation dictionary
- ❖ [#16232\(open\)](#): Set x-content-type-options header to nosniff
- ❖ [#1549\(open\)](#): Added steps in quick-start readme

- **Open source Pull Requests as part of Hacktoberfest**

Hacktoberfest is a **month-long program to promote open source** contributions.

Pull Requests:

- ❖ [#32\(merged\)](#): Add blurb explaining what is React
 - ❖ [#7\(open\)](#): Minor changes to Readme
 - ❖ [#275\(closed\)](#): Add clap by seventeen
 - ❖ [#1266\(closed\)](#): Upgraded client and admin to react-scripts 3
-

PROJECT DETAILS:

Title: Resolve 100 issues in Music Blocks

Coding Mentors: [Walter Bender](#), [Favour Kelvin](#), [Sumit Srivastava](#), [Vaibhav Aren](#)

Assisting Mentors: [Jaskirat Singh](#) and [Devin Ulibarri](#)

Introduction:

There are 200+ open issues on Music Blocks. Some are trivial and some will require some major effort to resolve. But a dedicated effort could probably tackle half of them -- on average one a day -- over the course of GSoC.

Synopsis:

Music Blocks currently has 206 open issues (at the time of writing). While working on Music Blocks I've divided the issues in the following classification:

- Trivial issues/ Issues that require little to medium effort and time.
e.g. [#2157](#)
- Some open issues that have been fixed / not relevant anymore.
e.g. [#1211](#)
- Issues that require refactoring of code of certain files.
e.g. [#2053](#)
- Issues that require major effort to solve / prospective GSoC projects.
e.g. [#1963](#)

In my approach I aim to traverse that list top to bottom i.e. Start with resolving trivial issues and identifying non-relevant or fixed issues and closing them, gradually moving towards issues requiring refactoring of certain files and dedicating the later phase of the

program to an attempt to solve major issues. Meanwhile also countering any new issues that pop up during the duration of the program. This methodology will ensure that we're able to solve a large number of issues rather than getting stuck on a single complex issue going nowhere.

Project Plan:

Goals:

- Resolve nearly 100 issues in Music Blocks.

The Problem:

Music Blocks has been undergoing development for the last five years and recently there has been significant development both as part of GSoC / GCI programs and contributions from other collaborators. We are gradually moving towards better code management and adding necessary features.

One thing that certainly brings down the whole experience is the number of issues that are lying around. The issues range from CSS fixes to revamping features of a block. Some issues have been around for years now.

There is not a single massive issue that is addressed in this project. It's about eliminating a number of mini issues and making the experience wholesome.

A structured approach along with support from mentors and collaborators can significantly bring down the number of issues in turn greatly improving the experience of the awesome project Music Blocks is.

The Solution:

The project will mainly address the above-mentioned problems. I have given a small glimpse of how I would like to tackle them elegantly in the Proposal abstract. I'll emphasize over that here to clear our vision on how we wish to approach:

- **Trivial Issues**

This is the classification where I wish to put issues that do not go into the other three defined categories. This classification, in particular, is a little vague. I mostly wish to incorporate issues that look solvable and the effort involved is moderate. Apart from that, we are clear on the design and implementation of the feature. In essence, what is required is work and anything else is not a hurdle in solving the issue.

- **Some open issues that have been fixed / not relevant anymore / need more info:**

While working on Music Blocks I realized that there are a number of issues that:

- Have been fixed but are still lying open as they weren't closed when the fix to them was merged.
- Issues that aren't relevant anymore (e.g. Issue pertaining to a feature that has been removed or revamped)
- Issues that we've figured a workaround for and thus need not to be addressed anymore in their present state.
- A number of issues that are labeled [WF1-Need more info](#). Some of these issues were a one-time thing that generated due to some unexpected behavior on the user's system and has nothing to do with Music Blocks.

To better support, my hypothesis here are some examples of such issues:

- [#325](#): Scale Image if it exceeds a maximum size in activity.js

This issue was open even though we had a workaround for it in place already.

- [#1245](#): Downloading My Project.html without clicking save

This is another example of an issue that was already fixed but was lying open.

- [#1287](#): Tell user when a newer version is available

This issue was also open for a couple of years even though it was not worthy to implement.

I aim to identify all such issues and close them. This would give a better idea of how much work is needed and where.

- **Issues that require refactoring code of certain files:**

Since Music Blocks is constantly expanding and new features are being added, it has made some files monolithic and difficult to manage. This has resulted in the following complications:

- Generated some new issues that are now difficult to solve due to improper code management
- Made it difficult to implement a new feature due to side-effects
- Led to a lot of repetitive code in some files

Although there is already another GSoC project floated for Model-View-Controller refactoring of the entire codebase and if done it would counter this issue pretty much. Then we can work on solving those issues or implementing the features.

If however even after MVC some files remain as such, I aim to properly refactor them first and then tackle the issues/features.

Examples of such issues are:

- [#2053](#): Star in auxiliary menu should change state on click

There have been a lot of attempts at implementing this feature but [palette.js](#) code in its present state is pretty brittle and doesn't allow this feature as of now.

- Lots of repetitive code in [js/widgets/help.js](#):

I recently implemented the [Block Help](#) feature and after implementation, I felt that it led to almost similar copies of a function with slightly varying functionality.

I aim to work on refactoring files like these to **resolve current issues** as well as **suppress further issues that can generate** from them.

- **Issues that require a major effort to solve / prospective GSoC projects:**

There are a few issues that if solved add great functionality but the complexities and work involved in them are gigantic. Some can even be labeled separately as a GSoC project.

Instances of such issues are:

- [#2058](#): Scale Degree Design Path Proposal

This has in fact been floated as one of the projects for Music Blocks as the work involved is considerable.

- [#1963](#): **Git-based Backend**

Again this is a large scale issue requiring proper design review and work since a lot of functionality needs to be added.

- [#2020](#): **Scale/Mode Meta**

As it has been mentioned in the issue itself that it can be a potential GSoC project. The issue has been broken down and explained beautifully by [Devin Ulibarri](#), but still, it'll develop more as we start working on it.

In my approach, I have thought of two methodologies for issues of this scale:

1. Either **tackle them towards the end of the program duration** when we've already solved a considerable number of small issues.
2. Pick one of them that provides the most utility and **work on it alongside other issues**, dividing time. By the end, we'll be able to **implement at least one** of these several issues and that would be quite an accomplishment.

I am more sold on using the second approach as there are some issues in there, I seriously look forward to working upon it because the functionality they add is beautiful.

- **Occasional issues that require a design review to get resolved:**

Since Music Blocks is a pivot between two distinct concepts: Music and Code, there are a bunch of **issues that need design review** and further since Music relates to learning and

Music Blocks is, in fact, being used to teach children, certain **issues have pedagogical implications** up for discussion.

Example of such issues are:

- [#518](#): A "for" block addition to the flow palette
- [#1949](#): Way (for teachers) to lock blocks
- [#1363](#): Do we want "compile" as a block?

The prerequisites discussed above for issues of this kind are best developed when they're lined up for discussion. Therefore, in my approach, I aim to engage not only the mentors but the entire Music Blocks (and Sugar Labs) community and then move forward with solving them.

Timeline:

The nature of this project is as such that we can't have hard and fast parameters of work done by the end of a deadline. Also, it's **difficult knowing beforehand which issues** will certainly get solved. But I've still attempted to prepare a **list of issues** I'll be working upon:

Phase 1:

<u>Timeline</u>	<u>Issues No.</u>	<u>Title</u>	<u>Comments</u>
<u>Pre GSoC</u>	---	---	Mostly get this list in accordance with mentors. Discuss the design aspects of a few issues. Refactor a few files if needed.
Week 1	#291 #324 #354	<u>Add plugin selector</u> <u>Add Plugin Support Stop Support</u> <u>Grab files from a local server in pluginsviewer.js</u>	There are a lot of issues surrounding the Plugin Viewer. I look forward to resolving by defining the pluginViewer.js

	#355 #378 #1243	<u>this.pluginFiles.sort() in pluginsviewer.js</u> <u>Add plugin support for RUN</u> <u>Plugins broken on Chromium</u>	<p>There is already a significant amount of work done already on PluginViewer but it's not complete. Hence not implemented yet.</p>
Week 2	#747 #1360 #1073 #1463 #1185 #746 #663	<u>Down/Up arrow on keyboard scrolls screen in FF (and Left/Right does nothing)</u> <u>Pressing down arrow in FF scrolls the window down (until nothing is visible)</u> <u>add aliases for block search</u> <u>Make middle mouse scroll more sensitive</u> <u>Publish button should use Wait cursor</u> <u>Copy should allow some degree of mouse/trackpad "wiggle"</u> <u>Complete Software Keyboard by Piggeh</u>	<p>Arrow key functions have been defined but I've often felt they do not serve the purpose right. While addressing these 2 issues I look forward to redefining the way arrow keys work. This week is mostly for all keyboard/mouse-related issues.</p>
Week 3	#819 #1059 #1101 #1194 #1900 #1103	<u>support for music as it relates to learning</u> <u>Videos (Documentation)</u> <u>Contributor Guidelines</u> <u>Need to make help shown more interactive</u> <u>Update Musical Keyboard Documentation</u> <u>Dynamic Inline examples</u>	<p>I've grouped together all the help and documentation related issues. Will try to wrap them up in the initial phase.</p> <p>For making help feature more interactive I wish to incorporate intros if it goes well with Music Blocks.</p>
	#791	<u>Repeat for Lilypond Output</u>	

Week 4	#2039	<u>Better Handling of Forever Loop for Lilypond Export</u>	Lilypond exports have been facing a lot of bugs lately. Some of these issues are concerned about making Lilypond export more efficient while some target the existing bugs or abnormal behavior.
	#782	<u>lilypond should not use abbreviations for custom names</u>	
	#2040	<u>Peculiar Tie Output Lilypond</u>	
	#2035	<u>Lilypond: Staccato inside slur is not exporting properly</u>	
	#2036	<u>Bad Lilypond Output: Forensics Requested</u>	
	#1352	<u>Characters confusing Lilypond</u>	
	#781	<u>lilypond output is not accurate when using duplicate block</u>	

Milestone Reached: Phase 1 Evaluation

I look forward to resolving these issues until the phase 1 evaluation. The list is **tentative and not exhaustive**.

Additional: At this point in time I'd like to pick an issue either [#1963](#) or [#2020](#) and kickstart the work on it, continuing it throughout the program as part of my approach of taking one large scale issue and working on it alongside.

Phase 2:

<u>Timeline</u>	<u>Issues No.</u>	<u>Title</u>	<u>Comments</u>
Week 5	<u>#1979</u> <u>#1855</u> <u>#1882</u> <u>#2157</u> <u>#1947</u> <u>#1853</u> <u>#1211</u> <u>#1655</u> <u>#1632</u>	<u>Add an optional UID when accessing the Planet</u> <u>store UID in projects on Planet</u> <u>Tag Suggestions for Planet Publish</u> <u>Conditional rendering of like button tooltip in planet</u> <u>Drop and Load Project Feature Available in Planet View</u> <u>Upload and download functionality in Planet does not seem to work</u> <u>Disable "Save as PDF" when running offline</u> <u>Projects vanish if you select 2 categories</u> <u>"Like project" button is not working</u>	There are a lot of micro enhancements suggested for Planet mode. In Week 5, I plan on addressing issues related to the Planet.
Week 6	<u>#2052</u> <u>#1432</u> <u>#518</u> <u>#1694</u>	<u>Enable scrolling in palettes menu</u> <u>Add zoom option to the Palette</u> <u>A "for" block addition to the flow palette</u> <u>Position of the search bar, home, show/hide block icon, expand/collapse block icon,</u>	This group mainly contains all palette and toolbar issues. Issues related to

	#1966 #1936 #2143 #1965 #2053	<u>increase/decrease icon</u> <u>Update "Run step by step" button</u> <u>Top Menu Disappears</u> <u>Wrap graphics(as a setting on toolbar)</u> <u>Redo Button</u> <u>star in auxiliary menu should change state on click</u>	palette enhancements and toolbar buttons / UI are grouped together.
Week 7	#376 #756 #1316 #1949 #719 #1919 #1244 #1363 #1873 #568 #1891	<u>Are there return or arg blocks in blocks.js</u> <u>More elegant way to show time elapse in Rhythm Ruler</u> <u>Q: Do we want "subdivide"?</u> <u>Way (for teachers) to lock blocks</u> <u>"factor of" block</u> <u>Blocks in macros briefly appear in upper left of the screen</u> <u>Tie as a block similar to v space?</u> <u>Do we want "compile" as a block?</u> <u>A forgiving broken 'pitch' block</u> <u>Sing Block?</u> <u>`Note Counter` and `Sum Note Values`</u>	A lot of additional blocks have been proposed over time. They're all grouped here. For all block addition issues, I first aim to find out if the block is really needed or not and implement accordingly. Other issues are also related to blocks in general like locking feature or UI enhancement.
	#1153	<u>Problems with adding multiple filters</u>	

Week 8	#1887	<u>tweenjs.min.js and other "Violations" in Console</u>	Music Blocks has had its share of performance issues. Most issues here are related to unresponsiveness or slowing down. There is also an idea of implementing Performance benchmarks and removing console warnings/violations.
	#1262	<u>Loading a project with many blocks multiple times slows MB down quite a lot</u>	
	#1261	<u>Better User experience for loading projects with many blocks</u>	
	#1260	<u>User-experience slowed when opening a different project after closing project with many blocks</u>	
	#1501	<u>MB gradually slows down and eventually unresponsive for long time use</u>	
	#1032	<u>Implementing Stress Test/Benchmarks?</u>	

Milestone Reached: Phase 2 Evaluation

This is the work I propose by the end of Phase 2. Some issues might go out and some might go into that list.

Additional: Continue work on one of [#1963](#) / [#2020](#)

Phase 3:

<u>Timeline</u>	<u>Issues No.</u>	<u>Title</u>	<u>Comments</u>
Week 9	#2126	<u>Bottom part of Widget window not visible</u>	<p>This section is purely dedicated to UI/UX issues. Some issues address bugs while others question the present UI of a block/widget.</p> <p>Some UI issues are also present in other sections pertaining specifically to that section. Here I've included general UI issues.</p>
	#855	<u>Drag to edge of screen with "enable scrolling" on should scroll the screen</u>	
	#1510	<u>MB often gets into a state where only half the screen (left side) initiates vertical scroll</u>	
	#2172	<u>Tour window imperfect positioning</u>	
	#1143	<u>UI issues with timber -> filter</u>	
	#1144	<u>Cannot see the left<-->right scrollbar at bottom of PTM (unless up<-->down is at lowest)</u>	
	#1276	<u>missing characters on Windows?</u>	
	#1675	<u>Resizing MB causes thumbnail to misbehave</u>	
	#2173	<u>Tour window fullscreen problems</u>	
Week 10	#692	<u>Improvements in Timbre widget's interface</u>	This section is for widgets. Issues related to an individual widget or something that applies to all of them. Saving widget
	#2051	<u>Repeats Skipped in Phrase Maker</u>	
	#1872	<u>Solfège in Phrasemaker with Scalar Step should generate Solfège</u>	
	#2017	<u>Saving for Widgets</u>	

	#1589 #1934 #1932	<u>Disappeared grid lines in Pitch Time Matrix</u> <u>Infininote in Phrase Maker</u> <u>Cell Improvement for Grids (Phrase Maker, Keyboard)</u>	state is probably the pick of them, others are small enhancement issues or bugs.
Week 11-12	#1107 #2174 #1860 #1818 #574 #1973 #1898 #1842 #1767 #1291 #1877 #1603 #1982	<u>Row Row Row your Boat in 12/8 meter</u> <u>Block Pie Menu position doesn't change with scrolling</u> <u>musickeyboard only saves default synth</u> <u>synth length</u> <u>Play Chunks Backwards</u> <u>G 5 plays too short</u> <u>Combination of `On strong beat do` and `On every note do` not working</u> <u>bug in 120 beats per minute, 1/12 beat value</u> <u>missing Meter Widget features</u> <u>A (JS) way to choose .svg from our mouse art folder (without choosing from desktop)</u> <u>modifications to the grid view breaks playback</u> <u>Scale resetting itself automatically</u> <u>Error when measure is incomplete (XML)</u>	Miscellaneous Issues. Addressing issues that do not relate to each other too much towards the end, one at a time.

	#2165 #2162 #2013 #1388 #1213 #1691 #1515	<u>updated pitch values not reflected in grid playback</u> <u>Set Drum Overrides Set Instrument</u> <u>collapsing notes with embedded notes is broken</u> <u>all wet?</u> <u>When browser loses focus, the blocks shuffle back to a default</u> <u>Add more music instruments</u> <u>Two Recommendations for saved .html file</u>	
Post GSoC	#1963 / #2020	<u>Git-based Backend/Scale / Mode Meta</u>	Whatever work is done until now on this issue alongside other issues, I wish to take it forward after GSoC.

Milestone Reached: Final Evaluation

This is a rough list sketched by me which will serve as a guideline for working. As there's always more to do that is proposed, I will not restrict myself to this static list rather try to be more flexible in my approach, doing what is necessary during the program. My commitment to my goals will be unquestionable.

How many hours will you spend each week on your project ?

Due to the COVID-19 outbreak, my Institute has shut down unexpectedly and currently, we don't have clear notice of vacations so I can't provide the exact dates.

For what it's worth, they were scheduled to take place from 9th May 2020 - 12th July 2020.

During vacations, I can easily devote 50-55 hours per week until my college reopens and 40-45 hours per week after that. I am also free on weekends.

Other than this project, I have no other commitments or vacations planned for the summer. I shall keep my status posted to all the mentors and community members on a weekly basis and maintain transparency in this project.

If you will be off-the-grid for a few days, then mention those in the timeline.

Since I am in my sophomore year, I don't have any commitments during the period and so I will be available for almost all of the time frame. Furthermore, I do not have any planned vacations or other engagements.

How will you report progress between evaluations ?

In between evaluations, I am reachable anytime through Email, IRC, Slack or a well-planned video session if required.

Further, the nature of this project is as such that I'll be continuously working on Github issues, interacting with one or more mentors. I'll let them know of the progress then and there.

I also plan to write weekly Medium articles discussing my progress with the project. My Medium username is [@aviralgangwar24](#). This would be my first time writing a technical blog post.

Discuss your post GSoC plans. Will you continue contributing to Sugar Labs after GSOC ends ?

I do plan to keep contributing to Sugar Labs after the GSoC period because I believe that since the web is a continuously evolving place, so are its needs. Thus there will always be instances where my project will need to be worked on to have more features added to address an ever-growing populace and to cater to its needs.

Furthermore, this will also give me the opportunity to put my skills into practical use and give back to the community which I wholeheartedly want to keep doing through the platform of Sugar Labs.

Till now, the highlight of my experience with Sugar Labs has been the active involvement of the mentors. With the community growing continuously, I feel responsible for the projects I contribute to. Having picked up a lot of development skills, my major focus after GSoC would be to enhance my mentorship skills so that I can give back to this community by helping other people navigate around and hope to mentor future GSoC/GCI students.

Looking forward to contributing this summer to Music Blocks !
