

Sugar Labs GSoC 2021 Proposal

Joykirat Singh

Github: [joykirat18](#)

Email: Joykirat Singh

CV: [Link](#)

Town, Country: Delhi, India

About Me

I am a 19-year-old sophomore student studying at Indraprastha Institute of Information Technology (IIIT) Delhi, pursuing B.Tech in Computer Science and Design.

I have been coding since the start of my college days and have taken multiple design courses. I am well versed in working with Git. I have adequate experience in javascript (ES6), react, typescript and am a music enthusiast. I have been exploring the musicBlocks projects and looking at all the features it has to offer. During this time, I have made some PRs. My past contribution to musicBlocks:

1. [#2774](#) BugFix and refactoring in modewidget.js
2. [#2753](#) adding ES6 classes to widgets/meterwidget.js
3. [#2763](#) Porting ES6 widgets/rhythmruler.js
4. [#37](#) adding test cases to Temperament.test.ts
5. [#2778](#) Refactor: widgets/pitchstaircase.js - lint, prettify
6. [#2906](#) sampler.js: Linting and Prettify
7. [#41](#) adding test Cases to scale.test.ts

Most of my pull requests are porting to ES6 and linting, but during this process, I tried to understand the functionality of major functions and test them thoroughly when I was porting them to ES6. Therefore these PRs have helped me understand the depth and working of musicBlocks. Additionally, I have also participated in [discussions-39](#) to suggest some wireframes for the new musicBlocks and how we

can improve the design so that new users can easily navigate the blocks and other features.

As musicBlocks-v4 will be written in React and Typescript, I also have experience in React and have done some exciting projects. I recently worked on creating my portfolio, where I used React and design skills that I learned in my college courses to make the website user friendly. ([Link to Website](#), [Link to GitHub Code](#)).

Furthermore, I also spent time in Feb-March brushing up on my concepts of typescript. I am very interested in working with musicBlocks so that I can learn more and grow.

Moreover, being a design student, my design skills like making wireframes, the information architecture followed in websites, grid systems, evaluation of websites will help in redesign of the whole application to fix its UX issues.

Some of my design projects are :

1. [MoneySpin](#): Worked to create a UI for an application that helps people cope with their gambling addiction and channel their energy to invest in stock markets through games.
2. [Sahayak](#): Worked to redesign the FMS service application(College's hostel staff management) such that it works in an organised and effective manner.

Project Title: MusicBlocks Menus and Palettes

Overview

As MusicBlocks is being refactored, this gives us an opportunity to revisit several UX issues, including designing and implementing the menu bars and block palettes.

I would like to design the wireframes of the new musicBlocks UI and implement them in React + Typescript so that all UX issues are resolved, and users can easily navigate through the palettes and menus.

Checkpoints for this project

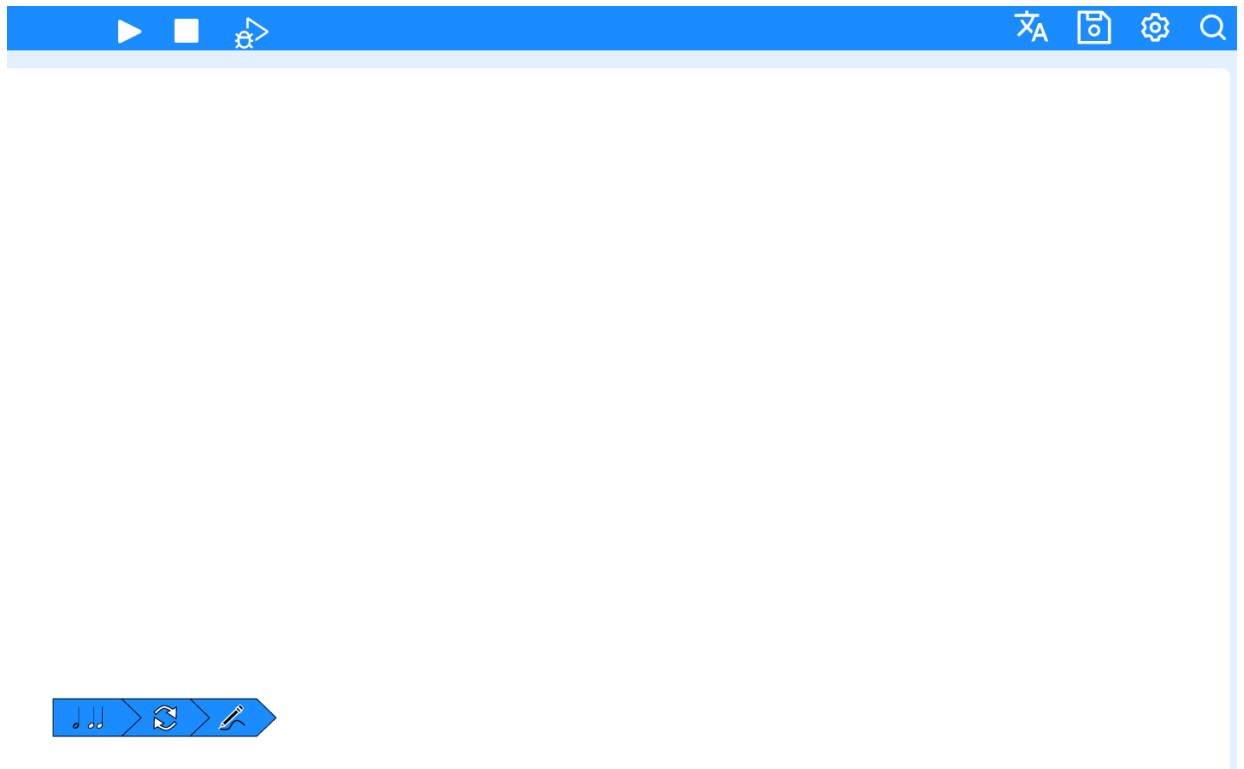
Wireframe

To design the menus and palette in musicBlocks, the first step is to finalise the wireframe ideas, so I build upon some wireframes that Mentor Anindya Kundu has submitted in the discussion tab. For creating the UI, we will be using Material UI.

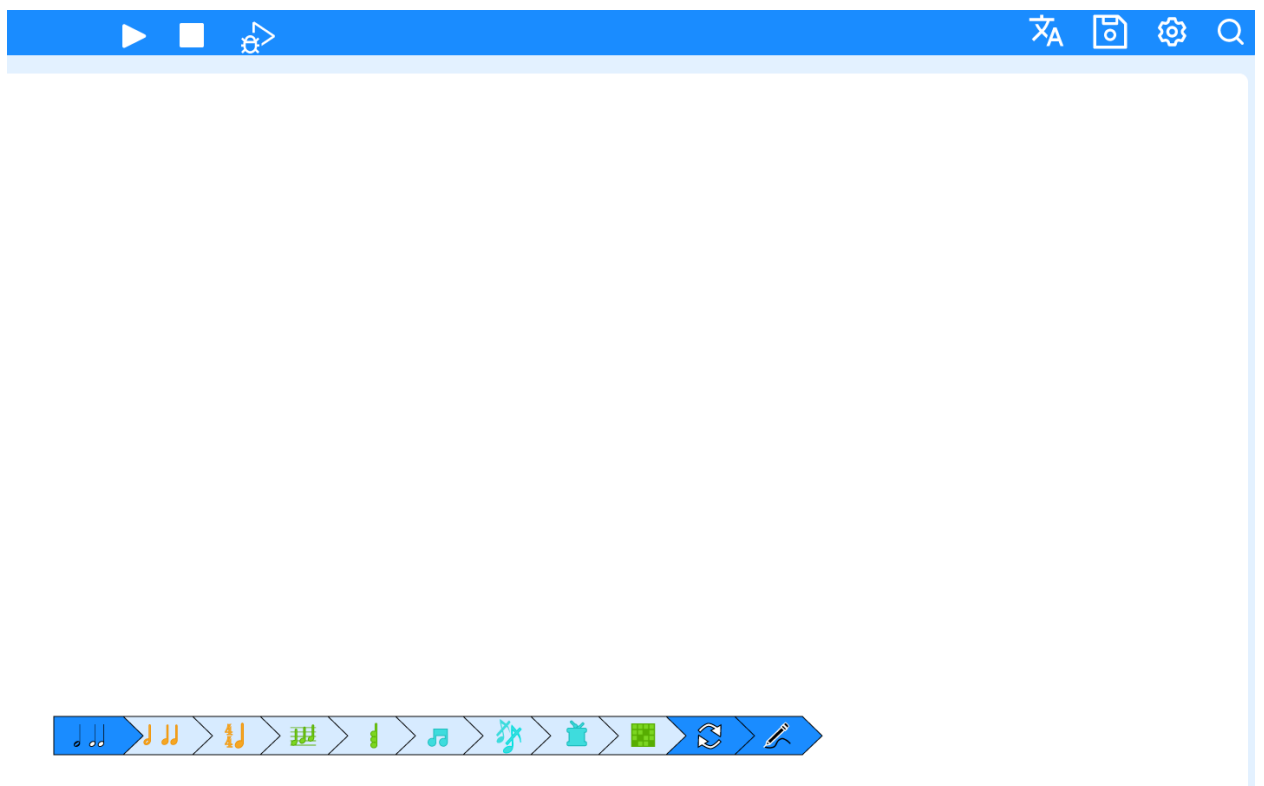
```
$ npm install @material-ui/core
```

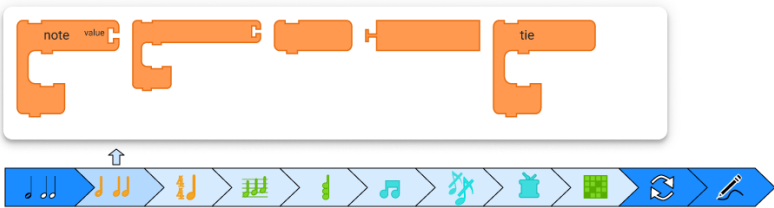
1. Palette

For the palette blocks, we need to design the wireframes to optimally use the space on the screen's narrow side. Therefore, I placed the palette at the bottom left side of the screen, which expands to cover the bottom screen horizontally. In this way, the design will stay minimal and make a lot of room in the canvas to develop some fantastic musical tunes. Here are some of the screenshots of the implementation.



Clicking on any of the three sections (<...>) opens up the categories of blocks. On clicking on any of the categories, the corresponding blocks open up, and the user can drag them onto the canvas.





Video demonstrating the above video. ([Link](#))

2. Music

There are quite a few event buttons in musicBlocks. Some of the buttons are frequently used and some rarely. So I have decided to divide the buttons into two categories for the menus, low shelf (always at hand) and high shelf (a few clicks away). Low shelf button will comprise of the buttons which are frequently used by the users, while high shelf buttons will be a few click away.

- **Low Shelf Buttons :**

Play, Stop, collapse, clean, Grid, show/hide block, expand/collapse block, increase the block size, decrease block size, search button.

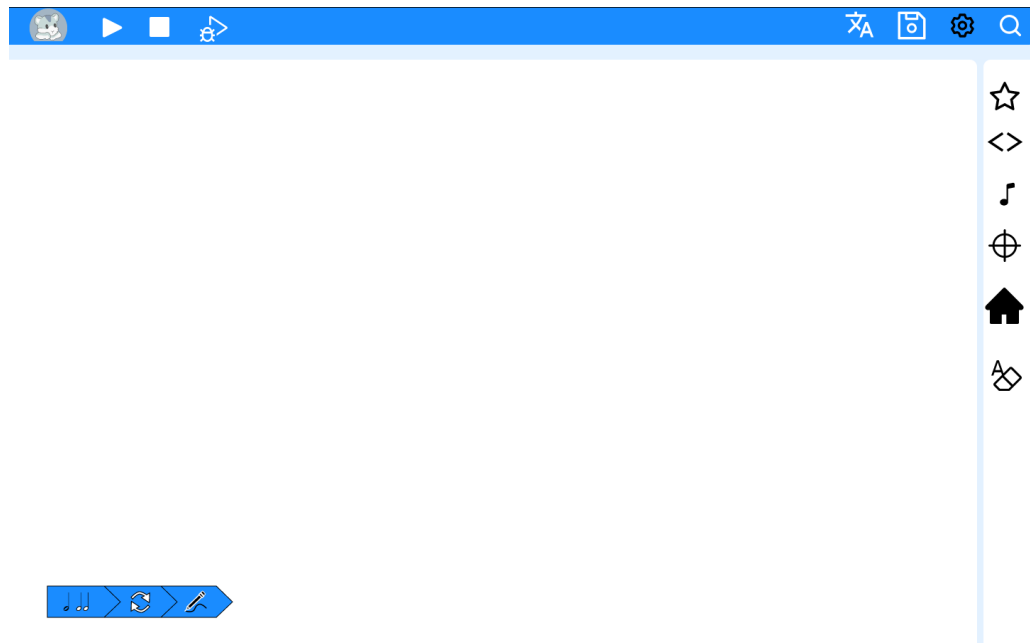
- **High Shelf Buttons :**

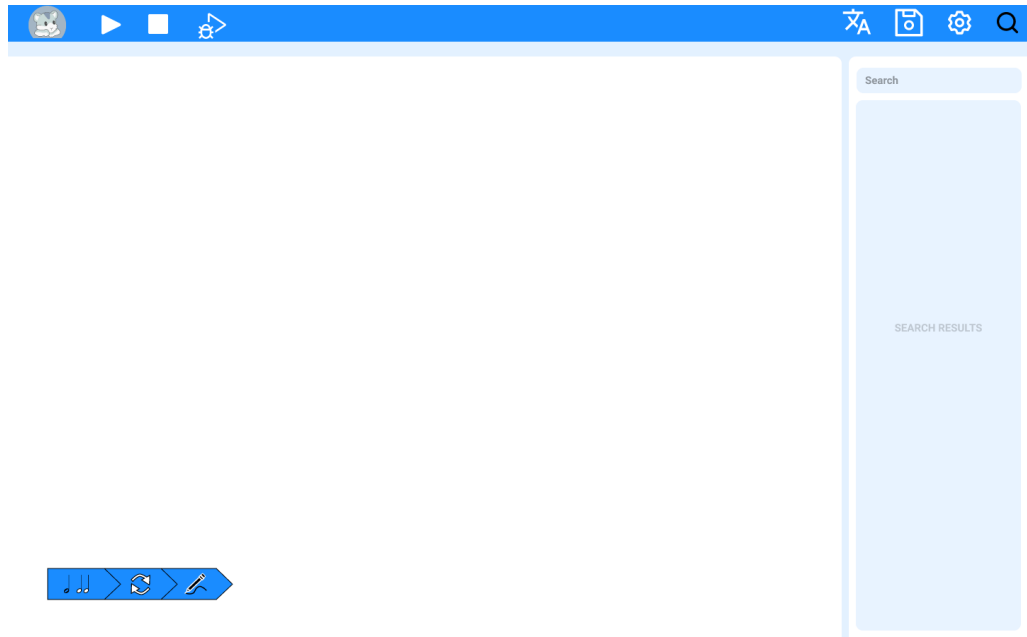
Settings Button will include: Change language button, switch modes, toggle javascript editor, set Pitch review, load plugin, delete plugin, turtle warp off, merge with the current project, help.

File Button will include: New project, load project, save the project, find and share project, restore.

Debug button will include: Show stats, run slowly, run step by step.

The dropdowns can be implemented, as shown in the screenshots below :





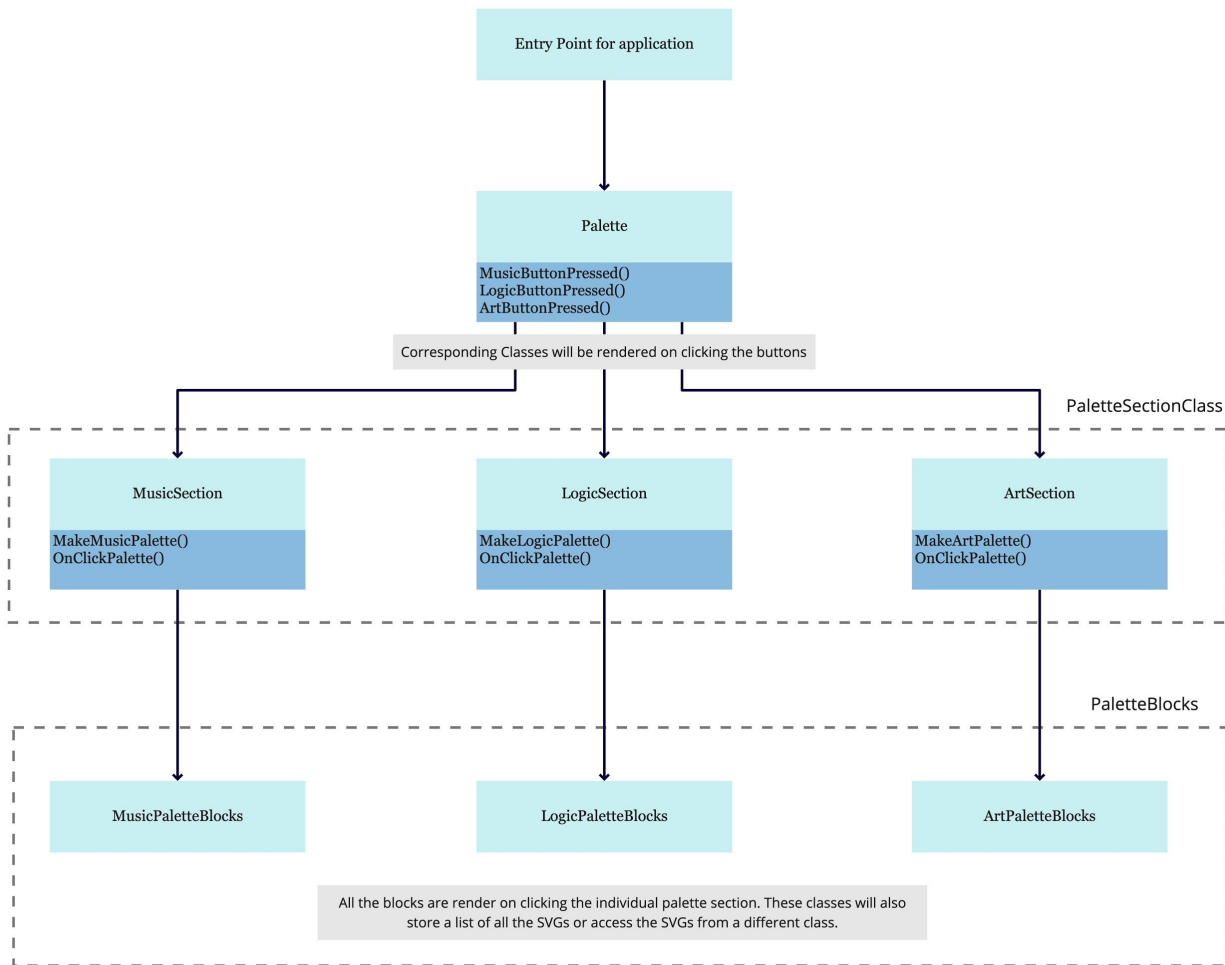
These wireframes were developed in Figma ([link](#)).

Class Structure of palette.js

The class structure of the palette will be divided into three sections.

1. Palette: This will render the main three sections of the palette (music, logic, and art).
2. PaletteSection: This will render subpalette for the three palette sections. For example, Rhythm, meter, pitch etc., for the music section.
3. PaletteBlock: Will render the blocks corresponding to each subpalette section.

Class diagram for palettes.



Link to PDF of class Diagram ([Link](#))

For making the expandable palette, we can use the accordion component of Material UI.

```

const Palette = () => {

  return (
    <div className={root}>
      <Accordion>
        <AccordionSummary>
          <Typography className={heading}>Music</Typography>
        </AccordionSummary>
        <AccordionDetails>
          <PaletteMusicSection /> //Call the paletteMusicSection Class
        </AccordionDetails>
      </Accordion>
    </div>
  )
}
  
```



```

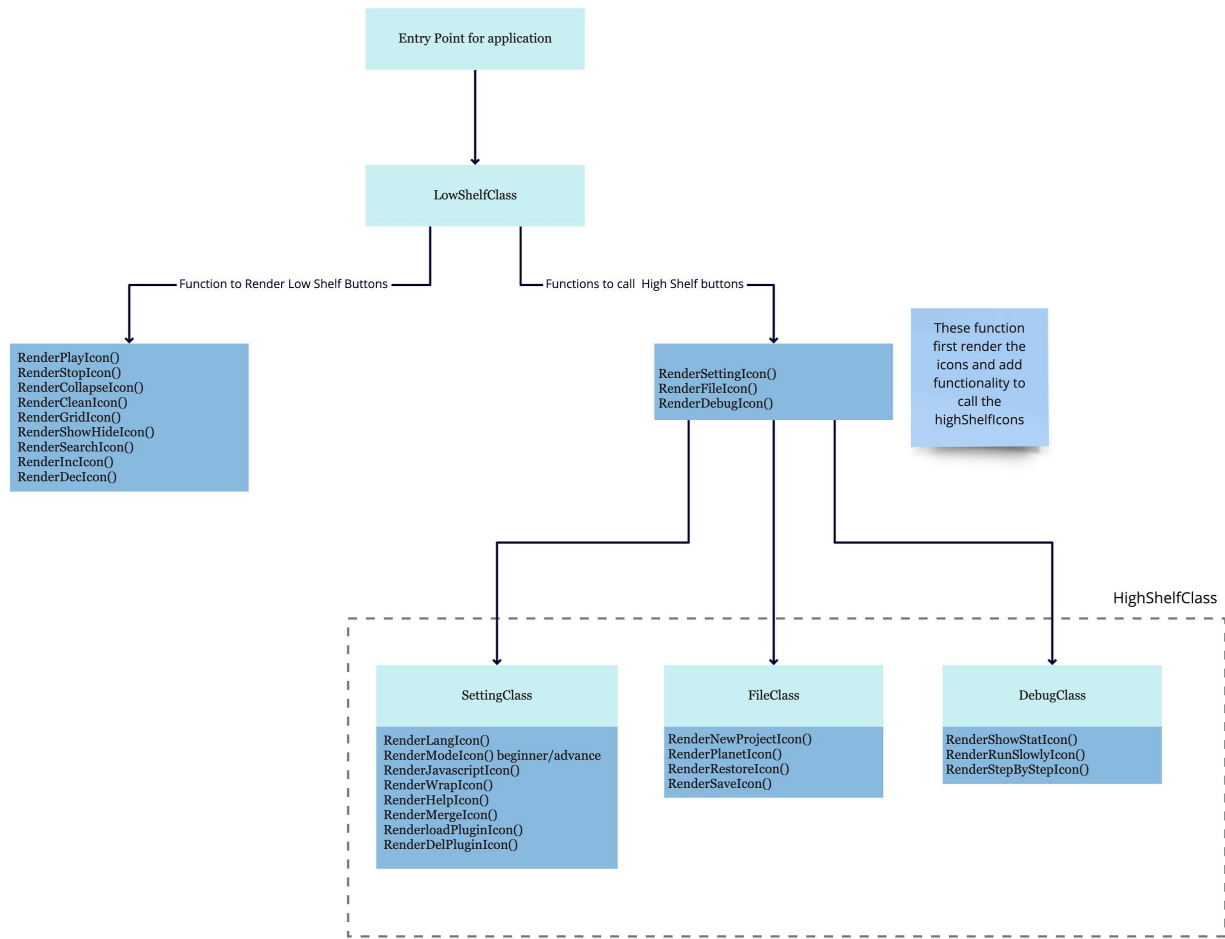
    <Accordion>
      <AccordionSummary>
        <Typography className={heading}>Logic</Typography>
      </AccordionSummary>
      <AccordionDetails>
        <PaletteLogicSection /> //Call the paletteLogicSection Class
      </AccordionDetails>
    </Accordion>
    <Accordion>
      <AccordionSummary>
        <Typography className={heading}>Art</Typography>
      </AccordionSummary>
      <AccordionDetails>
        <PaletteArtSection /> //Call the paletteArtSection Class
      </AccordionDetails>
    </Accordion>
  </div>
);
}

```

Class Structure of toolbar.js

The whole toolbar.js will be divided into two classes according to my above wireframe. First will be a class that will render the low shelf buttons, and the other class will contain render functions related to the corresponding high shelf buttons.

Class Diagram for Menus.



Link to PDF of class Diagram ([Link](#))

I have also written a small code in React, which will show how the high shelf buttons will show up on the website. ([Link](#))

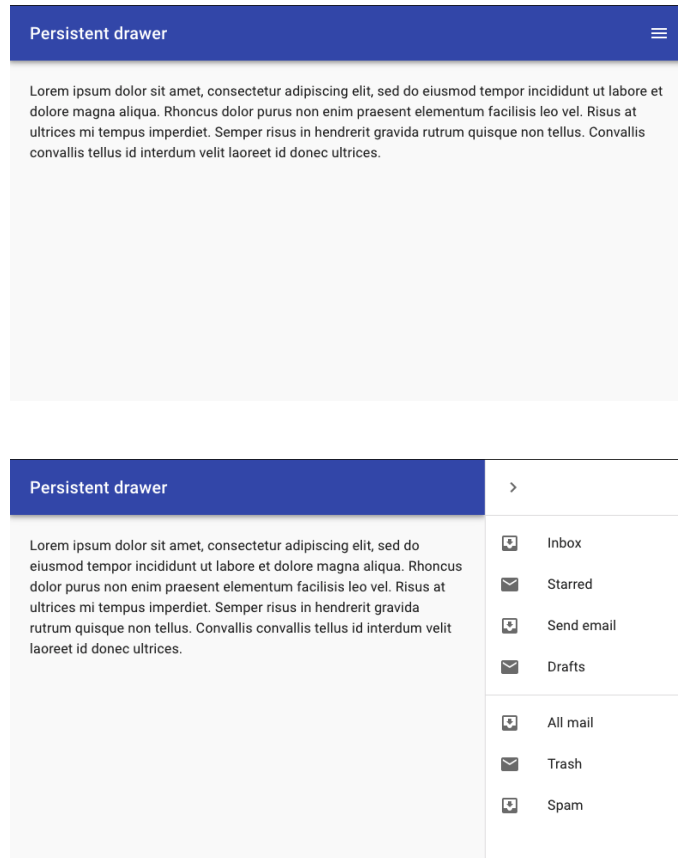
We can also implement this using Material-UI.

```

<Drawer
  className={drawer}
  variant="persistent"
  anchor="right"
  open={true || false}
>

```

It will look like this.



Advantage the new design will bring

Maximum work space

With the left palette section placed at the bottom of the screen which is collapsible to just 3 buttons, and all the buttons divided into different categories, the user workspace will become distraction free.

Quick Access to Palettes and buttons

With the palette and the buttons redesigned the user will be able to easily navigate through the blocks. An additional search button for the palette and buttons will minimize the effort of users in finding the different blocks and buttons.

Decluttering

Some buttons are frequently used while the rest are used much less. Some buttons are used rarely (users will probably use the Language buttons only once). Dividing the buttons into categories of high self and low self will remove a lot of buttons from the main working space of the user.

Addition Ideas

- While going through these palettes, I noticed that there are too many blocks, and the user can sometimes get confused about the location of the block. This also happens with the buttons. Therefore we should also look at improving the search section of the website. The search should also be able to look for buttons. Additionally, a new section can be created where the user can access his most used/searched blocks.
- Dividing the blocks into common and niche case blocks, will help in decluttering the palette section. The user will be able to navigate and find the blocks easily.
- The print statements, the console logs and be displayed in a status bar. The status bar can be placed at the bottom left side of the screen or in the debug section of the application. This status bar will stay collapsed until expanded.
- We could also separate the beginner mode and advance mode function, avoid using if-else statements and make the `style.display none` and `block`. This could be easily done by creating another class component for beginner and advanced mode. This change will help in debugging the code easily.

Milestones

During the coding period, I will follow Agile software development techniques. Since musicBlocks is being built from the ground up, we will require communication with each other since all the proposed projects in Gsoc are connected. So I will be in constant touch with my mentors, the other selected participants and the wider community at every step. Also, ensure that every PR is with proper documentation and testing.

Checklist for the coding period

- ☐ Finalise the wireframe for Menus and Palette.
- ☐ Decide the class structure based on the wireframe.
- ☐ Implement the class structure in React + Typescript

- ❑ Documentation and testing for the class Components.
- ❑ Testing the website for both small screen and large screen devices.

Application Review Period | April 13 - May 17

- In this phase, I will continue contributing to the organisation and keep building on react and typescript skills.
- I will familiarise myself with all the music terms and also with the rest of the musicBlock code.

Community Bonding Period | May 17 - June 7

The community bonding period would be a great time to get feedback from the mentors regarding the various project ideas given above. That would ensure that we are on the same page so that the coding period can be utilised effectively to achieve the milestones. I will start developing the basic class structure of menus and palette from Day 1. I will also make use of function components and react hooks that were introduced in React 16.8.

Phase 1 | June 7 - June 28

- Start working on the Menus.
- Build up the basic framework and the UI of the website in React.
- Complete my implementation of low shelf buttons class.

Phase 2 | June 28 - July 5

- Start my work on the palette.
- Complete the palette class and start working on my paletteSection class.
- I will be in constant touch with my mentor during all these phases with daily reports and review invitations.

Phase 3 | July 5 - July 19

- If any of my work from the previous 2 phases requires some correction or is incomplete, two week buffer period.
- If time permits will start working on the high shelf class.

Phase 4 | July 5 - July 26

- Complete the work on the high shelf class
- In this phase, I would have completed my menus section and some part of the palettes.
- Now I would start working on the palette section. First, complete the paletteSection class (if anything is left) and move onto the paletteBlock.

Phase 5 | July 26 - August 2

- Complete the paletteBlock section.
- Also, complete any of my unfinished work from the previous phases.

Phase 6 | August 2 - August 16

- Start wrapping up my work and complete the testing for the final evaluation.
- Also, complete the documentation which I was maintaining during all the phases.
- Create newcomer friendly issues to help easier onboarding of newcomers in this open source project

If I complete the tasks for any phase early, I will move on to the functions planned ahead. Also I will be maintaining the weekly or fortnightly blogs, where my progress and learning will be updated.

Other Commitments

My summer break will start on 15th May 2021. I have no major commitments during my GSOC period. So, I will be able to dedicate at least 40 hours per week during this period. My institute reopens in mid-August. But since the first month of the new semester remains quite relaxed and pressure is not too much, I am confident that I can work on the project and meet my deadlines.

Post GSoC

I would be proud to have completed GSoC under a prestigious organisation like Sugar Labs. Definitely, by the end of my project, I would be a well-connected

member of the community and would love to keep contributing towards Sugar Labs and helping other fellow members with the issues that they are facing. After all that the community would have taught me, I would try my best to give as much back to the community as possible !

Thank you for your time!

