Project Title

Music Blocks 4 Project Builder Integration

Synopsis

The new Project Builder Framework for Music Blocks v4 will revolutionize the way children and beginners learn programming. Instead of relying on traditional text-based programming, the graphical blocks manager module will provide a more intuitive and engaging learning experience. By visualizing different components as UI elements, users can easily interact with the code, leading to a smoother learning curve.

To implement this project, we will integrate the Project Builder Framework with the new Music Blocks and wrap it in a component to facilitate easy integration with the Specification and Syntax Tree APIs. Additionally, we will create a palette component and refactor the existing codebase. The final Result after the Implementation of idea will provide kids to learn Programming concepts in an integrative and fun way.

This project is a significant step towards the complete new Music Blocks, with a focus on performance improvements and new features. Leveraging modern JavaScript technologies and a better application architecture, the new Music Blocks will ensure a smooth and reliable user experience. Overall, this project has the potential to transform the way we teach programming to children and beginners, making it more accessible and enjoyable for everyone.

Bio of Student

My name is Abhinav Jha and I am from India. I am currently in my 2nd semester of my pre-final year of Computer Science and Engineering, which will be completed in early May. As a result, I will have ample time to devote to my GSoC project. If selected, I am willing and able to commit approximately 40 hours per week to this project, and I am open to putting in additional time if necessary to ensure its success.
While I have primarily worked with JavaScript frameworks and tools, I have also gained a deep understanding of their inner functionalities. In addition, I am highly skilled in TypeScript and React. I have experience working with both React JS and React Native, which are JavaScript libraries. This demonstrates my ability to work with a wide range of JavaScript frameworks and libraries, and I am confident that I can quickly adapt to any new technologies that may be required for this project.

I am deeply passionate about JavaScript and its frameworks, and I strive to learn new things every day. As part of my preparation for this project, I have been studying the existing codebase and gaining a better understanding of how things work. This will enable me to implement the selected idea more efficiently and effectively. My enthusiasm for programming and my ability to learn quickly are key strengths that I believe will enable me to make a meaningful contribution to this project.

Here are Link to some of My Contribution

Merged PRs

1	Replaced Bootstrap tour by IntroJs in Food Chain activity	LINK
2	Replaced Bootstrap tour by IntroJs in Reflection Activity	LINK
3	Replaced Bootstrap tour by IntroJs in EBook Activity	LINK
4	Replaced Bootstrap tour by IntroJs in TamTam Micro Activity	LINK
5	Replaced Bootstrap tour by IntroJs in Labyrinth Activity	LINK
6	Chess Activity Moves Alignment Fix	LINK
7	Alignment issue Fix	LINK

Issues

1	Irregular Alignment of Moves in Chess Activity	LINK
2	Irregular Alignment of Social Buttons	LINK
3	Matching not Allowed from right side	LINK

I have developed various web and mobile applications for hackathons and technical fests organized by my university other than the projects listed at the bottom of this document for reference. Currently working on a SaaS-based solution for the optical industry, which aims to help sellers manage their complicated mathematical data more efficiently and reduce costs while increasing profits. Additionally, I am developing a project management system for my university to centralize project information for professors.

For more information about my past projects and experience, please refer to the bottom of this document. There, you will find a detailed overview of the projects I have worked on in the past, as well as the skills and expertise I have developed through those experiences. I believe that this background will be a valuable asset as I take on this new project and work to achieve its goals.

In addition to my technical skills, I have also been active in the developer community, giving talks on topics such as Git & GitHub, ReactJS, Competitive Programming, and more. I am passionate about sharing my knowledge and helping others to develop their skills, and these talks have given me valuable experience in communicating complex ideas to a diverse audience. Additionally, I have organized a successful university-level hackathon, Dexterix 3.0, where I worked closely with a panel of judges composed of corporate leaders and CTOs. This experience provided me with a deeper understanding of real-world project management and the workflows used in professional environments.

I am an active member of the Google Developer Student Club and Facebook Developer Circle, as well as a core member of my university's technical community (TechnoJam).

Contact Information

Student name: Abhinav Jha
Student postal address: Flat no.09 Vishal Kunj Apartment, Kazipur Road No.04, Patna , Bihar - 800004
Telephone(s): +91 7700850095
Email(s): abhinavnarayan2002@gmail.com, jha.abhinav2710@gmail.com
LinkedIn: www.linkedin.com/in/abhinavjha27
GitHub: https://github.com/ABHINAV-JHA-27

Student Affiliation

Institution: Galgotias University
Program: B.Tech in Computer Science and Engineering with Specialization in Artificial Intelligence and Machine Learning
Stage of completion: 3rd Year (expected June 2023)
Contact to verify: verification@galgotiasuniversity.edu.in
Or contact: Mr. Deependra Rastogi: +91 84496 87767 (Associate Professor)

Schedule Conflicts

No Conflicts

Mentors

@AnindyaKundu
@WalterBender

Coding Plan and Methods

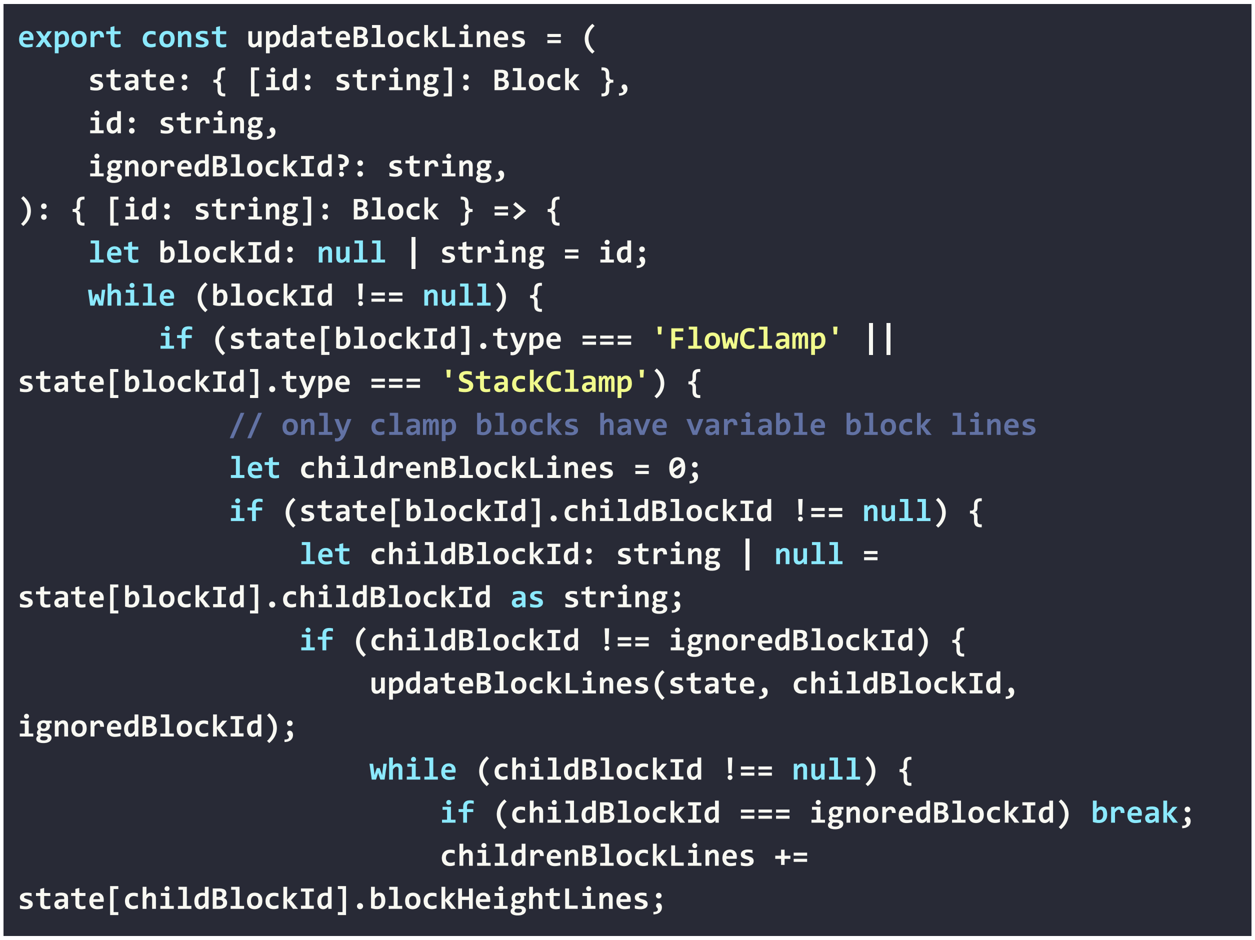
Background

During the GSOC 2023 program, my goal is to work on implementing the new Project Builder Integration for Music Blocks v4 using the Prototype Project Builder Framework. This project will allow children to learn programming in a visually intuitive way. By providing a visual programming environment, kids and beginners can experience a smoother learning curve compared to traditional text-based programming. The code components will be represented as UI elements, which can be easily manipulated and interacted with. This will enable children to learn the fundamentals of programming in a fun and engaging way, while also building their confidence in their programming abilities.

Deliverables

1) Refactoring of prototype code

Existing Code :



```
export const updateBlockLines = (
    state: { [id: string]: Block },
    id: string,
    ignoredBlockId?: string,
): { [id: string]: Block } => {
    let blockId: null | string = id;
    while (blockId !== null) {
        if (state[blockId].type === 'FlowClamp' ||
state[blockId].type === 'StackClamp') {
            // only clamp blocks have variable block lines
            let childrenBlockLines = 0;
            if (state[blockId].childBlockId !== null) {
                let childBlockId: string | null =
state[blockId].childBlockId as string;
                if (childBlockId !== ignoredBlockId) {
                    updateBlockLines(state, childBlockId,
ignoredBlockId);
                    while (childBlockId !== null) {
                        if (childBlockId === ignoredBlockId) break;
                        childrenBlockLines +=
state[childBlockId].blockHeightLines;
```

```
                        childBlockId =
state[childBlockId].nextBlockId;
                    }
                    childrenBlockLines -= 1; // clamps have one empty
block line by default
                }
            }
            console.log(childrenBlockLines);
            state[blockId].blockHeightLines =
                (state[blockId].defaultBlockHeightLines as number) +
childrenBlockLines;
        }
        blockId = state[blockId].nextBlockId;
    }
    return state;
};
```

It would Look something like this after refactoring

```
export const updateBlockLines = (state: BlockState, id: string,
ignoredBlockId?: string): BlockState => {
  let blockId: string | null = id;

  while (blockId !== null) {
    const currentBlock = state[blockId];
    if (currentBlock.type !== 'FlowClamp' && currentBlock.type !==
'StackClamp') {
      blockId = currentBlock.nextBlockId;
      continue;
    }
    // Only clamp blocks have variable block lines
    let childrenBlockLines = 0;
    if (currentBlock.childBlockId !== null &&
currentBlock.childBlockId !== ignoredBlockId) {
      const childBlockId = currentBlock.childBlockId;
      updateBlockLines(state, childBlockId, ignoredBlockId);
      let currentChildBlockId = childBlockId;
      while (currentChildBlockId !== null && currentChildBlockId !==
ignoredBlockId) {
        childrenBlockLines +=
state[currentChildBlockId].blockHeightLines;
        currentChildBlockId = state[currentChildBlockId].nextBlockId;
      }
      childrenBlockLines -= 1; // Clamps have one empty block line by
default
    }

    console.log(childrenBlockLines);
    currentBlock.blockHeightLines =
currentBlock.defaultBlockHeightLines + childrenBlockLines;

    blockId = currentBlock.nextBlockId;
  }

  return state;
};
```

Implementation Strategy

- Identify areas for improvement: Look for redundant code, duplicate functions, or complex logic that can be simplified.
- Refactor the code: Improve the structure and functionality of the code by rewriting portions of it, reorganizing code into smaller functions, and reducing dependencies between components.
- Modularize the code: Break it down into smaller, more manageable components to make it easier to maintain and debug, as well as making it easier to reuse code in other projects.
- Improve overall quality: Focus on improving the code's overall quality by ensuring it is well-structured, follows best practices, and is easy to read and understand.

2) Integration of framework into the music blocks-v4

The Code for Integration of Builder Framework would look something like this

```
import React from 'react';
import BuilderWrapper from './BuilderWrapper';

const App = () => {
  return (
    <div>
      //Existing MusicBlocks Code
      <BuilderWrapper />
    </div>
  );
};

export default App;
```

Implementation Strategy

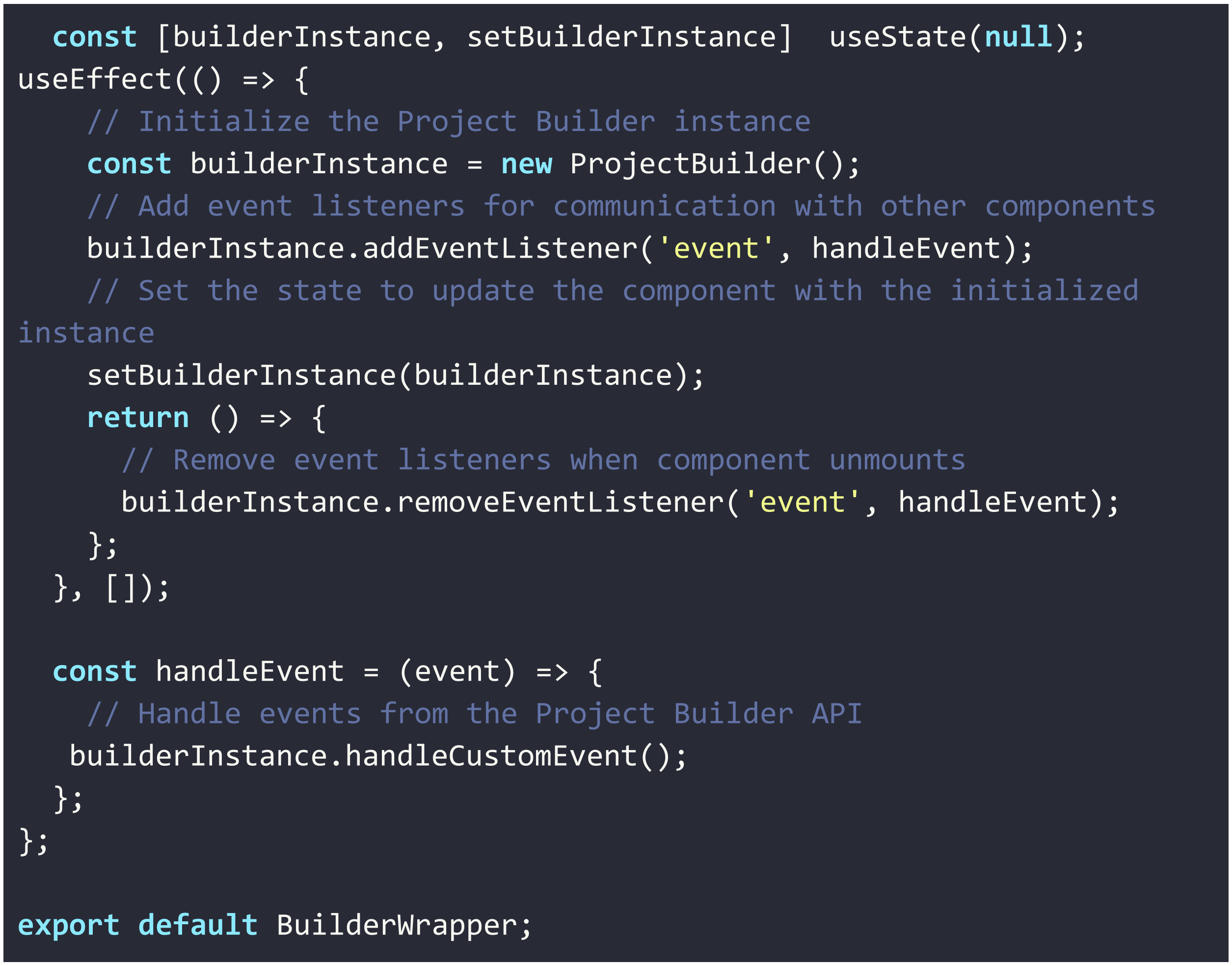
- Evaluate compatibility: Determine if the Project Builder framework is compatible with the existing music blocks-v4 platform in terms of programming languages, libraries, and dependencies.
- Identify dependencies: Identify any shared libraries, APIs, or other components that need to be present in both systems for them to work together.
- Integrate the framework: Integrate the Project Builder framework into music blocks-v4 by ensuring that it functions correctly and that any dependencies are met.
- Test the integration: Test the integration thoroughly to ensure that the framework works correctly within the music blocks-v4 platform.

3) Creation of a wrapper component

The proposed implementation would look something like this

```
import React, { useState, useEffect } from 'react';
import ProjectBuilder from 'musicblocks-v4-builder-framework';

const BuilderWrapper = (props) => {
```



```
  const [builderInstance, setBuilderInstance]  useState(null);
useEffect(() => {
    // Initialize the Project Builder instance
    const builderInstance = new ProjectBuilder();
    // Add event listeners for communication with other components
    builderInstance.addEventListener('event', handleEvent);
    // Set the state to update the component with the initialized
instance
    setBuilderInstance(builderInstance);
    return () => {
      // Remove event listeners when component unmounts
      builderInstance.removeEventListener('event', handleEvent);
    };
  }, []);

  const handleEvent = (event) => {
    // Handle events from the Project Builder API
   builderInstance.handleCustomEvent();
  };
};

export default BuilderWrapper;
```

Implementation Strategy

- Interface development: Design the user interface elements that allow users to interact with the Project Builder framework. This may include buttons, menus, or other interface elements that trigger actions within the Project Builder.
- Bridge development: Develop a wrapper component that acts as a bridge between the Project Builder framework and the music blocks-v4 platform. The wrapper component should handle communication between the Project Builder and other components of music blocks-v4.
- Integration testing: Test the wrapper component to ensure that it functions correctly and that it can communicate with the Project Builder framework and other components of music blocks-v4.

By following these steps, we will develop a wrapper component that provides an interface for users to interact with the Project Builder framework while also acting as a bridge between the framework and the music blocks-v4 platform. This will make it easier for users to create and edit music projects using the Project Builder framework within the music blocks-v4 platform.

4) Add utilities for communication with the Programming Framework

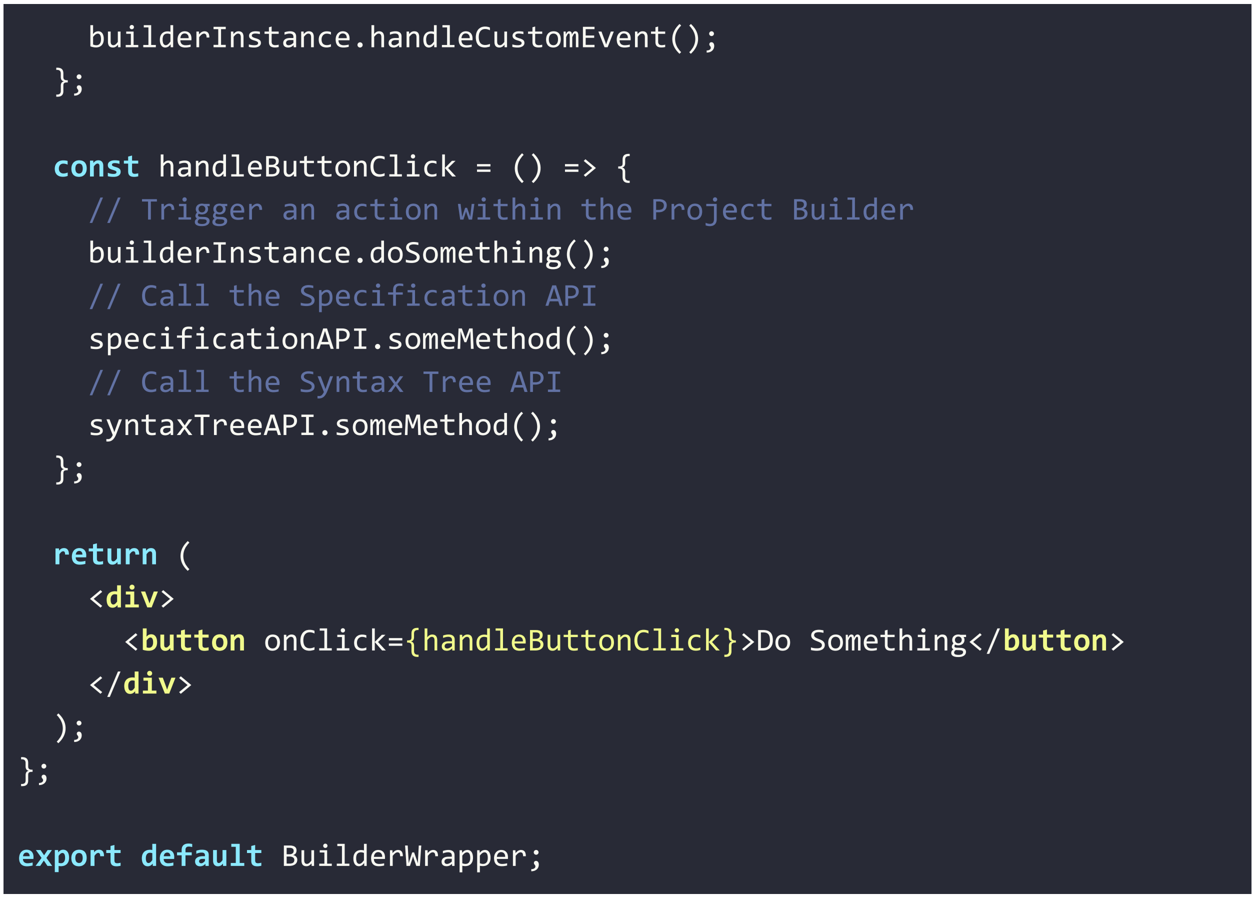
```
import React, { useState, useEffect } from 'react';
import ProjectBuilder from 'musicblocks-v4-builder-framework';
import { SpecificationAPI, SyntaxTreeAPI } from 'musicblocks-v4-programming-framework';

const BuilderWrapper = (props) => {
  const [builderInstance, setBuilderInstance] = useState(null);
  const [specificationAPI, setSpecificationAPI] = useState(null);
  const [syntaxTreeAPI, setSyntaxTreeAPI] = useState(null);

  useEffect(() => {
    // Initialize the Project Builder instance
    const builderInstance = new ProjectBuilder();
    // Initialize the Specification API
    const specificationAPI = new SpecificationAPI();
    // Initialize the Syntax Tree API
    const syntaxTreeAPI = new SyntaxTreeAPI();
    // Add event listeners for communication with other components
    builderInstance.addEventListener('event', handleEvent);
    // Set the state to update the component with the initialized instances
    setBuilderInstance(builderInstance);
    setSpecificationAPI(specificationAPI);
    setSyntaxTreeAPI(syntaxTreeAPI);

    return () => {
      // Remove event listeners when component unmounts
      builderInstance.removeEventListener('event', handleEvent);
    };
  }, []);

  const handleEvent = (event) => {
    // Handle events from the Project Builder API
```



```
    builderInstance.handleCustomEvent();
  };

  const handleButtonClick = () => {
    // Trigger an action within the Project Builder
    builderInstance.doSomething();
    // Call the Specification API
    specificationAPI.someMethod();
    // Call the Syntax Tree API
    syntaxTreeAPI.someMethod();
  };

  return (
    <div>
      <button onClick={handleButtonClick}>Do Something</button>
    </div>
  );
};

export default BuilderWrapper;
```

Implementation Strategy

- Identify communication needs: Identify the specific types of data that will need to be passed between the Project Builder component and the Specification and Syntax Tree APIs of the Programming Framework.
- Implement utilities: Implement the necessary utilities in the wrapper component by building upon the existing codebase for the Programming Framework. This may involve modifying existing components, creating new components, or integrating external libraries and APIs.
- Test the utilities: Test the utilities to ensure that they function correctly and that they can communicate with the Specification and Syntax Tree APIs of the Programming Framework.

By following these steps, we can add utilities to the wrapper component that enable it to communicate with the Specification and Syntax Tree APIs of the Programming Framework. This will make it possible for users to create more complex Music Blocks programs that incorporate advanced features and functionality.

5) Creation of Palette component

The code for Palette component will look something like this

```
import React, { useState, useEffect } from 'react';

const Palette = (props) => {
  const [items, setItems] = useState([]);
  useEffect(() => {
    // Fetch the items for the palette from the server or local storage
    const items = fetchPaletteItems();
    // Set the state to update the component with the fetched items
    setItems(items);
  }, []);

  const fetchPaletteItems = () => {
    // Make a request to the server or local storage to get the palette items
    const items = [];
    // Return the fetched items
    return items;
  };

  const handleItemClick = (item) => {
    // Trigger an action with the selected item
    props.onItemClick(item);
  };

  return (
    <div>
      <h2>Palette</h2>
      {items.map(item => (
        <div key={item.id} onClick={() => handleItemClick(item)}>
          {item.name}
        </div>
      ))}
    </div>
  );
```

```
};

export default Palette;
```

Implementation Strategy

- Implement the Palette component: Develop a graphical user interface (GUI) that displays the available Music Blocks that users can add to their programs.
- Implement the Drag and Drop feature: Add a drag and drop feature to the Palette component so that users can easily select and add Music Blocks to their programs by dragging them from the Palette and dropping them into the program.
- Integrate with the wrapper component: Ensure that the Palette component is integrated with the wrapper component so that it can communicate with the Project Builder framework and other components of music blocks-v4.
- Test the Palette component: Test the Palette component to ensure that it functions correctly and that users can easily select and add Music Blocks to their programs.

By following these steps, We will develop a Palette component that provides a graphical user interface for selecting and adding Music Blocks to a program, making it easier for users to create Music Blocks programs. The drag and drop feature makes it even more user-friendly and intuitive.

6) Documentation

Document the Project Builder framework. This would include documentation and tutorials that help users get started with the framework and use it to create Music Blocks programs.

If time is left then I’ll love to work on the following issue

- Add Tests for the components present in the Implemented projects.

Timeline

TIme Period	Plan
	Application Review Period **April 4, 2023 - May 4, 2023**
Week One/ Two	● Explore more about Music Blocks v4 Repo and Its Architecture
Week Three/Four	● Gaining a good experience with prototype Builder Framework
	Community Bonding Period **May 4, 2023 - May 28, 2023**
May 12 - May 28, 2022,	End-Semester Examination ● Discuss Project strategy with the mentor(s) ● Get to know about mentor(s) better. ● Ask about the actual design, Implementation and grab more about the features by mentor(s). ● Learn more about the internal management of the SugarLabs Github Repo(s) from the mentor(s) ● Create the initial setup and set up the project in the GitHub repository.
	Coding Period **May 29, 2023 - August 28, 2023**
Week One	● Setting up the development environment and Start working on Refactoring the CodeBase
Week Two	● Refactor the remaining codebase of the Prototype Builder Framework ● Make Sure everything is Working as expected
Week Three	● Start Working on the Integration of Builder Framework to MusicBlocks v4
Week Four	● Complete work of Integration of Builder Framework ● Start work on creation of a wrapper component for the framework
Week Five	● Complete the Wrapper Component ● Setup utilities for communication with programming framework

Week Six	**Mid Evaluation (July 25, 2022 - July 29, 2022)** In this I propose to complete and submit following ● Refactored Code for the Builder Framework ● Wrapper Component with Utilities for Communication with Programming Framework
Week Seven	● Start Working on Palette Component for the Framework
Week Eight	● Complete work for the Palette Component ● Test Code for any kind of bugs or any missed edge cases in the Implementation
Week Nine	● Write documentation ● Take feedback from the mentor(s) and perform the suggested changes ● Make sure that all the features are working fine and the Project Builder is working well.
Week Ten	● Try to add Tests for all the components and the Implementations done
Week Eleven	● ***Test*** and ***resolve*** bugs then clean up the code. ● Submit final deliverables. ● Create a summary and project report
	Coding Period Ends

Management of Coding Project

Given my experience with Git and Github through contributing to Open Source projects and working on personal projects, I plan to use Github as my primary platform for managing code on this project. I find it to be an effective tool for code management and collaboration, and I believe it will allow me to efficiently manage my work.

I plan to commit my work every 2-3 days, depending on the complexity of the issue I am working on. If an issue is resolved and the code is tested within a single day, I will push the code to the Github repository on that same day. If an issue takes longer to resolve, it may take me longer to push the code. I understand the importance of timely updates and will communicate any changes in the timeline to my mentors in advance.

In the event of unexpected delays or problems, I have allocated the last 2 weeks of the project to handle unforeseen issues. During this time, I will prioritize completing any outstanding tasks to ensure that the project is completed on time.

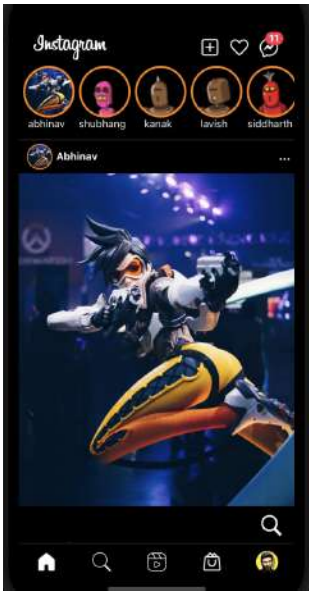
Benefits to SugarLabs

If I am chosen for this project, I believe that my skills in problem-solving and my extensive experience with Javascript and its frameworks will enable me to implement the selected idea efficiently.With a comprehensive understanding of the advanced topics and implementations, I feel confident that I can successfully execute the project from start to finish. Furthermore, my contributions to Sugar Labs have given me a solid foundation in its codebases and an understanding of its workings.

For gaining a deeper insight into my experience here are some of my Personal projects

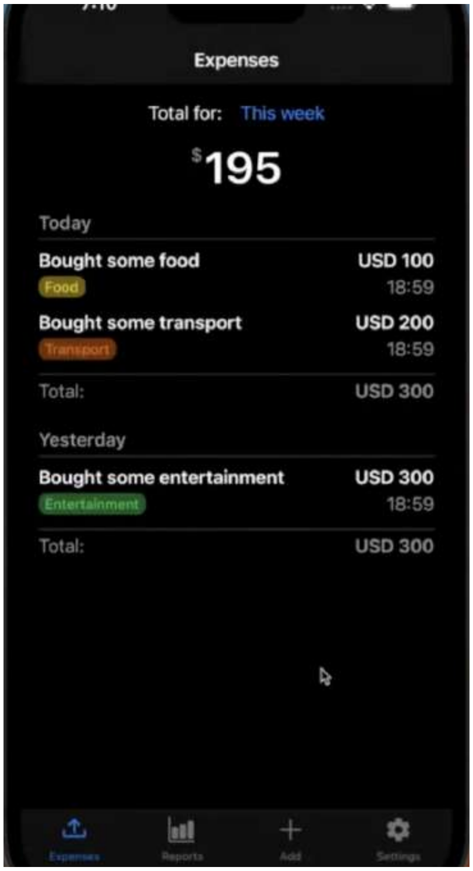
1) Instagram Clone

In this project, I developed an Instagram clone using React Native. The clone has many of the same features as the real Instagram app developed by Meta, such as posting photos and videos, liking and commenting on posts, and viewing stories. The app utilizes Firebase Authentication for sign up and login functionality, as well as Firebase Realtime Database to store user data and posts. As React Native is a cross-platform framework, the app can be deployed on both iOS and Android devices. This project demonstrates my experience working with cross-platform frameworks and my ability to develop complex applications using popular technologies.



2) Expense Tracker App

I developed an expense tracker app using React Native. The app enables users to track their expenses on a weekly, monthly, and yearly basis, with features such as visualization of expenses through charts for better insights. To implement this, I utilized my skills in the Redux toolkit and utilized different JavaScript packages such as d3 for visualization of charts. I prioritized keeping the codebase clean and optimized for better performance.



Expenses

Total for: This week

$195

Today

Bought some food — USD 100
Food — 18:59

Bought some transport — USD 200
Transport — 18:59

Total: USD 300

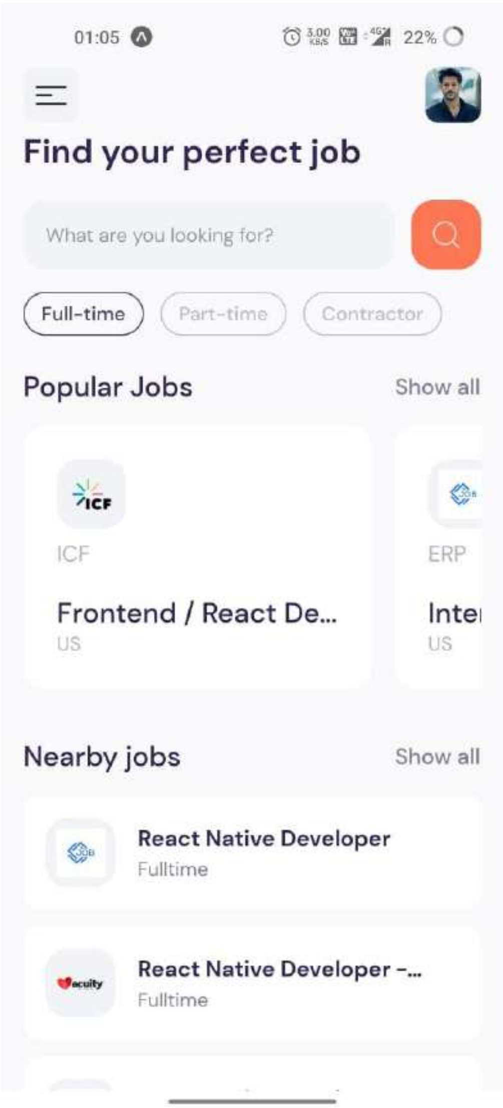
Yesterday

Bought some entertainment — USD 300
Entertainment — 18:59

Total: USD 300

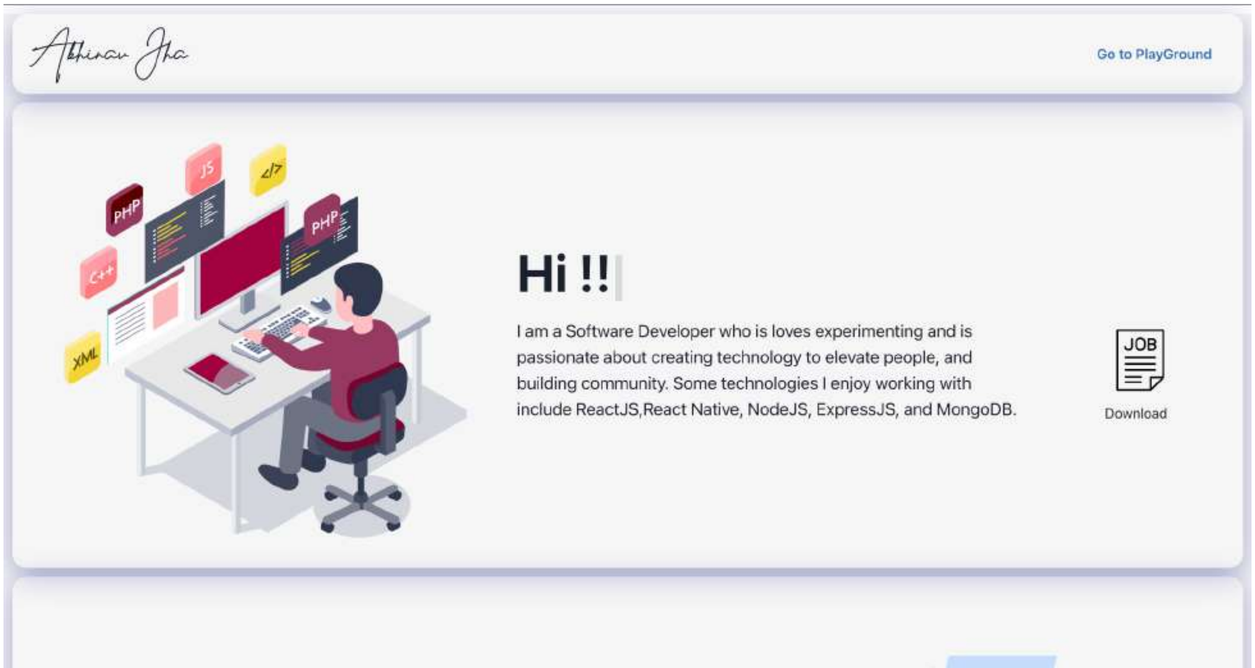
3) JobSeeker App

In this project, I developed a job search application for students that enables them to browse through various job opportunities from multiple websites. The app allows users to save jobs for later viewing and sharing with friends. I implemented the app using React Native and utilized the Jsearch API for providing job information. In the future, I plan to integrate a job scraper to gather data from more websites. Firebase was used for authentication and real-time database storage for saved job information.



4) Portfolio

My personal portfolio showcases the various projects I have built, my skill set, and my previous experiences. I developed the portfolio using React and utilized the Chakra UI framework for the frontend. The portfolio features illustrative animations that enhance the user experience, and I made sure to keep the design minimal and optimized. By building my own portfolio, I gained valuable experience in frontend development and demonstrated my ability to create a polished and professional website.



Hi !!

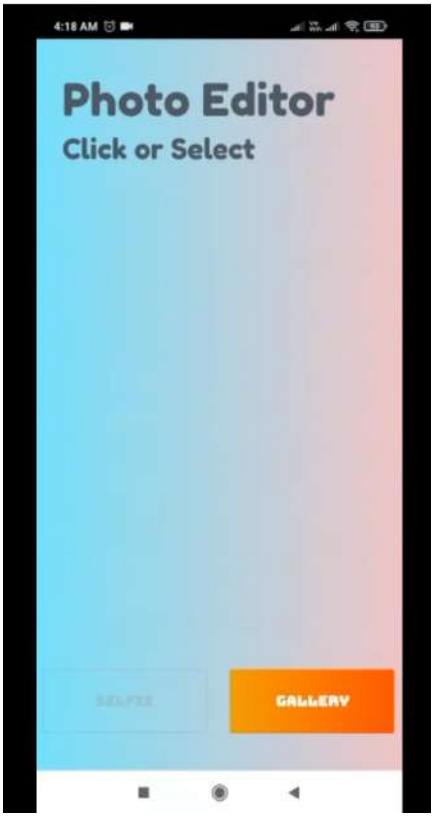
I am a Software Developer who is loves experimenting and is passionate about creating technology to elevate people, and building community. Some technologies I enjoy working with include ReactJS,React Native, NodeJS, ExpressJS, and MongoDB.

5) Mr.Meme App

In this project, I created a meme app that allows users to discover various kinds of hilarious memes scraped from Reddit. Built with React Native, the app includes features that enable users to set custom preferences for their feeds, such as topics they like and content preference like Dank, Anime etc. Users can discover the latest memes from all kinds of categories, send and share memes with other users, and save them as posts. To support these functionalities, the project utilized Firebase Authentication and Realtime Database. The app's intuitive interface and user-friendly design were optimized for an enjoyable user experience.

6) Image Editor App

In this project, I developed an Android app for image editing using Kotlin. The app offers a range of basic and advanced features for editing images, implemented with the help of an external image processing SDK. Users can choose to select a picture from their gallery or take a real-time photo and edit it using multiple image processing features. The app boasts a simple UI that has been optimized for a better user experience.



These projects provide a clear demonstration of my technical expertise and experience with various tech stacks