
GSoC 2024 Student Proposal

Sugarizer

Project: Sugarizer VueJS App

sugarizer



→ Divyanshu Kumar
20th March 2024

About

Name: Divyanshu Kumar

University: [FunctionUp School Of Technology](#)

Major: Bachelor Of Technology in Computer Science And Engineering

LinkedIn: [divyanshu-kumar](#)

Medium: [divyanshu_blog](#)

Email: kumardivyanshu118@gmail.com

Timezone: Indian Standard Time UTC+05:30

portfolio: [My_portfolio](#)

My Background

My name is Divyanshu Kumar, and I'm currently pursuing a Bachelor of Technology degree in Computer Science from Madhubani, Bihar, India. My passion for computer science stems from a lifelong fascination with technology and problem-solving.

Early on, I developed a keen interest in how things work. This curiosity began with simple devices like my family's Nokia phone, where I was captivated by games like Snake. I'd spend hours pondering the mechanics behind these games. Later, games like Hill Climb Racing, a popular game, fueled my interest further, particularly the way it simulated lunar gravity (around 1.62 m/s^2). While the physics behind it weren't entirely clear to me at the time, these experiences sparked a passion for understanding the world around me.

This fascination with applying scientific principles to the digital realm led me to excel in physics. I even secured a second-place rank in a regional physics Olympiad-style exam, demonstrating my aptitude for the subject. My experiences have instilled a deep desire to pursue a career in computer science. I'm eager to delve deeper into the field and contribute to the creation of innovative technological solutions. This interest in how technology can be used to create engaging and immersive experiences is what motivates me to participate in this project.

Coding Skills

I possess extensive coding experience in Java JavaScript and Typescript. Additionally, I am well-versed in Python. These languages have been my primary tools for developing and implementing various projects. Here are some notable examples:

- **My Portfolio-1:** I designed and built my personal portfolio using Vue.js, JavaScript, and HTML. It showcases my skills, projects, and achievements. [>>Project Link](#)
- **GitHubAccountFinder:** I created a project using HTML. It utilizes an API to search GitHub users and display their profiles [>>Project Link](#) .
- **My Portfolio-2 :** I designed and built my personal 2nd portfolio using TypeScript, NextJs and TailwindCss. It showcases my skills, projects, and achievements. [>>Project Link](#)
- **Programming:** Javascript, TypeScript, HTML And CSS(Tailwind CSS),Java(Core), Python,
- **Frameworks:** VueJS,Vuetify,NextJs, React.js and Node.js.
- **Tools:** Git, GitHub, GitLab
- **Database:** MongoDB Compass, Mysql
- **Software:** Docker, postman, android studio, xcode
- **openSource Library:** leaflet.js, echart.js ,Acertinty(not open source)

[My Journey with the Sugarizer App.](#)

Through my teacher, I was introduced to the Google Summer of Code (GSoC) program and the world of open source. One aspect of this community that particularly resonates with me is the opportunity to exchange ideas with individuals from diverse backgrounds, transcending geographical boundaries. Additionally, the prospect of contributing to projects that impact users worldwide is immensely rewarding.

Driven by my interest, I delved into exploring various organizations and discovered Sugarizer. However, setting it up on my local device presented challenges. With invaluable assistance from the community channel and my mentor, I overcame these hurdles and successfully configured it using Docker.

Subsequently, I embarked on a journey of learning through tutorials, including both the Sugarizer Vanilla JavaScript and Sugarizer Vue.js activity development tutorials. These experiences provided insight into real-world coding practices, such as file manipulation and activity creation. Currently, I am actively engaged in contributing to the project through pull requests and issue tracking, and on a daily basis I was also writing a blog on linkedin and medium, please checkout.

Proof of Work:

>>>[Locally steup through docker](#)

>>>[Vanilla Javascript activity tutorial](#)

>>>Raised an issue: <https://github.com/llaske/sugarizer/issues/1553> [Approved by mentor]

>>> Raised a Pr: <https://github.com/llaske/sugarizer/pull/1556> [Merged]

Although I raised multiple issues, yes! I am a bugger finder 😊.but they all not appropriate well i can also give the link of the issues which has been closed:

>>><https://github.com/llaske/sugarizer/issues/1565> [Approved by mentor]

>>><https://github.com/llaske/sugarizer/issues/1559> [Approved by mentor]

>>><https://github.com/llaske/sugarizer/issues/1557> [Approved by mentor]

>>><https://github.com/llaske/sugarizer/issues/1554> [Approved by mentor]

>>><https://github.com/llaske/sugarizer/issues/1544> [Approved by mentor]

>>><https://github.com/llaske/sugarizer/issues/1543> [Approved by mentor]

>>><https://github.com/llaske/sugarizer/issues/1542> [Approved by mentor]

Some other pull request in the open source:

[\[change\] Updated X logo #743 by d1vyanshu-kumar · Pull Request #755 · openwisp/openwisp-wifi-login-pages \(github.com\)](#) [merged]

<https://github.com/openwisp/netjsongraph.js/pull/248>
[open]

<https://github.com/openwisp/openwisp-radius/pull/506> [open]

<https://github.com/osl-incubator/scicookie/pull/213> [closed]

Synopsis

About Sugarizer:

Sugarizer is a free/libre learning platform designed specifically for children aged 6 to 12. But I am 20, I also like to play with Sugarizer 😊, With its intuitive interface and collaborative features, Sugarizer provides a unique learning experience.

Key Features:

1. **Activities:** Instead of standalone apps, Sugarizer offers a set of activities covering reading, writing, exploration, creativity, simulation, and play.
2. **Collaboration:** Encourages interaction and joint exploration among learners.
3. **Field Approved:** Based on ergonomic principles from the Sugar platform, used globally by thousands of children.
4. **Privacy-Respectful:** Zero ads and no tracking tools.

How to Use Sugarizer:

- App: Install locally on your device (Windows, GNU-Linux, MacOS, Android, iOS).
- Web App: Access via a browser (requires a Sugarizer Server).
- Server: Facilitates collaboration and network features.

Explore over 50 activities and empower young minds with Sugarizer! ✨

About The Project:

Project Goals:

1. Migration to VueJS: The primary objective of the project is to transition the Sugarizer Core UI from EnyoJS, which is deprecated, to VueJS, a modern and widely supported framework. This migration is essential for maintaining the sustainability and relevance of the Sugarizer platform.

2. Completion of UI Screens: The project aims to finalize the implementation of remaining UI screens, specifically focusing on the Neighborhood view and the Journal View. These screens are integral components of the Sugarizer interface and must be seamlessly integrated into the VueJS framework.

3. Tutorial Implementation: Another significant goal is to implement a comprehensive tutorial within the Sugarizer VueJS App. This tutorial will serve as a guiding tool for users, enabling them to navigate and utilize the features of the application effectively.

4. Electron Porting: The project entails porting the Electron part of Sugarizer to ensure compatibility with the new VueJS implementation. This task is crucial for maintaining cross-platform functionality and expanding the reach of Sugarizer across different operating systems.

5. Android Compatibility: The project includes building Sugarizer for Android and thoroughly testing the new VueJS implementation on this platform. Ensuring compatibility with Android devices is essential for reaching a broader user base and enhancing the accessibility of Sugarizer.

How am I going to implement the solution?

Implementing the solution for the Sugarizer VueJS App project involves several steps, each requiring careful planning and execution. Below is a detailed explanation of how I can implement each aspect of the project. This is my idea and after receiving guidance from a mentor we can for sure make this better.

1. Requirement Analysis:

- Begin by thoroughly examining the existing Sugarizer Core UI and the VueJS implementation developed in the previous GSoC project.
- Identify the specific requirements for migrating the UI from EnyoJS to VueJS, including any missing features or functionalities.
- Consider user feedback and input from stakeholders to ensure that the final implementation meets their needs and expectations.

2. Screen Implementation Planning:

- Create a detailed plan for implementing the remaining screens, including the Neighborhood view and the Journal View.
- Break down the implementation process into smaller tasks, such as UI design, component development, and data integration.
- Allocate resources and set timelines for each task to ensure timely completion of the screens.

3. Implement Remaining Screens:

- Begin by designing the UI layout for the Neighborhood view and the Journal View, taking inspiration from the existing Sugarizer UI design.
- Use VueJS to create Vue components for each screen, incorporating data binding and event handling to enable dynamic interaction.
- Integrate any necessary data sources or APIs to populate the screens with relevant information, such as user activities or journal entries.

4. Tutorial Development:

- Design a step-by-step tutorial to guide users through the features and functionalities of the Sugarizer VueJS App.
- Create interactive elements, such as tooltips or guided tours, to help users navigate the app and perform common tasks.
- Use VueJS to implement the tutorial interface, ensuring smooth transitions between steps and clear instructions for users.

5. Electron Porting Planning:

- Analyze the existing Electron part of Sugarizer to identify dependencies and compatibility issues with VueJS.

- Plan the porting process, including updating dependencies, refactoring code, and integrating with VueJS components.
- Consider any platform-specific functionalities or APIs used in the Electron part and ensure compatibility with the new VueJS implementation.

6. Port Electron Part:

- Begin by updating dependencies and refactoring code to ensure compatibility with VueJS.
- Integrate VueJS components into the Electron part of Sugarizer, replacing any deprecated or incompatible code.
- Test the Electron part thoroughly to identify and resolve any issues or bugs introduced during the porting process.

7. Android Build Planning:

- Research the requirements and best practices for building Sugarizer for the Android platform using VueJS.
- Identify any additional tools or technologies required for Android development, such as Cordova or Capacitor.
- Plan the build process, including compiling VueJS code, optimizing assets, and generating APK files for distribution.

8. Build Sugarizer for Android:

- Set up a development environment for building Android apps with VueJS, including installing necessary dependencies and configuring build settings.
- Compile the Sugarizer VueJS code for Android using tools like Cordova or Capacitor, ensuring compatibility with the target devices and Android versions.
- Test the Android build on various devices to ensure functionality and performance, addressing any issues or errors encountered during testing.

9. Testing and Quality Assurance:

- Conduct thorough testing of the Sugarizer VueJS App on different platforms and devices, including web browsers, Electron, and Android.
- Use automated testing tools to perform unit tests, integration tests, and UI tests to ensure the stability and reliability of the app.
- Gather feedback from users and stakeholders to identify any remaining bugs or usability issues and address them promptly.

10. Documentation and Reporting:

- Document the implementation process, including design decisions, code changes, and testing procedures.
- Create user documentation and tutorials for using the Sugarizer VueJS App, including instructions for installation, navigation, and troubleshooting.

- Provide regular progress reports and updates to stakeholders, highlighting key milestones, achievements, and any challenges encountered during the project.

By following this detailed implementation plan and leveraging technologies like VueJS, Electron, and Android development tools, I can successfully migrate Sugarizer Core UI to VueJS and deliver a fully functional and optimized application that meets the project goals and requirements.

By implementing these solutions with careful consideration of the project requirements and goals, the Sugarizer VueJS App can offer a robust, engaging, and accessible educational platform for children worldwide. Regular testing, user feedback, and iterative development practices will be essential to refine and enhance the application over time.

>>I also want to Implement some extra features as the guidance of a mentor if it is necessary.

How will it impact Sugar Labs?

In this project, we embark on enhancing the Sugarizer Core UI by leveraging the Vue.js framework, renowned for its exceptional speed, efficiency, and performance. By adopting Vue.js, Sugarizer stands to undergo a transformation, becoming not only more efficient but also more maintainable. This evolution is poised to have a profound impact on Sugar Labs, offering the following benefits:

- **Enhanced User Experience:** Vue.js enables the creation of highly responsive and interactive user interfaces, fostering a more engaging and intuitive user experience within Sugarizer. This enhancement aligns with Sugar Labs' mission to provide accessible and empowering educational tools for learners worldwide.
- **Increased Development Efficiency:** With its simplified syntax and component-based architecture, Vue.js streamlines the development process, allowing Sugar Labs developers to build and iterate upon features more efficiently. This efficiency translates to faster development cycles and quicker deployment of updates and improvements.
- **Scalability and Performance:** Vue.js is renowned for its scalability and performance optimization capabilities, making it an ideal choice for applications like Sugarizer that cater to diverse user bases and usage scenarios. As Sugarizer continues to grow and evolve, Vue.js ensures that the application remains responsive and performs optimally across various devices and environments.

Technologies which i used:

Technology Stack Used in the Project:

1. **Vue.js:** Vue.js is the primary frontend framework chosen for implementing the Sugarizer Core UI. Its reactive and component-based architecture facilitates the development of interactive and dynamic user interfaces.
2. **Vue Test Utils with Jest:** For testing Vue.js components, Vue Test Utils in conjunction with Jest are utilized. Jest provides a simple and efficient testing framework, while Vue Test Utils offers utilities for testing Vue components.
3. **Axios:** Axios is employed for making HTTP requests within the Vue.js application. Its simplicity and versatility make it an ideal choice for handling asynchronous data fetching and interacting with backend services.
4. **i18Next:** i18Next is utilized for internationalization (i18n) within the Sugarizer Vue.js App. It provides features for managing translations and supporting multiple languages, enhancing accessibility and usability for global users.
5. **IntroJS:** IntroJS is incorporated into the project for creating interactive tutorials and guided tours within the Sugarizer application. It allows users to familiarize themselves with the interface and features, improving user onboarding and engagement.
6. **Node.js:** Node.js serves as the runtime environment for executing JavaScript code on the server-side. It is used for various backend tasks, such as API development, server configuration, and build automation.

7. **HTML and CSS (Tailwind CSS):** HTML and CSS are the foundational technologies for structuring and styling the user interface of the Sugarizer Vue.js App. Tailwind CSS is a utility-first CSS framework utilized for rapid UI development and styling.

8. **RequireJS:** RequireJS may be utilized for managing dependencies and loading modules within the frontend application, ensuring modularity and efficient resource utilization.

This comprehensive technology stack encompasses frontend frameworks, testing utilities, HTTP clients, internationalization libraries, and build tools, ensuring a robust and efficient development environment for implementing the Sugarizer Vue.js App.

Potential TIMELINE For the GSoC Period:

Time Period	Start Date	End Date	To Do
Community Bonding Period	May 1	May 26	<ul style="list-style-type: none">- Completing the required setup for the project- Discussing the overall workflow of the project with my mentor- Understand the current architecture of sugarizer in-depth and I will contribute to the existing sugarizer repo achieving the same.

Week 1	May 27	June 2	Requirement Analysis and Planning <ul style="list-style-type: none"> • Conduct a thorough analysis of the existing Sugarizer Core UI and Vue.js implementation. • Identify specific requirements and features for migrating the UI to Vue.js. • Create a detailed project plan outlining tasks, milestones, and timelines. -
Week 2	June 3	June 9	Screen Implementation Planning <ul style="list-style-type: none"> • Develop a plan for implementing the remaining screens, including the Neighborhood view and Journal View. • Break down the implementation process into smaller tasks and allocate resources accordingly. • Set clear objectives and deliverables for each screen implementation. -
Week 3	June 10	June 16	Implement Remaining Screens <ul style="list-style-type: none"> • Begin implementing the Neighborhood view and Journal View screens using Vue.js components. • Design the UI layout and incorporate necessary functionalities such as data binding and event handling. • Collaborate with designers to ensure consistency with Sugarizer UI design principles. -
Week 4	June 17	June 23	Tutorial Development <ul style="list-style-type: none"> • Design and develop a comprehensive tutorial module within the Sugarizer Vue.js App. • Create interactive elements and instructional guides to assist users in navigating the application. • Test the tutorial module to ensure its effectiveness and usability.

			-
Week 5	June 24	June 30	Electron Porting Planning <ul style="list-style-type: none"> Analyze the existing Electron part of Sugarizer to identify dependencies and compatibility issues with Vue.js. Plan the porting process, including updating dependencies, refactoring code, and integrating Vue.js components. Consider platform-specific functionalities and ensure compatibility with the new Vue.js implementation.
Week 6	July 1	July 7	Port Electron Part <ul style="list-style-type: none"> Begin porting the Electron part of Sugarizer to work with the new Vue.js implementation. Update dependencies and refactor code to ensure compatibility and seamless integration. Test the Electron part thoroughly to identify and resolve any issues or bugs introduced during the porting process.
Week 7 (Mid-term evaluation)	July 8	July 12	<ul style="list-style-type: none"> - Present a demo of the new features to mentors. - Collect feedback and refine the implementation plan based on the evaluation.
Week 8-9	July 13	July 27	Android Build Planning <ul style="list-style-type: none"> Research the requirements and best practices for building Sugarizer for the Android platform using Vue.js. Identify additional tools or technologies required for Android development, such as Cordova or Capacitor.

			<ul style="list-style-type: none"> Plan the build process, including compiling Vue.js code, optimizing assets, and generating APK files for distribution. <p>-</p>
Week 10-11	July 28	Aug 11	<p>Build Sugarizer for Android</p> <ul style="list-style-type: none"> Set up a development environment for building Android apps with Vue.js. Compile Sugarizer Vue.js code for Android using tools like Cordova or Capacitor. Test the Android build on various devices to ensure functionality and performance. <p>-</p>
Week 12	Aug11	Aug 18	<p>Testing and Quality Assurance</p> <ul style="list-style-type: none"> Conduct thorough testing of the Sugarizer Vue.js App on different platforms and devices. Use automated testing tools to perform unit tests, integration tests, and UI tests. Gather feedback from users and stakeholders to identify any remaining bugs or usability issues and address them promptly. <p>Documentation and Reporting</p> <ul style="list-style-type: none"> Document the implementation process, including design decisions, code changes, and testing procedures. Create user documentation and tutorials for using the Sugarizer Vue.js App. Provide regular progress reports and updates to stakeholders, highlighting key milestones, achievements, and challenges encountered during the project.

			-
Week 13 (final week)	Aug 19	Aug 26	<ul style="list-style-type: none"> - Final evaluation and code submission - Collect feedback from the community implementation if required/final fixes. - Blog on complete experience and final demonstration video.
Mentors submit final student evaluations	Aug 26	Sep 2	<ul style="list-style-type: none"> - Mentors submit the evaluations
Post GSoC			<ul style="list-style-type: none"> - Maintain the Sugarizer VueJs app in the future. - Keep contributing to Sugarizer VueJs app and keep improving its functionality.

Other Deliverables

Weekly report and blog

Participation

Progress Report

- I will remain online on Discord during my working hours (1 pm to 11 pm UTC +5:30)
- I will write weekly blog posts at (<https://medium.com/@kumardivyanshu118>)
- I will share my blogs on [Linkedin](#)
- Write weekly scrum reports and updates.
 - What did I do last week?
 - What will I do this week?
 - What is currently preventing me from reaching my goals?

I will submit a Project Presentation

I will use GitHub to manage bugs and task



Availability:

→ I am committed to dedicating 40 hours weekly to coding throughout the program. This availability ensures that I can fully immerse myself in the project and make substantial progress each week. By allocating a significant portion of my time to coding, I aim to maximize productivity and effectively tackle the tasks outlined in the project proposal. This dedicated commitment reflects my passion for learning and contributing to the project, and I am eager to invest the necessary time and effort to achieve success.

Post GSoC Plans:

As a newcomer to Sugar Labs, I am deeply inspired by its commitment to fostering a free/libre open-source environment aimed at enhancing the learning experience for students. While the prospect of contributing may seem daunting at first, I am enthusiastic about embracing this opportunity to grow and learn within the organization. My goal is to actively engage as a contributor, immersing myself in the community's collaborative spirit and leveraging its wealth of resources to expand my knowledge and skills. Looking ahead, I envision myself evolving into a mentor for programs like GSoC, equipped with the experience and expertise gained through my journey with Sugar Labs. Regardless of the outcome of GSoC selection, I am dedicated to making meaningful contributions and fostering positive growth within the Sugar Labs community.

→ Interests:

- **Development**
- **Robotics**
- **Artificial Intelligence and Machine Learning -{ AI-Trader}.**
- **Ethical hacking**
- **Blockchain**

