



Sugarlabs: Sugarizer VueJS Core

About me

Name	Ayesha Shaik
Email	shaikayesha95155@gmail.com
Github	https://github.com/Ayeshashaik759
Language	English
Location	Bhimavaram, Andhra Pradesh, India - IST (UTC +5:30)
Matrix	Shaik Ayesha

Introduction

Hi, I am Ayesha, a third year computer science undergraduate from Vishnu institute of technology, Bhimavaram. Being a computer science student I really like coding and working with different technologies.

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

I got to know about open source as a part of an event called MLH global hackweek conducted by an organization called Major League Hacking (MLH) in which I participated in some fun activities. Later I participated in hacktoberfest-2022 and as a part of it I started to contribute to some beginner level issues like.

[To-Do-list project](#)

[Rock-Paper-Scissors game](#)

[Css_bug](#)

[Css_bug](#)

[Adding new card](#)

Later I started to search for some good organizations to contribute as I got some familiarity with open source and got comfortable working with git and github. I found the Sugar labs organization interesting. I first tried the Sugarizer project in Sugar Labs. I visited the url <https://sugarizer.org/> and started playing a few games to understand the working of the Sugarizer project and I really loved playing those games. Then I started to contribute to the Sugarizer project.

You can see my contributions below

First I have gone through the issues and found the issue [Replace Bootstrap tour by IntroJS in tutorial for all activities](#) interesting and implemented this in few activities like:

[clock activity](#)

[calculate activity](#)

[Last one loses activity](#)

[QR Code activity](#)

[Sprint math activity](#)

[FotoToon activity](#)

[Tank operation activity](#)

[Flabracman activity](#)

[Fraction activity](#)

[Xmas lights activity](#)

[Implode activity](#)

[Planets activity](#)

[Physics js activity](#)

[Maze web activity](#)

Later I worked on the issues like [adding a score feature to Fraction activity](#) and [Css changes to make the background of the scroll bar of the journal container visible in some activities](#).

I have also done some UI contributions to the Internet health report organization.

[Link-1](#), [Link-2](#).

Convince us that you will be a good fit for this project, by sharing links to your contribution to Sugar Labs

I started using Sugarizer in October 2022. I like to play online games. This is one of the reasons why I got connected to Sugarizer. Before contributing to Sugarizer I have never used Vuejs so I started to learn Vuejs and as I am familiar with html, css and javascript before itself I searched for the beginner level issues to start with. Then I found the issue [Replace Bootstrap tour by IntroJS in tutorial for all activities](#) so I started working on this issue and implemented it for 14 activities and the links to my contributions are available above and I have also helped some other contributors who worked on this issue by giving suggestions to make the required changes. I have got a good understanding of Sugarizer core architecture while working on this issue.

Next I worked on the issue of adding a score feature to the Fraction activity. While working on this issue I got a good understanding of how to perform localization in Sugarizer.

Later I have raised an issue regarding the [background of the scroll bar of the journal container visible in some activities](#). And worked on this issue.

Prerequisites for Project:

I have Completed both the [Sugarizer Vanilla Javascript activity development tutorial](#) and the [Sugarizer Vue.js activity](#)

[My js tutorial link](#)

[My Vuejs tutorial link](#)

[Tutorial video link](#)

Next I have Explored the [VueJS Sugarizer UI components](#) and tried to use the components. And I read and tested the L10N study [here](#) by going through the test [html page](#).

As I have never worked with vue js before. So I tried to get myself comfortable working with Vuejs before working with the actual project so I implemented a project called [Restaurant-management](#) which is used to manage data of various restaurants. In this project I have implemented routes to different pages like sign-up page, login page, update page, add restaurant page which will be very helpful for me in this project as we are going to implement various screens here. And I have used json-server for backend then created and worked with two apis one is for users and another for restaurants. Whenever a new user signs in I send the data to api and whenever a user login I get the data from api which will also help while working with the login screen in this project.

I have also gone through the [EnyoJs wiki](#) to get some understanding code written in Enyojs in Sugarizer. One of my best qualities is that I am dedicated to my work. And if I don't know anything then I will spend dedicated time to learn them so thus I think that I will be a good fit for this.

Project Details

What are you making ?

In this project we are going to reimplement a part of Sugarizer Core UI using VueJS components which are implemented in the gsoc-2022 project. Here we are going to implement the screens for First screen, Login, Home view, List view, Settings using Sugarizer VueJS components and Sugarizer Server API. We can

implement this screens as follows by getting some reference from the code written in EnyoJs in js/firstscreen.js, js/homeview.js, js/listview.js, js/dailog.js

First screen:

In the first screen we will have a div tag for the toolbar with white background color then we have another div tag inside which we are going to have six div tags.

First div for the help button and we use the icon component to implement this help button and we are going to add a click event listener to it and the `startTutorial` method is implemented on click event but in case if a new person is visiting the website for the first time then it is automatically implemented.

Second div is for the new-user to sign in. For this also we are going to use the icon component and we are adding a click event for this. The `newUser` method is implemented on the click event.

Third div is to display the text “New user” below the new-user icon.

Fourth div is for the login to implement the login we are using the icon component and we are adding a click event for this. And the `login` method is implemented on the click event.

Fifth div is to display the text “Login” below the login icon.

Sixth div is for displaying the history i.e your previous logged in id. We can maximum display the 3 recent history buttons. To display the history we are having a method `setupHistory`. And we are using the icon-button component for this history button and the history button has a click event listener i.e when we click the click the history button `historyClicked` method is implemented.

This how we can see first screen in a web application for the desktop application we have additional icon i.e stop-button on top right below the header to quit the application it is implemented using the icon component and we are adding a click event to it that implements the `quitApplication` method on click action.

Login screen:

We can create a new user by clicking on New User Icon

step1:then we are prompted to screen that has a help-button with similar function as in first screen and next we have a div That contains a text “Choose

name:” below this We are div that contains an input tag that takes the input of user name from user Finally at bottom we have the prev and next buttons at left and right that prev button help us to go back to the previous phase and the next button help us to go forward. We are using the icon-button component to implement these buttons. The prev button has a click event listener and it implements the `previous` method on clicking it. The next button also has a click event listener which implements the `next` method on click action.

step2:On clicking the next button after entering a valid user name we again have the help button prev and next button similar as before and along with then we see a input box and some images to select our password we are using the password component to implement this and above this we have div containing the text “Choose at least 4 images” .

step3:After entering a valid password and clicking the next button we are prompted to a screen that has a user-icon in new and text above “click to change the color” here we can choose the color for our user icon and we are implementing this user icon with the help of icon-component. After clicking next in case of a web browser we get a screen which asks us to accept the cookies. Here we have a div tag with some text and help-button and instead of prev and next button here we have the accept and decline buttons at bottom left and right implemented using the icon-button component. For the decline button we add a click event listener that implements the `decline` method. The accept button has a click event listener which implements the `accept` method on click action. On accepting the cookies we are prompted to the home view.

The existing user can login by using their username by two types either by selecting the user from the history button if present or else by clicking the login icon in both the cases the next steps are going to be similar. Then we are prompted to the screen where we have entered a password similar to **step2** as mentioned in the new-user case and then if the user enters the correct password we are routed to the home screen.

Now at present the code for this is written in Enyojs in the file `js/firstscreen.js`. To implement these two screens in vue js we can create two files one for the first screen and another one for the login screen.

Home view:

In the home view we have a **toolbar** which consists of a search box. We can implement this with the help of the searchfield component. Next we have a help-button with functionalities as in the first and login screen and similar to that if it is the first time for the user to use

the Sugarizer then the `startTutorial` method is implemented automatically here. Then we have a sync button next we have a offline button to which we add a click event listener that implements the `doServerSettings` method on click action and the sync button and offline button are not visible in all cases. Next we have the show assignments icon. We can implement it with the help of the icon component and we are adding a click event to it that implements the `showJournal` method on click event. Then we have an assignment count. This is also not visible in all cases. Then we have the three buttons radial button, neighborhood button and list button these are used to change between different views. The radio button has a click event listener which implements the `showRadialView` method on clicking it. The radio neighborhood button has a click event listener which implements the `showNeighborView` method on clicking it. The list button has a click event listener which implements the `showListView` method on clicking it.

Initially below the toolbar we see the owner icon at the center of the page we can use the icon component to implement this below this we have a journal icon we are again using the icon component to implement this. Around this some activities are arranged in a spiral view. We arrange only our favorite activities in a spiral manner. For this we have to write a function to arrange the owner icon and journal in the center and get the list of your favorite activities and arrange them in a spiral manner. For displaying each activity we are using an icon component. We can see the activities are colorized of the activity is played at least once and all the instances of the activity played are saved and when we hover the mouse on the activity we can see the instance and whenever we click on a particular instance a click event is listened and the method `runMatchingActivity` is implemented. And when we hover the mouse on the desktop owner icon the `showBuddyPopup` method is implemented and we get a popup that contains two options one is for settings and another one for logout we see about the setting later below. For the logoff option we will add a click event and when we click the logoff option the `doLogoff` method is implemented and we are logged out of our account and we can see the first screen.

List view:

When we click on the list button in the toolbar of the home page we are routed to the list view

The list view also has the toolbar similar to the home screen below that all activities are arranged in a list manner. We get the list of activities from the activities collection from the Sugarizer server api.

We have a div tag for each activity and inside that again we have Five div tags. First one for favorite icon we can implement it using the icon component and a click event is added to this which implements `doSwitchFavorite` method on clicking it.

The second div is for the activity icon here we again use the icon component to display the activity icon. And we are adding a click event listener to it which implements `doRunNewActivity` method on click event.

Third div is to display the name of the activity.

Fourth div is to display the version of the activity.

Fifth div is for the help-button which gives some information about the activity on click event for this also we are using the icon component. It implements the `doHelp` method on click event

Settings:

In the home screen on mouse-hover on the user-icon the settings option appears and on clicking it we get the dialog-box we are going to use the dialog component to implement this. In this were going to complete the js/dailog.js

In settings when we click on about me an click event is listened and the `meClicked` method is implemented.on me clicked method the dialog box content changes and we can see the dialog box as follows: it contains a toolbar which consist of About me-icon and we are using icon component for this then we have div with text "About me" and we have a cancel button and ok button. We are adding a click event to the cancel button which implements the `cancel` method on click event. Next we also add a click event to the ok button that implements the `ok` method on click event. Below the toolbar we have a div tag that contains the text "click to change color".Below that we have five owner-icons with different colors all of these five have a click event which implements the `setColor` method on click event.Below that we have an input box that initially contains our username we can change our username from here.

On clicking the About my computer a click event is fired and we can see the content of the dialog box changed as follows: it contains a toolbar similar as in About me here in the toolbar instead of the About me-icon we have the About my computer icon, text beside it is has the text "About my computer" and below the that we have details of our system and browser. And code to this is also written in the gsoc-2022 project.

We also add a click event to 'About my server' and the method `serverClicked` is implemented on clicking it. In this method we are creating a component `DialogServer` that replaces the content of the dialog box it consists of toolbar similar to above but we use About my server icon in place of About me-icon, and the text beside it has the text "About my server" and below the tool bar we have a checkbox and div tag containing the text connected to the checkbox and then you have div tag containing the text server address, server name, Description and user name on the server. It is used to know our server details.

My security is used to change our password. On clicking My security in the dialog box we have a toolbar similar to above toolbars and in place of About me-icon we have security icon and text beside it has the text "My security" in toolbar and below the toolbar we will have a div tag with the text "if you want to change your images, first type current images" below that we are implementing the password component and we are also having next icon-button we are using the icon-button component to implement this. We will add a click event listener to the next icon-button which implements the `next` method on click action. If the correct password is entered then we are prompted to the screen containing the same toolbar and below that we have a div tag containing the text "now choose at least 4 new images" below that again we will have the password component and below that we have done icon-button for this we will use the icon-button component.

On clicking My privacy an event is fired and the `privacyClicked` method is implemented and the content of the dialog box is changed as follows with the toolbar similar as mentioned in about me here we use the privacy icon and text beside it is "My privacy". Below the toolbar we have 3 checkboxes. For the first checkbox we have text beside it as "Do not send to the server statistics about my usage of the app". For the second checkbox we have the text beside it as "Do not synchronize my local journal with the server". And for the third checkbox we have the text beside it as "Display delete account features" on checking the third checkbox i.e "Display delete account features" `showRemove` method is implemented and we can see two icon-buttons, "Delete local account" and "Delete server Account" we will add click event listeners for both this icon-buttons. The `removeLocalAccount` method is implemented on clicking the "Delete local account" icon-button and the `removeRemoteAccount` method is implemented on clicking the "Delete server Account".

Language setting is used to set our language on clicking this the dialog box content changes as follows: that has the toolbar with similar functionality as mentioned in above settings and here we use the language icon in the toolbar and below the tool bar we have div tag with text "Choose language you prefer:"

and below that we will use the selectbox component to select different languages.

Unit Testing and code coverage:

For unit testing I am thinking of using vue test utils and jest as they are previously used in the

Gsoc-2022 Sugarizer Vue.js UI project. And we can perform the code coverage with help of jest.

While performing unit testing for the first screen, login screen, home view, list view and settings I will create different files for testing each screen and write the various tests to check if all the methods are working properly and all screens are implemented correctly.

How will it impact Sugar Labs?

Sugarizer Core UI relies on EnyoJS, a deprecated frameworks initially developed for WebOS. Last year a GSoC project built the basic components stack of the Sugarizer UI (see [here](#)).The idea of this new project is to reimplement a part of Sugarizer Core UI using VueJS components. As the EnyoJS is a deprecated framework and we don't have much resources to learn EnyoJS so by implementing the Sugarizer core UI it will very helpful for the developers as there are many online resources available for learning VueJs so it will easy for the new people in the community to understand the codebase.

What technologies (programming languages, etc.) will you be using?

As mentioned in the project description we are going to use javaScript/HTML5 and Vue.js framework.

Timeline

Break down the entire project into chunks and tell us what you will work on each week.

Community Bonding (May 4 - May 28)

1. Connect with the Mentors
2. Discuss the proposal with the mentor & look for possible improvements or

changes.

3. Get familiar with codebase and the development environment
4. Get a better understanding of working on Sugarizer core UI and set up a development environment for this project.

Week - 1 (May 29 - June 4)

1.I will start implementing the localization component as it is mentioned as the first step of this project and try to complete it in the first week.

Week - 2 (June 5 - June 11)

1. I will start the implementation of the first screen and try to complete the implementation of the first screen in the second week

Week - 3 (June 12 - June 18)

1.I will start the implementation of the login screen. First I will begin with the new user login.

2.Next I will Implement the login for existing user and login of users from history box and I will try to complete it by the end of this week

Week - 4 (June 19 - June 25)

1.I will start with the home screen

2.I try to complete the implementation of toolbar of the home screen

3.If time permits I will start the implementation of spiral view of activities

Week - 5 (June 26 - July 2)

1.complete the home screen and logoff functionality.

Week - 6 (July 3 - July 9)

1.will keep this week as a buffer week to complete if any pending works

2.spend time on writing documentation of the work done till now

Week - 7 (July 10 - July 16)

1.start the implementation of the list view

2.Try to complete the implementation of list view.

Week - 8 (July 17 - July 23)

1.start with the implementation of settings

2.will complete the implementation of settings for About me, About my computer, and About my server.

Week - 9 (July 24 - June 30)

1.Continue with the settings implementation and complete the implementation for My security, My privacy, Language.

Week - 10 (July 31 - August 6)

1.start writing unit tests and complete writing unit tests for the first screen, login screen and home screen.

Week - 11 (August 7 - August 13)

1.continue unit testing for the remaining screens i.e for list view and settings

2.perform the code coverage.

Week - 12 (August 14 - August 20)

1.will keep this week as a buffer week to complete all remaining works

2.will complete the remaining documentation work.

Commitments & Availability

1.weekdays- 08:00 AM - 12:00 PM (GMT +5:30)

- 05:00 PM - 12:00 AM (GMT +5:30)

2.Weekends- 10:00 AM - 02:00 PM (GMT +5:30)

- 05:00 PM - 12:00 AM (GMT +5:30)

I have no other commitments during this summer so I can dedicate all my time to this project.

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

I will stay connected with the community after GSOC and continue to use Sugarizer and continue my contributions to Sugarlabs.