Sugarizer VueJS Core

About me:

Name	**:**	**Anuj verma**
Email	**:**	**panuj7295@gmail.com**
Github	**:**	**https://github.com/Anujverma89**
Language	**:**	**English**
Location	**:**	**Kolhapur, Maharashtra, India (UTC +5:30)**

I'm Anuj Verma, a 2nd year computer science and engineering student from Dkte society's textile and engineering institute ichalkaranji kolhapur maharashtra.

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

For now i haven't made any contribution to open source or sugarizer as i have recently came to know about open source but Considering my past experience with AI model developments and projects called **Credify** and other commercial project development for an interior design agency called **HD Design house** I can ensure I'm a good fit for the project.
Credify link
Interior client
To demonstrate that I'm capable of completing this project I have completed the activity development tutorial in vue.JS and vanilla js. I have also made a dummy(few functionality are working fine and few are not implemented) implementation of the sugarizer core UI using vue.JS and sugarizer vue.JS components. I have fully

implemented Login and first screen rest home view , list view and others are remaining.
NOTE:- This is just dummy, it's not responsive for small devices, user no password length validation, no activity icons.
Click to see Sugarizer core UI implemented in vue.JS

Project Details

This project aims to reimplement the Sugarizer Core UI by utilizing Sugarizer Vue.JS components. The current implementation of the Sugarizer Core UI relies on Enyo.JS, which is a deprecated web framework. Therefore, this project is necessary to update and improve the Sugarizer Core UI to ensure that it remains a modern and efficient tool. Work flow of how project will be implemented

What are you making?

1)First screen component

a) Contains :-
   - i) help
   - ii) newuser
   - iii) login
   - iv) history
   - v) tutorial

b) Implementation :-
   - i) Design a vue js component with following divisions
   - ii) Will be making value truthy and falsy on different screen interactions.

2)Login component

a) Contains:-

i) username component
ii) password component
iii) help
iv) choose color
v) back and next buttons
vi) server address form while in local mode, desktop or mobile

b) Implementations:-

i) API used:-
   1) auth/login → to create and login user
   2) auth/signup → to validate if user exists or not
   3) No additional headers are required for this api

ii) If new user
   1) Check if user exits → auth/singup
   2) If user exists throw error
   3) Else take the username
   4) Send user to password field
   5) Take password where password length is more than 4
   6) Send user to choose color
   7) Then collect all the data like password , color, name, and make requests to auth/login.
   8) Login the user
   9) Make request to activities
      (a) api/v1/acitivites
      (b) They take access token and x-key
   10) Make request to journal using user id
      (a) api/v1/journal/:uid
      (b) Get the journal info and store it in the local storage of the browser with the help of sugar web
   11) Now send to the Home **view** component .

iii) If login :
   1) Ask username

2) Check if user exists → auth/signup
3) If not exists throw error
4) If exists then send to password field
5) Take the password and make request to auth/login
6) If Login successful
7) Make request to activities
   - (a) api/v1/acitivites
   - (b) They take access token and x-key
8) Make request to journal using user id
   - (a) api/v1/journal/:uid
   - (b) Get the journal info and store it in the local storage of the browser with the help of sugar web
9) Now send to the Home **view** component .

3)Home view component

a) Contains:-

- i) Navigation
  1) Search
  2) Sync (v-if= “synchronizing”)
  3) Server connection error(v-if= “connection expired”)
  4) Home view icon
  5) Neighborhood view icon
  6) List view icon
- ii) Main view:-
  1) Activities in radial view (in color of the user)
  2) Xo buddy icon
  3) Journal icon (link to journal);

b) Implementation:-

- i) Get the activities from local storage(sugar_settings)
- ii) Design a radial view using a radius and coordinates
- iii) Add link to the activity index page to the icon
- iv) When clicking on an activity send to the index of a particular activity.

4)List view component

1) Contains :- Navbar (same of home view)
   a) List wise activities
   b) Version of activity
   c) Favorite
   d) Activity name
   e) Activity icon
2) Implementation :-
   a) Get all the activities from local storage
   b) Check is favorite is true or false
   c) If Fav is true set favicon coloured else set is empty colored
   d) Populate the data in the dom
   e) On click on activity send to their native index.html

5)Settings

i) **About me :-**
   1) Render xo buddies with different colors
   2) Get the prepared color
   3) Get the user name
   4) Make a put request to the server
   5) api→PUT('api/v1/users/:uid')
   6) Data : Xo buddy new color or new name

v) **About my server**
   1) Load UI with the serve data present in localstorage localStorage.get(sugar_settings.server.url)

vi) **About my computer**
   1) Get the serve data present in localstorage localStorage.get(sugar_settings.server.version)
   2) Get client info from (window.clinetInfromation)
   3) Populate the dom with the data

vii) **My privacy**
   1) Send stats to server

a) api→ POST(api/v1/stats/)
b) If not selected turn of
c) If selected turn on

2) Synchronize journal
   a) Turn on
   b) Turn off (if off no sync of the journal)
   c) Make update request PUT('api/v1/users/:uid')
3) If delete user:-
   a) Make delete request DELETE('api/v1/users/:uid')

viii) My security

1) api→PUT('api/v1/users/:uid')
2) Data→ userdata + new password
3) re-login()

ix) Language

1) Getting selected language from the form input
2) Making a put request
3) api→PUT('api/v1/users/:uid')
4) Data→ userdata + new language
5) Re login()
6)

6)Neighborhood view

1) Show the users connected to the same server

7)Journal view

1) Contains:-
   a) Navbar
   b) Bottom nav bar
   c) Shared filter
   d) Private filter
   e) Select box
   f) Favorite
   g) Pagination

h) Search box

2)Implementation:-

0).populate the journal data from local storage

1) API→ ("/api/v1/journal?type=shared")

2) ("/api/v1/journal?type=private")

3) (api/v1/journal/:jid)

How will it impact Sugar Labs?

Sugarizer uses enyo.JS a deprecated javascript framework in its core UI. Replacing enyo.JS with vue js will make the code base of sugarizer code base understandable by more people resulting in better and more contribution to the sugar labs and sugarizer. Also I have added to change **JOURNAL** and **NEIGHBORHOOD** view which is also a part of sugarizer core UI this will help to completely eliminate enyo js from sugarizer core UI.

What all technologies will you be using?

Timeline

Technologies that i will be using for this project are listed below:-

TECHNOLOGIES	USES
vue.JS	To implement main UI
Require.JS	As a module loader
Axios	To make Http request to server
i18n	For localization
Sugarizer vue.Js components	To achieve modularity
View test utils with jest	For unit testing

TIMELINE:

Community Bonding (May 4 - May 28)

1)Discussing the workflow of the project with a mentor.

2) Setting up the development environment

3)Understanding the sugarizer code base deeply and contributing to the same.

4)Discussing the communication medium and time for report submission.

DUE TO MY END SEMESTER EXAMS I MAY BE OFF GRID

Week 1st (May 29 - June 4)

1) Implementation of sugarizer localization
2) Making the sugar-web ready to handle new framework and UI

Week 2nd (June 5 - June 11)

1) Development off first screen
2) Development of Login Screen
3) Development the password screen

Week 3rd (June 12 - June 18)

1) Development of Navbar UI
2) Development of radial view

Week 4th (June 19 - June 25)

1) Continue developing the radial view(Home view)

Week 5th (June 26 - July 2)

1) Development of list view

Week 6th (July 3 - July 9)

1) Development of neighborhood view

Week 7th (July 10 - July 16)(Mid term evaluations)

Goals for mid term evaluations are :-

1) Completion of localization
2) Completion of first screen
3) Completion of login screen
4) Completion of Home view
5) Completion of list view

Week 8th (June 17 - July 23)

1) Implementation of journal view

Week 9th (July 24 - June 30)

1) Implantation of setting view

Week 10th (July 31 - August 6)

1) Unit testing and code coverage

Week 12th (August 7 - August 13)

1) Unit testing and code coverage

Week 13th (August 14 - August 20)

1) Code review and bug fixing (if there any)

Week 14th (August 21 - August 27)

1) Giving final touch to the product .

Week 15th (August 21 - August 27)
(Final Evaluation)(August 28 - September 4)

Goals for final evaluation is completion of all the seven screens (first screen, login, home view, list view and settings

Screen, neighborhood, journal), implementation of other important features of sugarizer like localization and saving history of user and unit testing coverage of the components.

How many hours do you plan to spend each week on your project?

I’m planning to give 4-6 hours daily to the project so it will sum of 25 to 35 hours weekly

How will you report progress between evaluations?

I would like to submit a weekly report of the project over the suitable medium that will be discussed.

Your plans after GSOC ends ?

I recently got to know about open source and gsoc through the internet and I will love work and be a part of any open source organization and contribute to it. Whether my proposal will be picked or turned down I will always be enthusiastic to contribute to sugarizer and open source.