

Sugarizer Knowledge Activity Pack

About Me

Name : Prakash Ujjwal

Email: prakashujjwal1010@gmail.com (Primary),
prakashu.co18@nsut.ac.in (Professional)

Github: <https://github.com/prakashujjwal1010>

Languages: Hindi (Native), English

medium: <https://medium.com/@prakashujjwal1010>

Location: I am located in New Delhi, India and my time zone is Indian Standard Time (UTC + 5:30). I am planning to work from 6:00 to 16:30 (UTC) but my timings are flexible. I'm very excited to work on this project and I can surely manage and adjust my time and be active when the mentors are available.

Degree : Bachelor of Engineering

Major : Computer Engineering

Institute : Netaji Subhas University of Technology, New Delhi (formerly NSIT)

What is your motivation to take part in Google Summer of Code?

GSOC is one of the most prestigious annual coding programmes which provides a very good opportunity to undertake projects in the midst of a community that leads to a lot of learning and growth.

Why did you choose Sugar Labs ?

Sugar Lab is a community that focuses on providing an open-source learning platform for children which uses technical platforms for a societal cause.

I too want to be a part of Sugarlabs which helps many children to learn something new in a fun way.

The idea of helping so many children by the programme i will code and the satisfaction it will give will be magnificent.

Why do you want to work on this particular project ?

The problem statement of the project is very interesting and the project is using technologies which aligns with my skills and which I have worked upon before.

I have developed a few sugarizer activities before also which makes it very intriguing for me to choose and work on it.

What are your expectations from us during and after successful completion of the program?

I expect to be mentored by the community during the GSoC period so that I can learn and contribute at the same time which will, in turn, help me to add value to the community and help others. Plus I expect the active communication between me and organization and mentors.

Skills : Js, Python, C++/C , Reactjs/ Redux, Vuejs, Nodejs, Ethereum Dev, Truffle, HTML, CSS, Docker, Flatpak.

Projects

I have experience in web development and decentralized apps development.

- 1) Split-Party:** DAPP built upon ethereum blockchain to plan, manage, share, track your party expenses in a trustless way. It is developed using Reactjs, truffle. The main idea was to use ethereum in some fun and unique way.
<https://github.com/prakashujjwal1010/Split-Party>

- 2) Stubble-Trouble** : it is a group project. Here, In India , Stubble burning is a very big problem (which is one of the reasons for pollution and smog in new delhi around november). We are aiming to provide a marketplace and an online self-sustainable business model which would connect farmers to potential buyers of stubble. In this way the stubble burning could be stopped and could be used in an efficient way.
We got an offer from our Institute's E-Cell for funding.
It is not yet completed. We have plans to continue working on it after a few months(after GSOC).
<https://github.com/prakashujjwal1010/SIH2020-1>

- 3) **Image-Uploader:** It is an implementation of an image uploader which sends an encrypted image to the server and then the server decrypts it and stores it.
<https://github.com/prakashujjwal1010/image-uploader>

Open Source Contribution

I have some experience in open source.

I have been contributing to the open source community for more than 1 year which includes contribution in sugarlabs, scorelab etc.

Sugarlabs Contributions

I started contributing to sugarlabs and started exploring its various projects in september 2019. I started to give attention to all the discussion that used to take place in the sugar-devel list.

During this journey I learnt many new things (e.g. flatpak, localization)

Pull requests:

- [porting Falabracman Activity to sugarizer](#)
- [added set-time and set-time-game mode in clock activity](#)
- [added changing generation speed feature in Game of Life Activity](#)
- [added changing grid size feature in Game of Life Activity](#)
- [added a feature to display trails of cell's deaths](#)
- [chess Activity for sugarizer as GSOC task](#)
- Packaging of sugar activities to Flatpak
 - [packaged Falabracman activity to Flatpak](#) (or [here](#))
 - [packaged Napier activity to Flatpak](#)(or [here](#))
 - <https://github.com/tchx84/flathub/pulls?q=is%3Apr+author%3Aprakashujjwal1010+>
- [fixing disappearance of clock in set time mode](#)

- [fixing responsiveness for falabracman sugarizer activity](#)
- [enabling touch events in XOEditor sugarizer activity](#)
- [added license tag](#)
- [fixing clear and save icon in XOEditor](#)

Issues:

- [touch events not enabled in XOEditor](#)
- [Stop Watch Activity is not responsive and has text overflowing problem.](#)
- [Bug While Resizing game screen after resizing setting screen in Falabracman activity](#)
- [Activity needs license metadata for AppStream file](#)
- [reappearance of tutorial on resizing window after clicking next button on last step of tutorial](#)

Others:

- [sugarizer vue activity template](#)

Contributions to Other Organizations

1. Scorelab:

- [Pull requests](#) in Senz

2. Publiclab:

- <https://github.com/publiclab/image-sequencer/issues/1241>

3. NSITulatorWeb:

- <https://github.com/NSITulator/NSITulatorWeb/pull/6>

ABOUT THE PROJECT

The name of the project is Sugarizer Knowledge Activity Pack which consists of two activities - Curriculum Activity and Vote Activity.

These two activities are requested by [Sugarizer deployment in Saint-Ouen](#).

1) Curriculum Activity :

The Curriculum activity will be a way for a student to self check its skill in a set of knowledge and provide multimedia elements to demonstrate these skills.

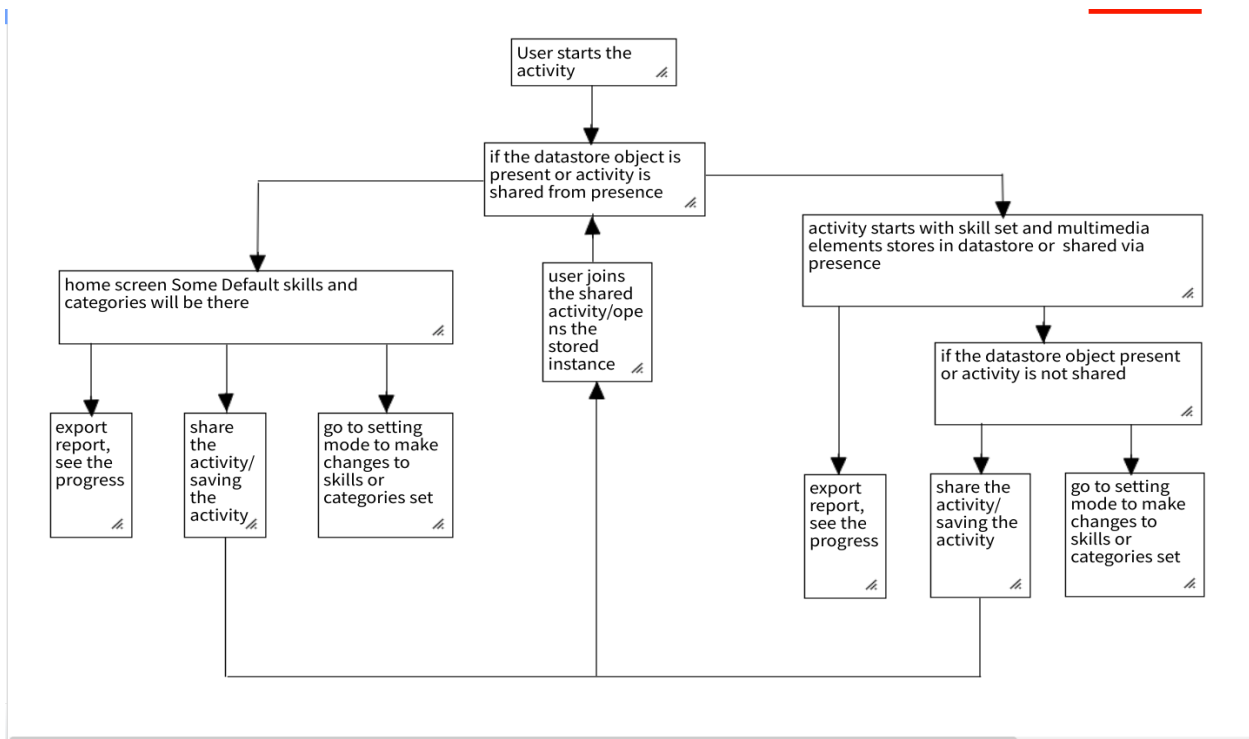
The activity will have following features

- The activity will have two modes - play mode and setting mode.
- In play mode,
 - activity will display a hierarchical set of skills grouped by categories then let the user explore the tree.
 - On each skill the user should be able to validate (i.e. skill acquired) and provide multimedia elements (pictures or sounds coming from Journal) to demonstrate the skill.
 - A category will have a title and a color, a skill will have a title and an image.
 - All the multimedia elements users have provided for skills are saved in the journal.
- In setting mode,
 - Users can create new skills and categories and can add their own title or image or colour .
 - Users can edit/delete/sort existing skills and categories.
 - All the changes users have done to the set of skills or categories should persist with the help of a journal.
- The user will also have an export to pdf or doc feature (in the form of a toolbar button) which basically creates a document with all skills and dated multimedia elements.
- Users can see their progress by a pictorial illustration using charts or some animation.
- Users can add the skill set by using fetching data over url.

- Users can share their activity using presence to share their skills with others in read only mode means users joining the activity can only see the user's skill set and their progress made so far with the multimedia elements provided.
- Activity should be localized, responsive, should have a tutorial, have a fullscreen button

Implementation:

The user flow is following-



Storing activity instance: We will use the journal for storage.

Datastore object for our activity should be following :

```

{
  skills : [
    {},
    {},
    ...
  ],
  categories : [
    {},
    {},
    ...
  ],
}

```

(There will be other fields also which will be used to facilitate storing activity.)

The skill object will be :

```
{
  title : "",
  image : "",
  category : "",
  aquired : false,
  multimedia-provided : [
    {
      type : "",
      data : ""
    },
    {
      type : "",
      data : ""
    }
    ...
  ]
}
```

And the category object will be:

```
{
  title : "",
  color : ""
}
```

We will use Lz-based compression algorithm to compress the data before storing it in journal using <https://github.com/pieroxy/lz-string> (as used in exerciser activity)

```
jsonData = LZ.compressToUTF16(jsonData);
```

Exporting to pdf:

We can use [html2canvas](#) and [jsPDF](#) to generate pdf files.
(the above approaches are also used in write activity)

Exporting report to doc: we can use the below approach.

```
document.getElementById('gen').addEventListener("click", function() {
  var title = 'experiment';
  var content = document.getElementById('content');
  var header = "<html xmlns:o='urn:schemas-microsoft-com:office:office' " +
    "xmlns:w='urn:schemas-microsoft-com:office:word' " +
    "xmlns='http://www.w3.org/TR/REC-html40'>" +
    "<head><meta charset='utf-8'></head><body>";
  var footer = "</body></html>";
  var sourceHTML = header + content + footer;
  var inputData = 'data:application/vnd.ms-word;charset=utf-8;base64,' + btoa(unescape(encodeURIComponent(sourceHTML)));
  var mimetype = 'application/msword';
  var metadata = {
    mimetype: mimetype,
    title: title + ".doc",
    activity: "",
    timestamp: new Date().getTime(),
    creation_time: new Date().getTime(),
    file_size: 0
  };
  datastore.create(metadata, function() {
    console.log("export done.");
  }, inputData);
});

document.getElementById('insert').addEventListener("click", function() {
  journalchooser.show(function(entry) {
    if (!entry) {
      return;
    }
    var dataentry = new datastore.DatastoreObject(entry.objectId);
    dataentry.loadAsText(function(err, metadata, data) {
      console.log(data);
    });
  }, {
    mimetype: 'image/png'
  }, {
    mimetype: 'image/jpeg'
  });
});
```

Sharing Activity: we will use presence here.

Once the user shares the activity, The skills and categories object array is sent from the host to every new user joining activity. But users can't make any changes to these skill's set i.e. user joining will be in read-only mode.

And whenever the host makes any change to its skill's set or validates any skill, all other users joined will get the updated info or array.

Host can see who are the users which joined the activity.

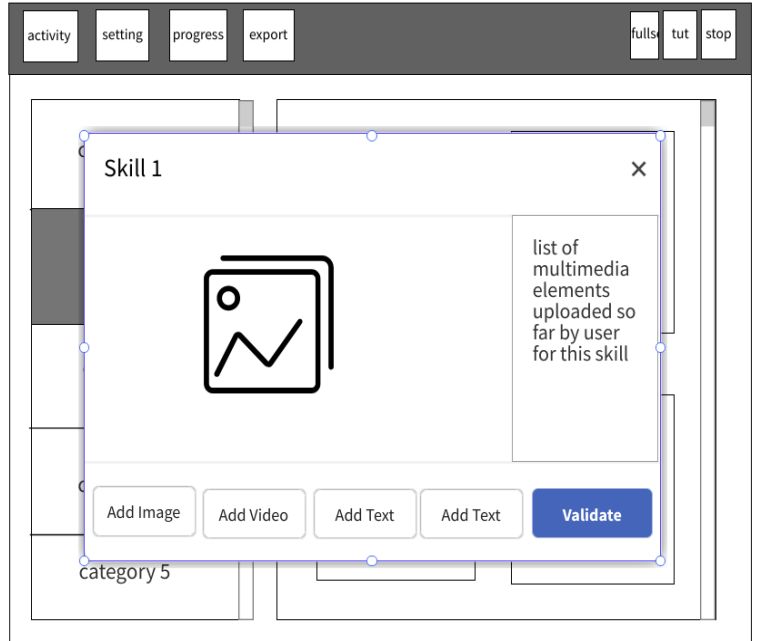
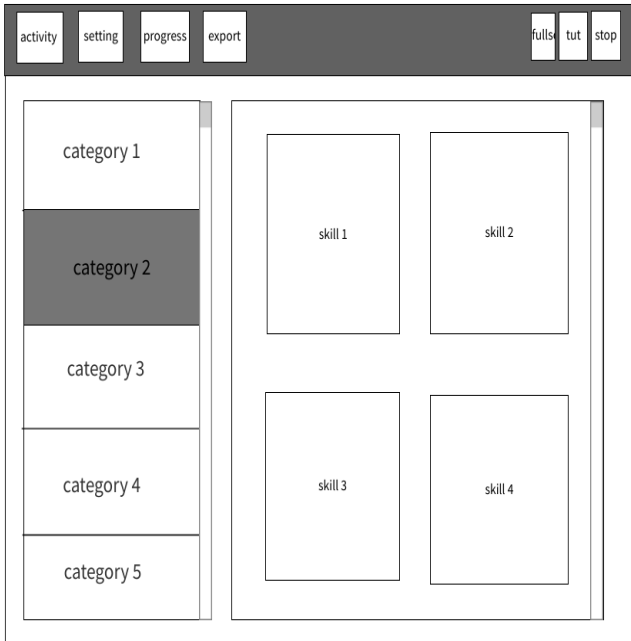
Presence actions:

- Init: for initializing read only mode and get activities from host
- Update: for updating the set of skills

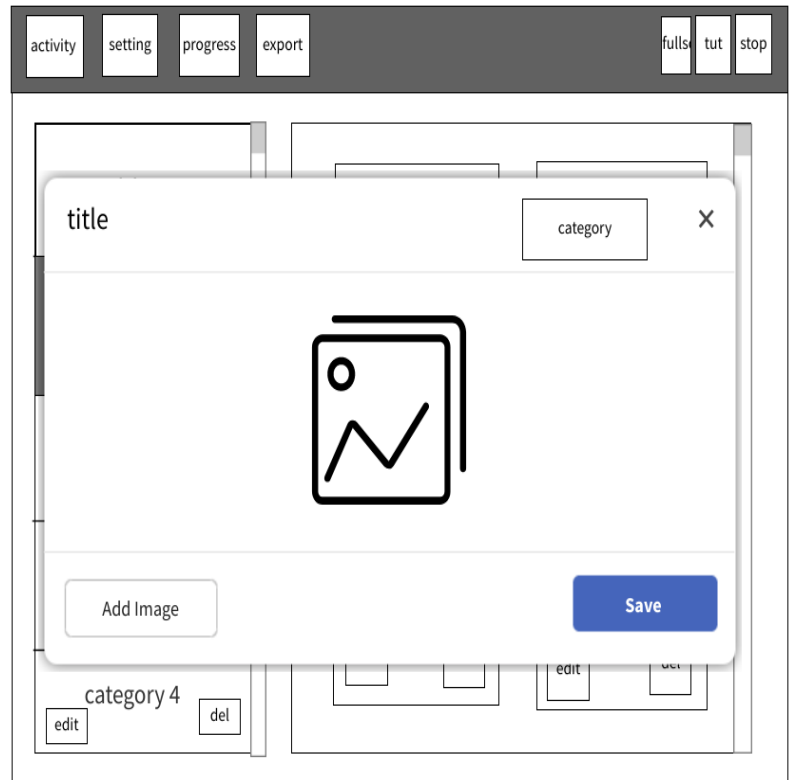
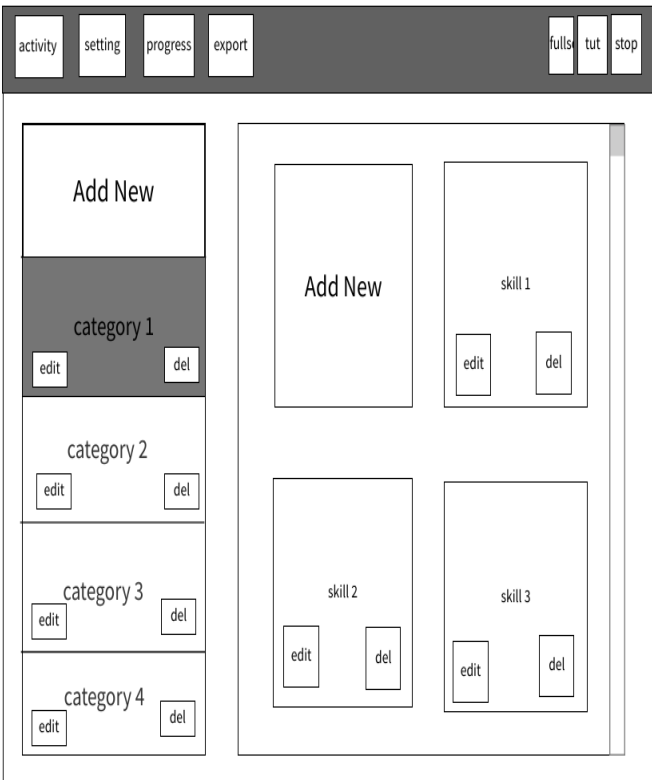
Colors: We will use buddy colors and category colors as much as possible.

Basic wireframe for ui(not finalized):

Home screen (play mode):



Setting screen:



Modal for creating new skill

Aim is to build simple ui as much as possible.

For framework for ui: we will use [vuejs](#)

For drag and drop used in sorting of skills: we will use [Vue.Draggable](#)

For building responsive design: we will use [bootstrap](#).

2) Vote Activity:

The Vote activity will allow easy building of a poll system. The user create a poll (yes/no, chooses a value in a list, enters a value) then shares it on the network so any user could vote in real time. At the end of the vote, a screen will sum up results of the vote: stats, who vote for what, who vote first, ...

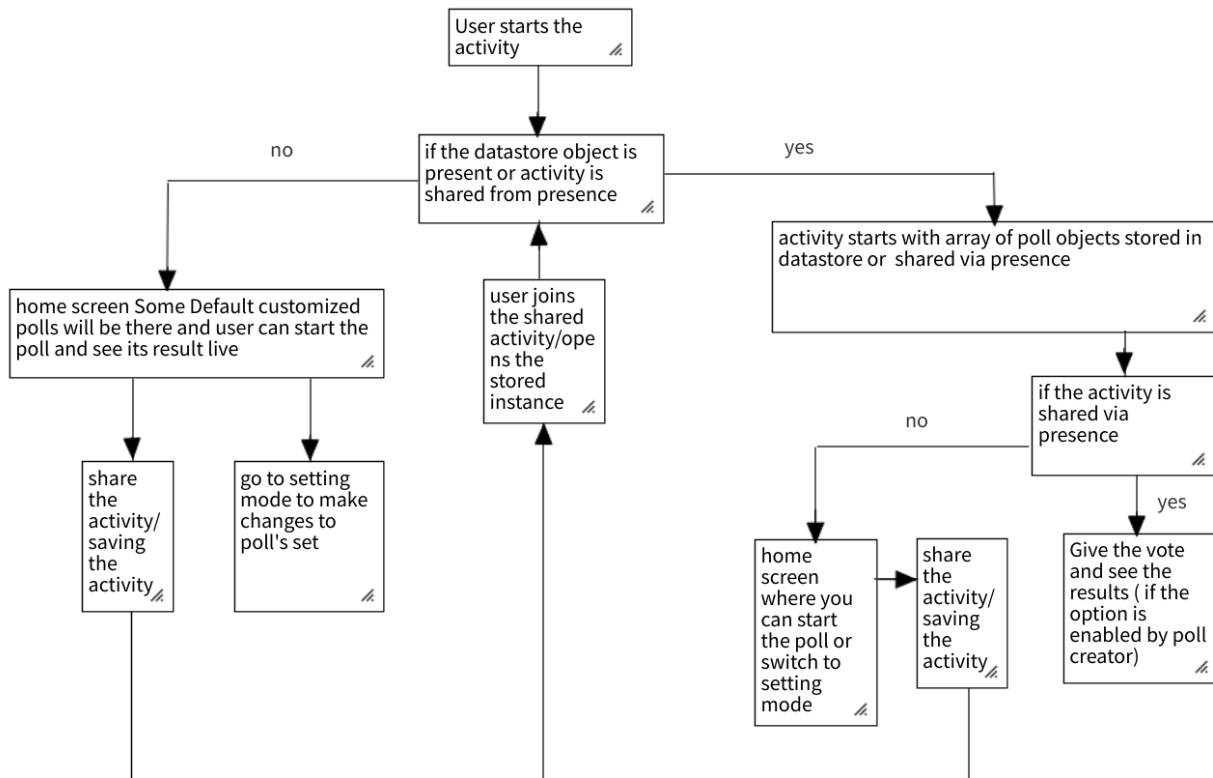
The activity will include following features:

1. We will have home screen in vote activity, in which,
 - activity will display a set of default polls with default templates and customization and users can quickly choose a poll and start a poll.
 - On each poll a user will have a button to start poll i.e. to share the poll on the network.
 - A poll will include:
 - question
 - thumbnail
 - Time period or expiry date/time of poll
 - Max number of votes allowed
 - Multimedia elements
 - Template
 - Flag showing whether users allowed to see live results immediately after their vote or not.
2. We will have setting screen, in which,
 - Users can create new polls and can customize it.
 - Users can edit/delete existing polls.
 - All the changes users have done to the set polls should persist with the help of a journal.
3. Users can share their poll to the network via presence and can see the live results which will include various info and stats and charts etc in the result screen.
4. Users joining the activity instance can't edit or create a new poll in that instance. They will have to act as a host or start new activity for the same.
5. Activity will use a journal to store the context.

6. Activity should be localized, responsive, should have a tutorial, have a fullscreen button
7. Activity should use buddy colors as much as possible.

Implementation:

The user flow is following:



Storing to activity instance: using journal

Datastore object will look like following:

(will include more fields also which will be decided later)

```

{
  polls : [
    {},
    {},
    ...
  ]
}

```

Poll object can be:

```
{
  template: "",
  ques : {},
  thumbnail : "",
  expiry date : "",
  immediateResult : false,
  multimedias : [
    {
      type : "",
      data : ""
    },
    {
      type : "",
      data : ""
    }
    ...
  ],
  state : "",
  userResponse : {}
}
```

Ques object will depend upon template, we will have following templates:

- Yes/No

```
{
  Q: ""
}
```

- Single/Multiple Response

```
{
  Q: "",
  a: "",
  b: "",
  c: "",
  ...
}
```

- Enter A value(open text)

```
{
  Q: ""
}
```

More templates can be added after GSOC.

We will use Lz-based compression algorithm to compress the data before storing it in journal using <https://github.com/pieroxy/lz-string> (as used in exerciser activity)

```
jsonData = LZ.compressToUTF16(jsonData);
```

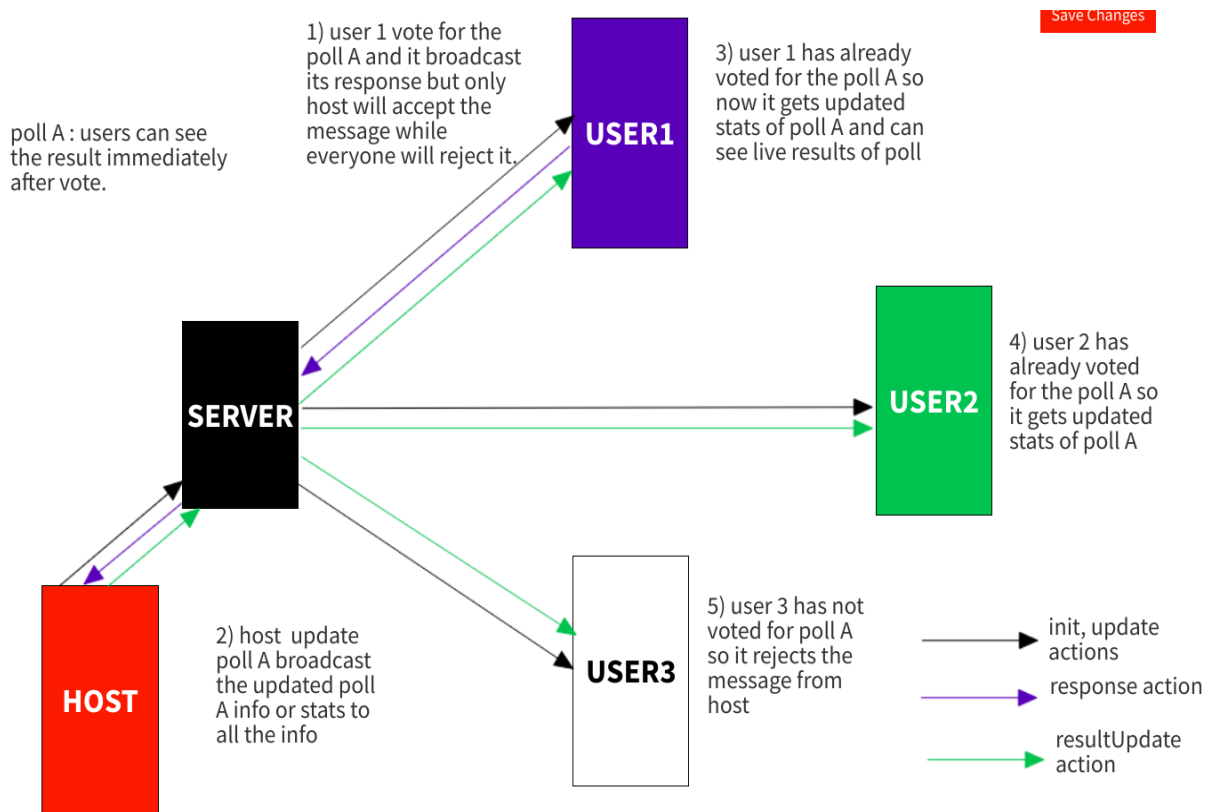
Sharing Activity: we will use presence here.

Once the user shares the poll, users joining the activity via presence will get shared polls objects(without userResponse fields). Then users can vote for the polls shared.

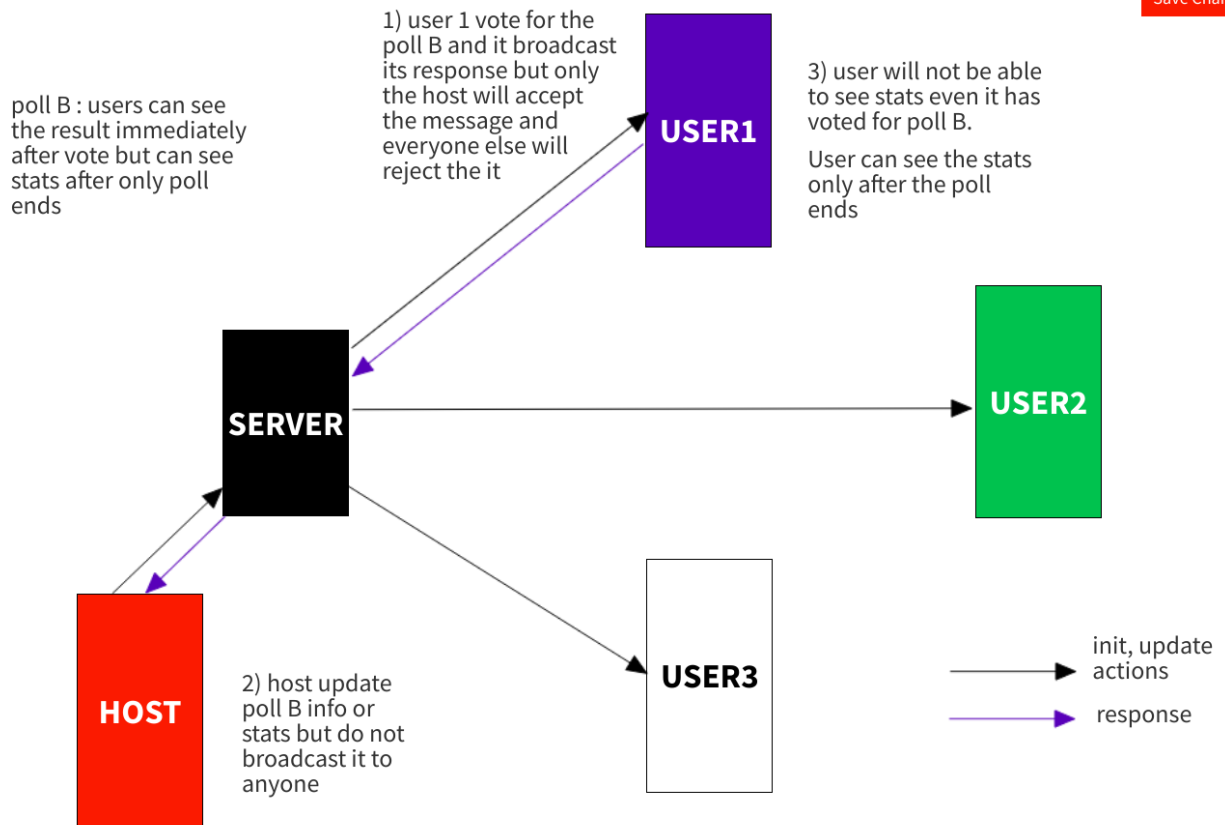
The flowing workflow takes place when a users vote for a poll

- Their response get back to the host
- Host changed stores the info
- If the immediate results option is enabled by the host for the poll, Host sends the updated info to all the users.
- The users who have voted for the activity get the info and update the state but the users who are still to vote will simply reject the data.

Case 1:



Case 2:



Host can see who are the users which joined the activity.

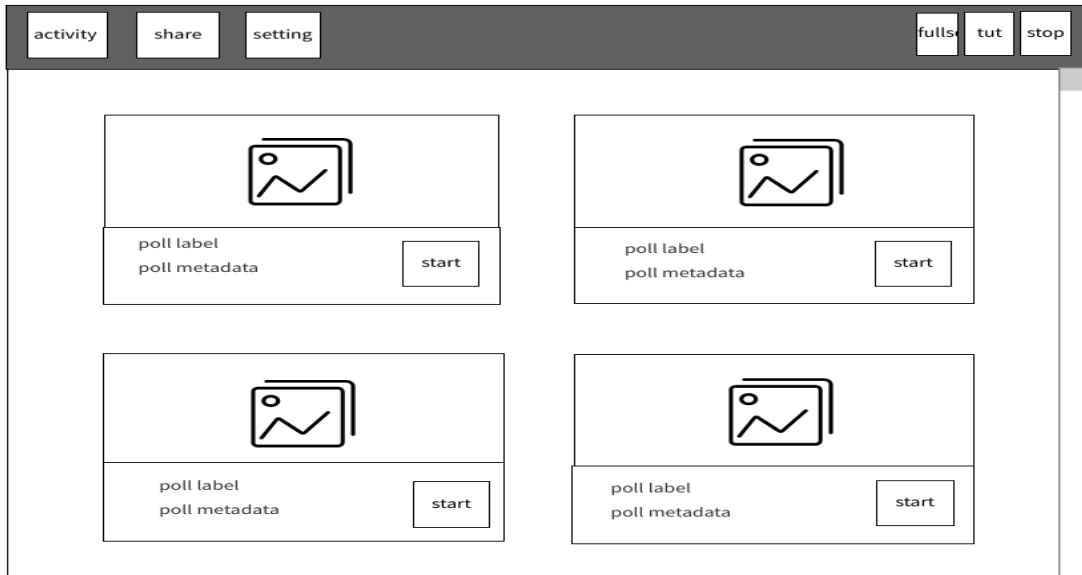
Presence actions:

- Init: for initializing read only mode and get poll's set from host
- Update: for updating the poll's set
- Response: for vote for a poll
- resultUpdate: for updating results or stats of a poll

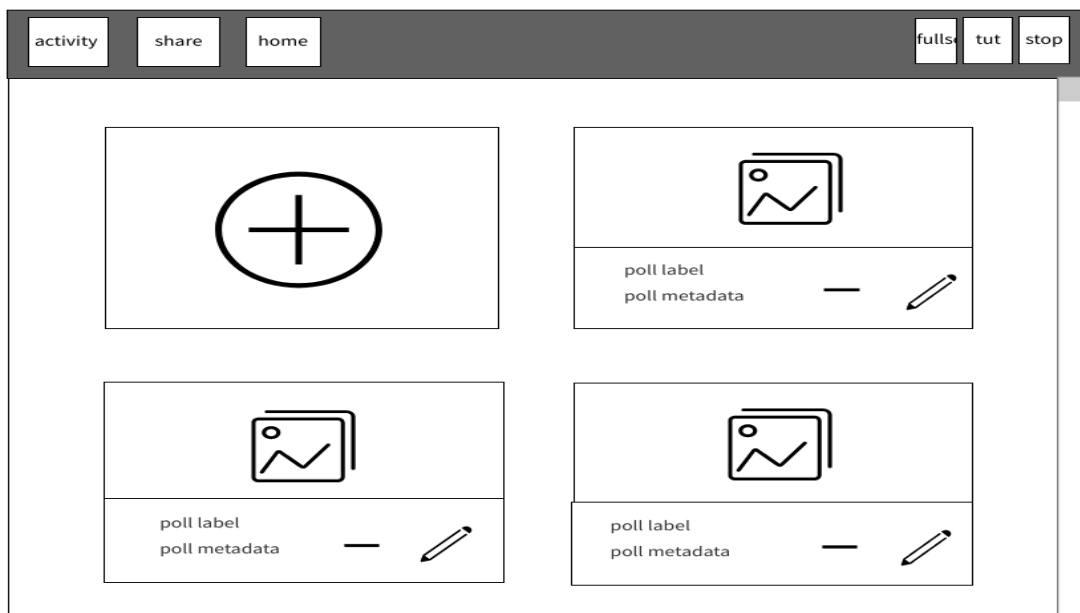
Basic wireframe for ui:

The aim is to have as simple ui as possible.

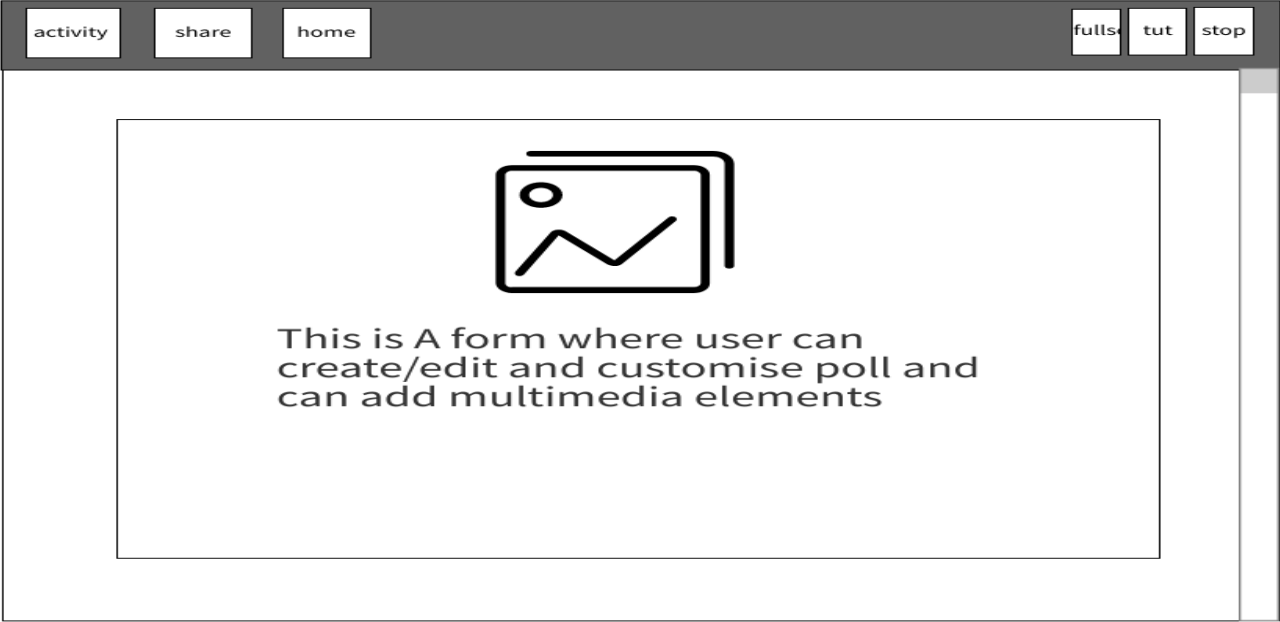
Home screen :



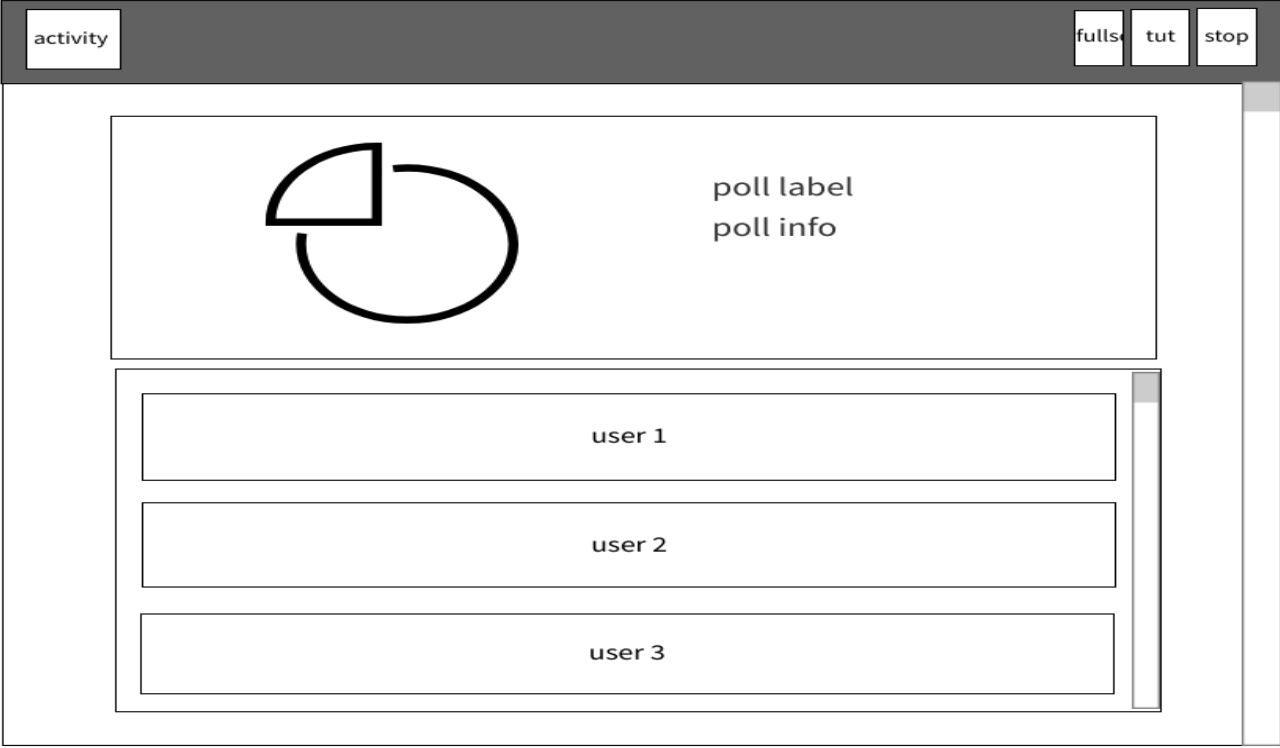
Setting screen:



New screen will open when we want to make changes to our poll's set

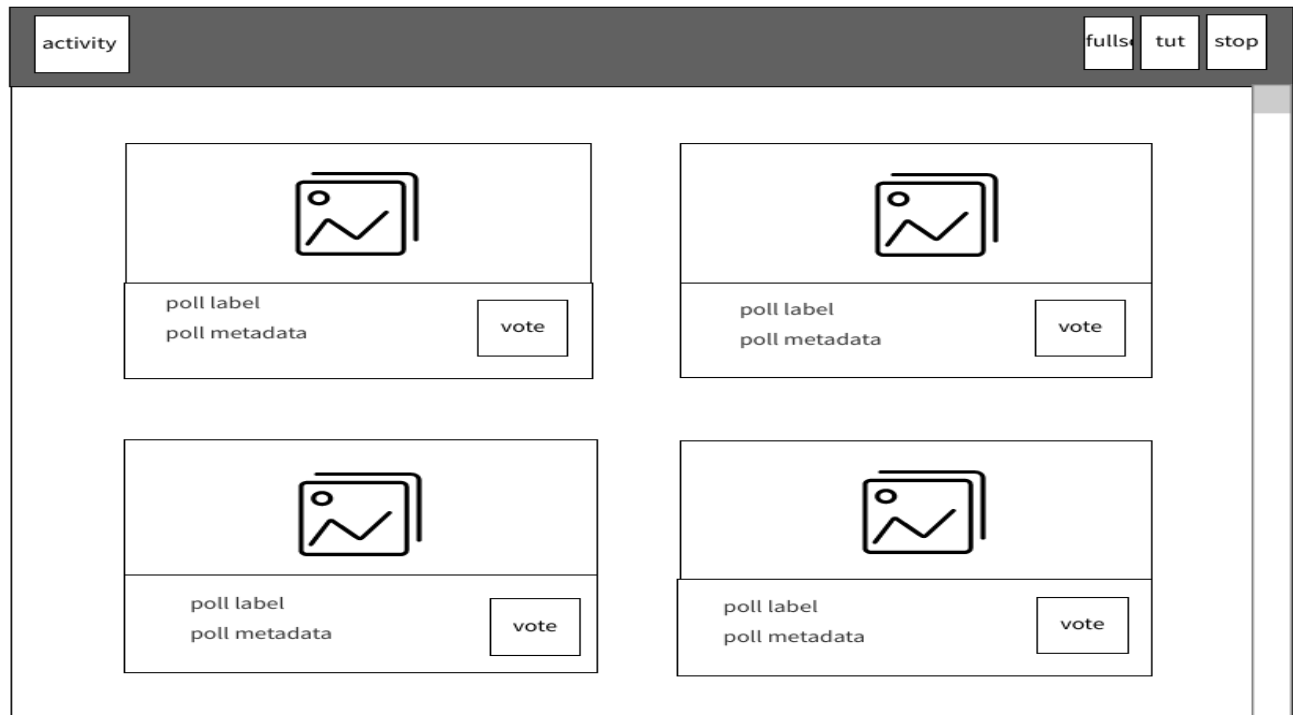


Stats screen which shows stats of a poll chosen :



In the poll stats screen, we will have bar charts, pie or donut charts.
Plus in case of “enter the value or open text” template we will have word cloud also.

When a user joins a activity he will have an option to vote polls plus will have an option to see results after the voting immediately or after vote period ends :



For framework for ui: we will use [vuejs](#)

For building responsive design: we will use [bootstrap](#).

For creating charts : we will use [Chartjs](#)

For creating word-cloud visualization: we will use [vueWordCloud](#)

How will it impact sugar labs ?

Both the activities are going to help children to engage themselves in learning new things and their education.

The curriculum activity will involve children in their evaluation so the child becomes the recipient in his own evaluation and has total control over his skills as the activities will have self declarative skills which need no validation from anyone except from the kid himself. The self evaluation will motivate children to put extra effort in their learning and will have less pressure.

The vote activity will help to build an interactive ecosystem for teachers and childrens. It allows users to build a poll in very easy and quick steps and allows others to vote for the poll in very easy steps. This will provide a very good alternative to traditional teachers-students questionnaires.

Also children can also use the activity among themselves.

The beauty of these activities lies in their simplicity which will be very helpful for children.

Addition of both the activities to sugarizer (a platform which is dedicated to help children) will promote the more involvement of children with sugarizer.

Mention how much time will you spend each week working on your project

I can easily spend 40-50 hours per week. And can extend my working hours if required. I will have to take a break of 10 days for my end sems examinations which were previously scheduled on 27 april 2020 but now due to covid-19 pandemic situation, exams can get rescheduled and I don't have any idea what will be the new schedule as it will depend on how better the situation will be in coming days.

Also discuss your plans after the GSOC period ends.

I will continue to work on this project post the completion of my GSOC tenure as an active contributor and will guide new contributors.

I will keep contributing to sugarizer and will start contributing to sugar python activities to enhance my python development skills and to learn something new.

I have packaged a few sugar applications into flatpak and will keep packaging more applications or activities.

The aim will be the growth and progress of the organization always.

Timeline:

DURATION	TASKS
27 april - 18 may	community binding period <ul style="list-style-type: none">● Explore the technologies and libraries that will be used in creating Curriculum and Vote Activities● Communicate with my mentors and other org members to finalize the features and UI of the Activity.
18 may - 25 may	<ul style="list-style-type: none">● Create a new Curriculum activity in Sugarizer and structure the project.● Start implementing the Curriculum activity.● Create skill and category object● Implement a home screen with cards and modals with default skill set.● build a toolbar.● Test● Write a blog
25 may - 1 june	<ul style="list-style-type: none">● Add journal chooser to home screen's modal pop ups to let users provide multimedia elements.● Add progress screen● Implement export to Doc/PDF feature● Start implementing setting screen and adding toolbar button to switch screen● Test● Write a blog
1 june - 8 june	<ul style="list-style-type: none">● Add modal pop ups in the settings screen to make changes to the skill's set or category's set.● Add a journal chooser in these modals to let users provide a skill image.● Add a toolbar button to let users retrieve skill's set by url.● Store and retrieve context from the journal.● Test● Write a blog
8 june - 15 june	<ul style="list-style-type: none">● Implement feature to share activity via presence

	<ul style="list-style-type: none"> ● Implement Read only mode which will used only when a user joins activity via presence ● Improve activity ● Implement localization via webl10n ● Test ● Write a blog
15 june - 19 june	<p>phase 1 evaluation</p> <p>progress:</p> <ul style="list-style-type: none"> ● All the Main features of Curriculum activity completed
19 june - 26 june	<ul style="list-style-type: none"> ● Make the required changes after evaluation ● Add tutorial ● Styling of activity ● Test activity on various platforms ● Try to find and fix bugs ● Create a new Vote activity in Sugarizer and structure the project ● Start implementing Vote Activity ● Write a blog
26 june - 3 july	<ul style="list-style-type: none"> ● Create poll object for each template ● Implement a home screen with cards with default polls. ● Build toolbar ● implement a setting screen with adding options to create/edit/delete polls. ● Test ● Write a blog
3 july - 13 july	<ul style="list-style-type: none"> ● Start Implement a poll customization screen where users can make changes to the poll's set. ● Implement all the poll templates(yes/no, single/multiple response, enter the value). ● Added the journal chooser to let users provide multimedia elements for creating polls. ● Store and retrieve context from the journal. ● Test ● Write a blog
13 july - 17 july	<p>phase 2 evaluation</p> <p>Progress:</p> <ul style="list-style-type: none"> ● Complete Curriculum activity

	<ul style="list-style-type: none"> ● Home, setting, customization screens of Vote Activity
17 july - 24 july	<ul style="list-style-type: none"> ● Make the required changes after evaluation. ● Implement features to share activity and polls over a network via presence. ● Start implementing the vote screen where users joined via presence can do the vote. ● In vote screen implement all the templates of poll (yes/no, single/multiple response, enter the value) ● Test ● Write a blog
24 july - 31 july	<ul style="list-style-type: none"> ● Start implementing a Poll stats screen. ● First implement the features in the stats screen which will be common for all each template. ● Implement charts/illustration for all polls :- yes/no, single/multiple, enter the value response type polls. ● Add features to allow users who joined activity via presence to see live results. ● Test ● Write a blog
31 july - 7 august	<ul style="list-style-type: none"> ● Implement localization via webl10n ● Add tutorial ● Styling of activity ● Test activity on various platforms ● Try to find and fix bugs ● Buffer time which can be used if some work left ● Write a blog
7 august - 10 august	<ul style="list-style-type: none"> ● Prepare documentation for both the activities ● Prepare for final evaluation
10 august - 17 august	<p>final evaluation</p> <ul style="list-style-type: none"> ● Both the activities completed ● Documentation of both the activities completed ● Both the activities tested thoroughly.

MISCELLANEOUS

- Google hashcode 2020 global rank: 1855, country rank: 216 [certificate](#)
- Set up sugar environment
- Set up sugarizer and sugarizer server environment
- Set up flatpak environment
- Completed GSOC 2020 task