

Google Summer of Code 2018 Proposal

Improve and create SUSI bots



Name :	Harshit Kedia
Country :	India
School and Degree :	Vellore Institute of Technology (B.Tech in Computer Science and Engineering)
Year at University :	Third Year
Email / Hangout :	harshitkedia32@gmail.com
Website :	http://harshitkedia.tech
Twitter :	https://twitter.com/Harshitkedia32
Phone :	+91 9585602122

Overview

There are a number of SUSI AI bots. Many of them are in their basic version and are not having full features of SUSI. Following are the major scope of improvements in these bots:

- Support of **multiple response** types like image, table, map, etc and format the reply using the platform's **markdown** language.
- Adding **personal assistant features** such as maintaining todo lists, maintaining assigned tasks list, getting songs, and getting latest news.
- User should be able to do **platform specific tasks** by interacting with SUSI. Such as adding the user to specific channels, getting list of all channels, and unsubscribing.
- Adding **games** using the platform specific integrations.
- Update obsolete APIs and fix documentation.
- Provide a **user friendly way** for the users to deploy the SUSI bots in their respective platforms.

This project aims at making significant improvements and adding personal assistant features in few SUSI AI bots such as [susi_slackbot](#), [susi_gitterbot](#), [susi_zulip_bot](#), [susi_telegrambot](#) and make them ready to use in production. The project will focus on the above bots firstly and then move on to other SUSI bots.

Project Goals

The project aims at solving existing issues and significantly improving many existing SUSI AI bots and take them to production level. Features which will be useful to the users and provide a more personal assistant experience will be added. The deployment process for the bots will be made simpler and well documented so that users can easily use the bots in their workspaces. The project will consist of improving the following SUSI bots firstly. Later, other SUSI bots will be examined and improved accordingly.

SUSI Slackbot

The [susi_slackbot](#) is in an early prototype stage. Further, the APIs used by the bot are no longer working. The documentation contains broken links and installation process is difficult to follow. *Slack is a very popular platform, and having a full working susi slackbot will be very useful.* This project aims to make a full personal assistant bot for slack.

Following improvements will be made:

- Make the bot working in development again. Change the SUSI APIs used and configure the slack bot.
- Update documentation, fix broken links, fix installation instructions.
- In slack channels, SUSI should reply to those messages only which mention susi bot. Currently she replies to all the messages in channels. Check for mention.
- SUSI should ignore messages from all other bots in common channels.

- Extract the correct query from slack and send it to susi server. Read the correct fields from the server's response and send it to slack api.
- Handle error cases. Currently, the app crashes if an error occurs.
- Currently the table response is not correctly structured. A new message is send for every table row. This should be changed, the table should be formatted into a single response using slack's markdown language.
- Fix the RSS type response. Read the correct fields from the response and use [slack's bot formatting](#) to format the response.

Following new features will be added:

- Add support for hyperlink response and format the reply properly.
- Add support for map type response (Eg. "@susiai Where is India?").
- Add a default response like "Hmm... I'm not sure if i understand you correctly." for gibberish inputs or when the SUSI server fails to answer.
- Add "tasks assign" feature. Users in Slack should be able to maintain a tasks list with the help of SUSI. Eg. "@susiai /assign 'Improve blog' to @akshit". **These features would make SUSI a powerful personal slack assistant.**
- Add /help feature. Users should be able to ask for help from SUSI bot with a command like "@susiai /help" or "@susiai /start", which will then list all the available commands and features.
- Provide feature for creating polls. Eg. @susiai /poll "which language you like?" "option1" "option2" "option 3" "option 4"
- Provide feature for users to maintain individual todo lists. (Eg. "@susiai /add bread to grocery list")
- Add reminder feature. Users can set reminders with SUSI bot.
- Add feature for playing games and quizzes by interacting with SUSI.
- Make SUSI reply to only text messages and for other types of messages such as file attachment, reply for eg. "Sorry, I can't understand that"
- Add introductory message for SUSI. Add profile and description. Make it more standardised like other slack bots.
- More ideas can be taken from other existing slack bots and make SUSI a powerful slack assistant.
- Deploy SUSI Slack bot to [Slack App Directory](#). This will enable single click "Add to slack" button for all users.

SUSI Gitterbot

The [susi_gitterbot](#) is in a basic prototype stage. It currently does not handle multiple actions response. The susi bot user name is hardcoded, therefore it will not work when installed with some other username. The [deployed bot](#) is no longer working. The instructions for a user to add susibot to their chat room is very difficult to follow. This project aims at removing the existing

bugs in the gitter bot, adding new features, and taking it to production level so that any user can easily install it in their gitter chat rooms.

Following improvements will be made:

- Remove static username instead, extract the username from [gitter API](#).
- Support multiple action types. Currently, the bot replies with only one of the actions.
- Respond to only those messages which tag susi bot anywhere in the message and not just in the starting. Remove the susi tag and send the query to susi server.
- Fix documentation. Change deployment instructions. Deploy in a new gitter chat room.

Following new features will be added:

- Add support anchor and map type responses
- Add a default response like “*Hmm... I’m not sure if i understand you correctly.*” for gibberish inputs or when the SUSI server fails to answer.
- Add “*/help*” feature. User should be able to get all the features of SUSI with this command.
- Provide features for playing games and quizzes.
- Provide feature for users to maintain individual todo lists. (Eg. “*@susiai /add bread to grocery list*”)
- Provide a **user friendly way to deploy SUSI gitter chatbot**. Currently deploying the chatbot involves manually using the gitter API and setting the environment variables in heroku. User should be able to easily deploy SUSI chatbot in their gitter rooms with the help of a User Interface (explained below).

SUSI Zulip bot

For developing bots for Zulip platform, we can refer the guide zulipchat.com/api/running-bots. It requires us to deploy our susi bot in [python-zulip-api](#) repository. We already have the susi zulip bot [here](#) in a very basic version. This project aims at adding support for more types of responses, adding todo list feature and other Zulip related actions, adding test cases, deploying the bot in production and making documentation for the same.

Following features will be added:

- Add support for anchor, image, map, and table type of responses.
- Format responses using Zulip’s [markdown language](#). For formatting tables, use [this markdown](#).
- Add *todo list* feature. Users can maintain a personal todo list. (Eg. “*@susiai /add bread to grocery list*”). Use zulip [default storage](#) for this.
- Provide subscribed channel lists to the user. (Eg. “*@susiai /show subscribed streams*”). Use [this api](#) to get the channel lists.
- Users can get summary of the previous messages in a particular topic by asking SUSI.
- Provide feature for playing games and quizzes by interacting with susi.
- Users can subscribe to specific streams by telling susi. (Eg. “*@susiai /subscribe to design*”). Use [this api](#) to subscribe to the zulip channel.

- Users can unsubscribe themselves from specific streams by telling susi. (Eg. “@susiai /unsubscribe to design”). Use [this api](#) to subscribe to the zulip channel.
- Users can get list of all users by asking susi. Use [this api](#). (Eg. “@susiai /show list of all users”)
- Deploy in production using [Zulip bot server](#).
- Write [test cases](#) for the bot.

SUSI Telegram bot

The [susi_telegrambot](#) is currently not working. The deployed bot is not responding. The APIs used by the bot are obsolete. Though, the bot does support all the response types and works good when deployed after making some minor changes. However, it is missing the advanced features of a “personal assistant” bot. More features can be added to make the bot interactive and more useful to the user. Further, telegram specific features such as adding inline functionality, showing default response, location request, and games can be added to increase the usability and popularity of SUSI telegram bot. This project aims at removing the bugs, adding functionality, improving documentation, and developing the telegram bot into a fully featured application.

Following features will be added:

- Enable [inline mode](#). Users should be able to talk with SUSI within any chat.
- SUSI currently doesn't handle the case when the user asks “Where am i?”. In such a case, first request [user's location](#) and then show the response from the server.
- Make [interactive game](#) using telegram's platform. Users should be able to play games by asking SUSI (Eg. “/start game quiz”).
- Make todo list feature. Users should be able to store and retrieve things by interacting with SUSI. (Eg. “/add bread to shopping list”)
- Make reminder feature. Users can set a reminder and SUSI will message them at the given time. (Eg. “/remind me to study at 5pm today”). We can use [node-schedule](#) for this.
- Make [custom keyboards](#) and [default commands](#). This will help user to easily give responses and search for SUSI's features.

Project Implementation

Following are the technical details and demonstrations of implementing some of the features mentioned above. More functionalities will be improved and features will be implemented in a robust manner in the actual project.

SUSI Slackbot

The API the [slackbot](#) currently uses is <http://api.asksusi.com/susi/chat.json> which is obsolete. It should be changed to <https://api.susi.ai/susi/chat.json> with proper parameters. A [local development environment](#) will be set using [ngrok](#) for speeding up the development and testing process. Also, an automatic deployment to heroku will be set up. A blog post can be written on how to set up slackbot locally using [ngrok](#).

Making SUSI reply to only mentions in channels or any message in direct message. Also extract correct query:


Save SUSI bot id in memory:


```
var rtm = new RtmClient(token, { logLevel: 'info',dataStore:
false,useRtmConnect: true,});
rtm.start();
var appData={};
rtm.on(CLIENT_EVENTS.RTM.AUTHENTICATED, (connectData) => {
  // Cache the data necessary for this app in memory
  appData.selfId = connectData.self.id;
});
```


Check SUSI bot id in messages:


```
rtm.on(RTM_EVENTS.MESSAGE, function(message) {
  var channel = message.channel;
  var text=message.text;
  //send reply only when mentioned or in direct message
  if(text.indexOf(appData.selfId)!=-1 || channel.startsWith("D")){
    //remove susi's mentioned before sending the query to susi server
    var susi_mention="<@"+appData.selfId+">";
    var text=text.replace(susi_mention,"");
    getAnswerFromSusi(text,channel);
  }
});
```


Result:


 **Derick** 1:34 PM
hi


 **hkedia321** 1:34 PM
Hi Derick, how are you?


 **Derick** 1:34 PM
I am good! thanks.


 **hkedia321** 1:35 PM
[@susitest-hkedia321](#) Who is donald trump?

 **susitest-hkedia321** APP 1:35 PM
full name | Donald John Trump
date of birth | Friday, June 14, 1946 (age: 71 years)
place of birth | New York City, New York, United States

 **hkedia321** 1:36 PM
[@susitest-hkedia321](#) Thanks!

 **susitest-hkedia321** APP 1:36 PM
Glad to hear it.

 **hkedia321** 2:42 PM
[@susitest-hkedia321](#) calculate 4+9?

 **susitest-hkedia321** APP 2:42 PM
4+9 = 13

Fixing anchor, map, and table responses. Process all action types:

```
//sending request to SUSI API for response
request(options, function(error, response, body) {
  if (error){
    msg = "Oops, looks like SUSI is taking a break, She will be back soon";
    rtm.sendMessage(msg, channel);
    return 0;
  }
  var res=(JSON.parse(body));
  var actions=res.answers[0].actions;
  for(var action_index=0;action_index<actions.length;action_index++){
    var action=actions[action_index];
    var type=action.type;
```

```

var answers=[];
var count=0;
var expression="";
if(type==="answer"){
    expression=action.expression;
    if(expression){
        rtm.sendMessage(expression, channel);
    }
}
else if(type==="anchor"){
    var text=action.text;
    var link=action.link;
    rtm.sendMessage(text+": "+link, channel);
}
else if(type==="table"){
    var tableData = res.answers[0].data;
    var columns = Object.keys(action.columns);
    var columnsData = Object.values(action.columns);
    var msgs="";
    for (var i = 0; i < tableData.length ; i++) {
        if(tableData[i][columns[0]]){
            msg = "*" + tableData[i][columns[0]] + "*" + ", " +
tableData[i][columns[1]] + "\n" + tableData[i][columns[2]] + "\n ";
            msgs=msgs+msg;
        }
    }
    rtm.sendMessage(msgs, channel);
}
}
})


```



Result:

hkedia321 4:49 PM
@susitest-hkedia321 where is India?

susitest-hkedia321 APP 4:49 PM
India, officially the Republic of India (Bhārat Gaṇarājya), is a country in South Asia. It is the seventh-largest country by area, the second-most populous country (with over 1.2 billion people), and the most populous democracy in the world.

4:49 PM Here is a map:<https://www.openstreetmap.org/#map=13/45.04816050876312/20.08164970208034>

 openstreetmap.org
OpenStreetMap
OpenStreetMap is a map of the world, created by people like you and free to use under an open license. (71 kB) ▾



hkedia321 4:52 PM
@susitest-hkedia321 universities in India

susitest-hkedia321 APP 4:52 PM
Sure, here is a list of universities

4:52 PM **University of Health Sciences Andhra Pradesh, India**
<http://59.163.116.210/>
Allahabad Agricultural Institute, Deemed University, India
<http://www.aaidu.org/>
Assam Agricultural University, India
<http://www.aau.ac.in/>
Ahmedabad University, India
<http://www.ahduni.edu.in/>
All India Institute of Medical Sciences, India
<http://www.aiims.ac.in/>
AISECT University, India
<http://www.aisectuniversity.ac.in/>
Alagappa University, India
<http://www.alagappauniversity.ac.in/>
Allahabad University, India

Implement todo list feature

```
function handleTodoList(text,message){
    var text=text.replace("/add","");
    var index_to=text.indexOf("to");
    if(index_to==-1){
        rtm.sendMessage("/add command must have a 'to' keyword.",
```

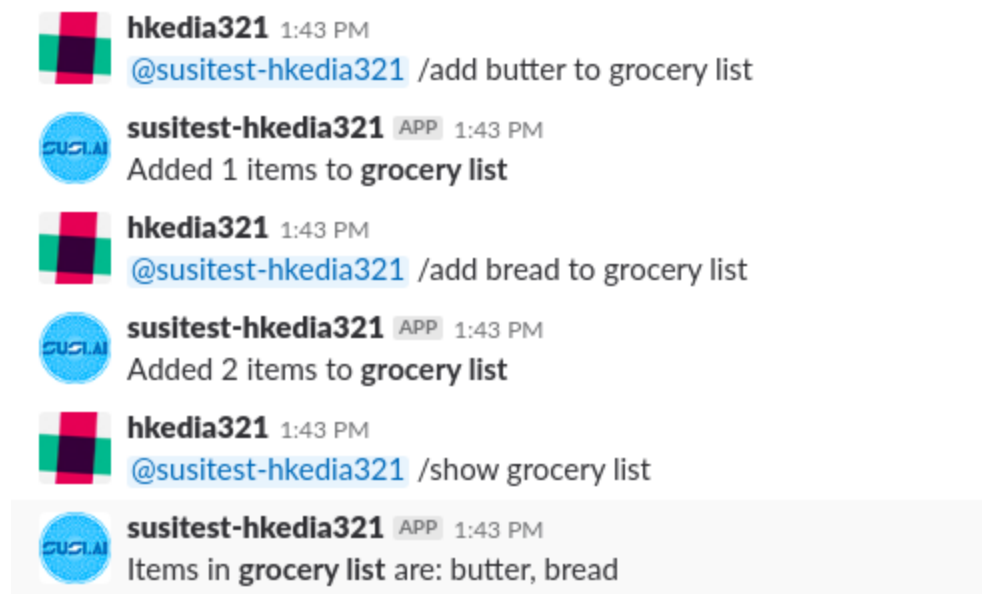
```

message.channel);
    return 0;
}
var item=text.substr(0,index_to).trim();
var list=text.substr(index_to+2).trim();
if(todo_lists[list]){// already exists
    todo_lists[list].push(item);
}
else{
    todo_lists[list]=Array(item);
}
rtm.sendMessage("Added "+todo_lists[list].length+" items to
*"+list+"*",message.channel);
}
function handleShowTodoList(text,message){
    var text=text.replace("/show","");

    var list=text.trim();
    if(todo_lists[list]){// list exists
        var items="";
        var l=todo_lists[list].length;
        for(var i=0;i<l;i++){
            if(i==0){
                items+=todo_lists[list][i];
            }
            else{
                items+=", "+todo_lists[list][i];
            }
        }
        rtm.sendMessage("Items in *"+list+"* are: "+items,message.channel);
    }
    else{
        rtm.sendMessage("No items found in *"+list+"*",message.channel);
    }
}
}

```

Result:



SUSI Gitterbot

SUSI Gitterbot allows user to add SUSI bot as a user in gitter chat rooms. We use [gitter API](#) to automate the responses of the bot. A local development environment will be set using [ngrok](#) for speeding up the development and testing process. Also, an automatic deployment to heroku will be set up. The issues in the current project will be solved and new features will be added to make SUSI a powerful chatbot and assistant for gitter. Also a user friendly way will be devised to allow users to easily deploy SUSI bot to their gitter rooms.

Remove static username extract username from gitter API:

```
var susi_username="";  
//make api call to get user name  
var gitterOptionsUsername = {  
  method: 'GET',  
  url: 'https://api.gitter.im/v1/user',  
  headers:  
  {  
    'authorization': 'Bearer '+ token ,  
    'content-type': 'application/json',  
    'accept': 'application/json'
```

```

    },
    json: true
  };
  // making the request to Gitter API
  request(gitterOptionsUsername, function (error, response, body) {
    if(error)
      throw new Error(error);
    susi_username=body[0].username;
  });

```

Result:

SUSI works from account of any username

	Harshit Kedia @hkedia321 @susiaitest Hello!	14:47
	susiaitest @susiaitest Hello!	14:48

Support multiple action types, add anchor and map type of response. Reformat table response.

```

// fetching the answer from Susi's response
var actions=data.answers[0].actions;
for(var
action_index=0;action_index<actions.length;action_index++){
  var action=actions[action_index];
  var type=action.type;
  if(type==="answer"){
    ans+=action.expression+"\n";
  }
  else if(type==="anchor"){
    var text=action.text;
    var link=action.link;
    ans+="["+text+"]("+link+")"+"\\n";
  }
  if(type === 'table'){
    var tableData = data.answers[0].data;
    var max_rows=50;
    var msg="";
    var columns = Object.keys(action.columns);

```

```

        var columnsData = Object.values(action.columns);
        for (var i = 0; i <
(tableData.length>max_rows?max_rows:tableData.length); i++) {
            if(tableData[i][columns[0]]){
                msg = (i+1)+" . *" +tableData[i][columns[0]]+"*" + ", " +
tableData[i][columns[1]] + "\n" + tableData[i][columns[2]]+ "\n ";
                ans=ans+msg+"\n";
            }
        }
        ans+="\n";
    }
}
sendReply(ans);

```

Result:



Harshit Kedia @hkedia321
@susiaitest where is India?

14:58



susiaitest @susiaitest
India, officially the Republic of India (Bhārat Gaṇarājya), is a country in South Asia. It is the seventh-largest country by area, the second-most populous country (with over 1.2 billion people), and the most populous democracy in the world.
[Here is a map](#)

14:58



Harshit Kedia @hkedia321
@susiaitest Who is Jimmy Wales?

14:58



susiaitest @susiaitest
full name | Jimmy Donal Wales
date of birth | Sunday, August 7, 1966 (age: 51 years)
place of birth | Huntsville, Alabama, United States

14:58



Harshit Kedia @hkedia321
@susiaitest recipe list for chicken

14:59



susiaitest @susiaitest


14:59 ✓ ...

1. *Home Fried Chicken Recipe*, milk, bacon grease, flour, potato, chicken
<http://www.grouprecipes.com/11241/home-fried-chicken.html>
2. *Chicken in a Pot Plus Stock or Dumplings*, allspice, bay leaf, carrot, celery, thyme, onions, parsley, black pepper, salt, salt, water, chicken
<http://www.recipezaar.com/Chicken-in-a-Pot-Plus-Stock-or-Dumplings-115070>
3. *Tipsy Chicken*, beer, cayenne, chili powder, cumin, curry powder, garlic powder, wood chips, onion powder, paprika, black pepper, salt, salt, water
<http://allrecipes.com/Recipe/Tipsy-Chicken/Detail.aspx>
4. *2 in 1 - Herbed Chicken & Leftover Chicken Salad*, chicken, garlic, margarine, marjoram, onions, black pepper, rosemary, sage, salt, thyme
<http://www.recipezaar.com/2-in-1-Herbed-Chicken-Leftover-Chicken-Salad-108955>
5. *Beer Can Chicken*, beer, brown sugar, salt, seasoning, chicken
<http://www.recipezaar.com/Beer-Can-Chicken-312508>

Making a User Interface for easy deployment of SUSI chatbot on gitter rooms

Currently the [instructions to setup](#) SUSI chatbot in your gitter rooms is long and difficult to follow. A User Interface (website) will be made to ease this process. User will be able to login to Gitter, select their room to which they want to add SUSI, and then deploy the chatbot to heroku with a single button and automatic configuration.

Mockup of the landing page:



LOGIN WITH GITTER

SUSI Gitter Chatbot

Meet SUSI.AI, Your Artificial Intelligence.
Your Personal Assistant and Chatbots.

Ask it questions. Tell it to do things. Always ready to help.

tortillas, jalapeno, mozzarella cheese, onions, tomato, vegetable oil
<http://www.recipezaar.com/Chicken-Quesadillas-Homemade-200392>

Harshit Kedia @hkedia321
@susiaitest thanks! 00:31

susiaitest @susiaitest 00:31
Glad to hear it.

Harshit Kedia @hkedia321 00:31
@susiaitest /add blog to todo-list

susiaitest @susiaitest 00:31
Added 1 items to todo-list

Harshit Kedia @hkedia321 00:31
@susiaitest /show todo-list

susiaitest @susiaitest 00:31
Items in todo-list are: blog

Harshit Kedia @hkedia321 00:31
@susiaitest thanks!

@ I

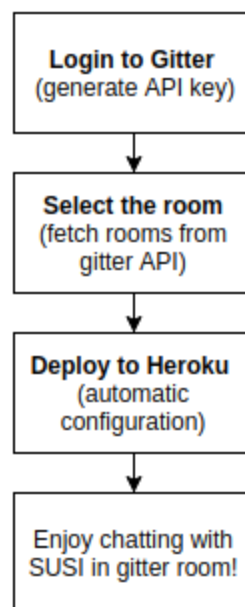
Steps to follow:

- Make a gitter account for SUSI bot. This account will be used for interacting with SUSI.
- Sign in to the gitter account using the below button. This will generate a token for authentication to be used by SUSI bot
- Select the chatroom you want SUSI to join
- Deploy your app to Heroku using the below button (visible after you sign into gitter). Make sure you already have a heroku account.
- Thats it! You are all set to interact with SUSI in your gitter room!

LOGIN WITH GITTER

Deploy to Heroku

Flow chart of the process:



SUSI Zulip bot

This project aims at improving the existing [susi zulip bot](#), which is in a rudimentary stage. It just supports a simple text response for now. Though it is not necessary to have the bot in zulip's repository for testing and developing purposes, I think for the final bot, we must have the bot in zulip's repository. We can directly contribute to the Zulip's repository or maybe we can have a

fossasia repository first for susi zulip bot and then update in Zulip's repository. I wish to discuss this with mentors.


Following are the implementation results of some of the goals of the SUSI Zulip bot.


Handle multiple response type and use Zulip's markdown language.


```
for action in actions:
    if action['type'] == 'answer':
        answer+=action['expression']+'\n'
    elif action['type'] == 'anchor':
        text=action['text'];
        link=action['link'];
        answer+='['+text+']('+link+')'+'\n';
    elif action['type'] == 'table':
        tableData = data['answers'][0]['data']
        msg=""
        columns = list(action['columns'].keys())
        columnsData = list(action['columns'].values())
        # using markdown make a table
        for heading in columnsData:
            answer+="| "+heading+" "
        answer+="\n"
        for heading in columns:
            answer+="| "+"-----"+" "
        answer+="\n"
        for i in (tableData):
            print(i)
            if(i[columns[0]]):
                msg = "| "+i[columns[0]]+" | " + i[columns[1]]
            + " | " + i[columns[2]]+" | ";
            answer=answer+msg+"\n";
        answer+="\n";
```



Result:


You and SUSI AI


 **Harshit Kedia** 1:44 PM
hi

 **SUSI AI** 1:45 PM
Hi, I'm Susi

 **Harshit Kedia** 1:45 PM
where is India?

 **SUSI AI** 1:45 PM
India, officially the Republic of India (Bhārat Gaṇarājya), is a country in South Asia. It is the seventh-largest country by area, the second-most populous country (with over 1.2 billion people), and the most populous democracy in the world.
[Here is a map](#)

 **Harshit Kedia** 1:45 PM
recipe list for chicken

 **SUSI AI** 1:45 PM

Recipe	Instructions Link	Ingredients
Home Fried Chicken Recipe	http://www.groupprecipes.com/11241/home-fried-chicken.html	milk, bacon grease, flour, potato, chicken
Chicken in a Pot Plus Stock or Dumplings	http://www.recipezaar.com/Chicken-in-a-Pot-Plus-Stock-or-Dumplings-115070	allspice, bay leaf, carrot, celery, thyme, onions, parsley, black pepper, salt, salt, water, chicken

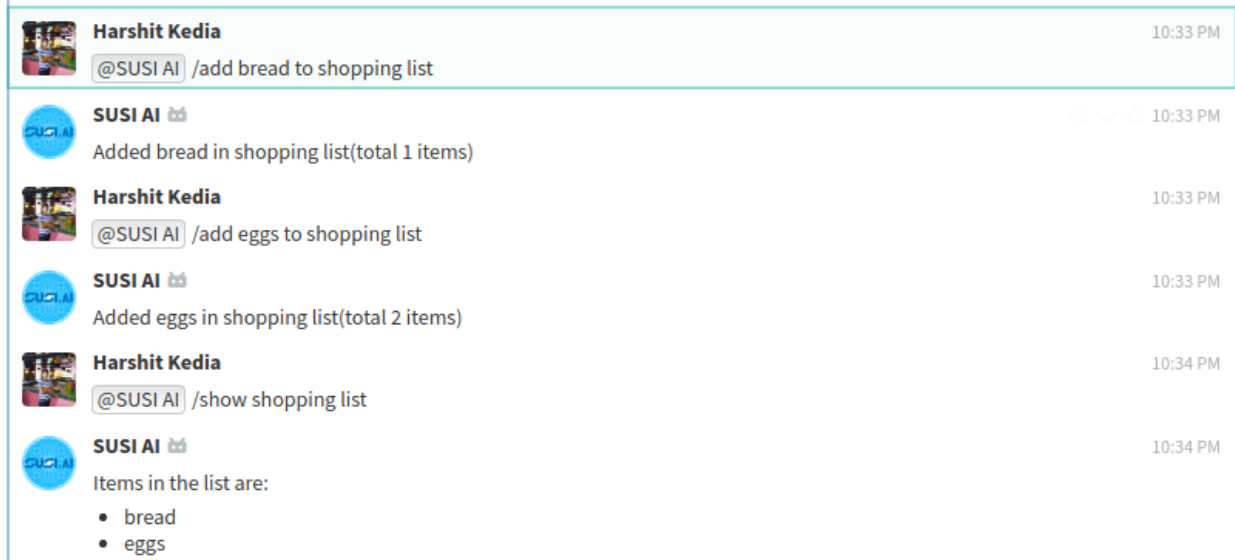
[\[More...\]](#)

Add todo list feature

```
def handle_todo_add(self,message: Dict[str, str], bot_handler: Any) ->
None:
    try:
        msg = message['content']
        to_index = msg.find('to')
        list_name = msg[to_index+3:]
        item = msg[5:to_index]
        if bot_handler.storage.contains(list_name) :
            prev_cont = eval(bot_handler.storage.get(list_name))
            prev_cont.append(item)
            l=len(prev_cont)
            bot_handler.storage.put(list_name, str(prev_cont))
        else :
            list_cont=[]
            list_cont.append(item)
            l=len(list_cont)
            bot_handler.storage.put(list_name, str(list_cont))
        answer = "Added "+item+" in "+list_name + "(total "+str(l)+"
items)"
    except Exception:
```

```
        answer = "Please use the correct structure for adding items in  
your lists"  
        bot_handler.send_reply(message, answer)
```

Result:



The screenshot shows a chat interface with the following messages:

- Harshit Kedia** (10:33 PM): @SUSI AI /add bread to shopping list
- SUSI AI** (10:33 PM): Added bread in shopping list(total 1 items)
- Harshit Kedia** (10:33 PM): @SUSI AI /add eggs to shopping list
- SUSI AI** (10:33 PM): Added eggs in shopping list(total 2 items)
- Harshit Kedia** (10:34 PM): @SUSI AI /show shopping list
- SUSI AI** (10:34 PM): Items in the list are:
 - bread
 - eggs

Add subscribe feature:









```
#!/usr/bin/env python3  
import zulip  
client = zulip.Client(config_file="./zuliprc-susi")  
def handle_subscribe(self,message: Dict[str, str], bot_handler: Any) ->  
None:  
    msg = message['content']  
    sender_email = message['sender_email']  
    channel_name = msg[11:]  
    result = client.add_subscriptions(  
        streams=[  
            {'name': channel_name}  
        ],  
        principals=[sender_email]  
    )  
    print(result)  
    ans = "Subscribed to "+channel_name  
    bot_handler.send_reply(message, ans)
```

Result:

	Harshit Kedia @SUSI AI /subscribe production help	11:22 PM
	SUSI AI Subscribed to production help	11:22 PM
	Harshit Kedia @SUSI AI /subscribe learning	11:22 PM
	SUSI AI Subscribed to learning	11:22 PM

Similarly other features will be added as well.

I deployed the SUSI Zulip bot for testing in [zulip bot testing channel](#) and received feedback from Zulip maintainers as well:

bot testing	susi		Mar 08
	Steve Howell @SUSI AI when is halloween?		8:00 PM
bot testing	susi		
	SUSI AI Wednesday, October 31, 2018		8:00 PM
	Steve Howell @SUSI AI who is lebron james?		8:00 PM
	SUSI AI full name LeBron James date of birth Sunday, December 30, 1984 (age: 33 years) place of birth Akron, Ohio, United States		8:00 PM
	Steve Howell so far so good, @Harshit Kedia !		8:01 PM
	Harshit Kedia (EDITED) @Steve Howell Yes! The bot does work for basic queries. However, I am planning to add more functionalities to the bot so that it can support more kinds of responses like <i>table</i> , <i>anchor</i> , <i>map</i> , etc.		8:02 PM
	@SUSI AI flip a coin		8:05 PM
	SUSI AI tails		8:05 PM
	Steve Howell sounds good!		8:13 PM

SUSI Telegram bot

This project aims at adding advance features in the [susi_telegrambot](#). SUSI telegrambot enables users to interact with SUSI within telegram app. We will be using [telegraf](#). Telegraf is a node library which helps to interact with the Telegram bot. It offers advanced features such inline response, creating games, etc.

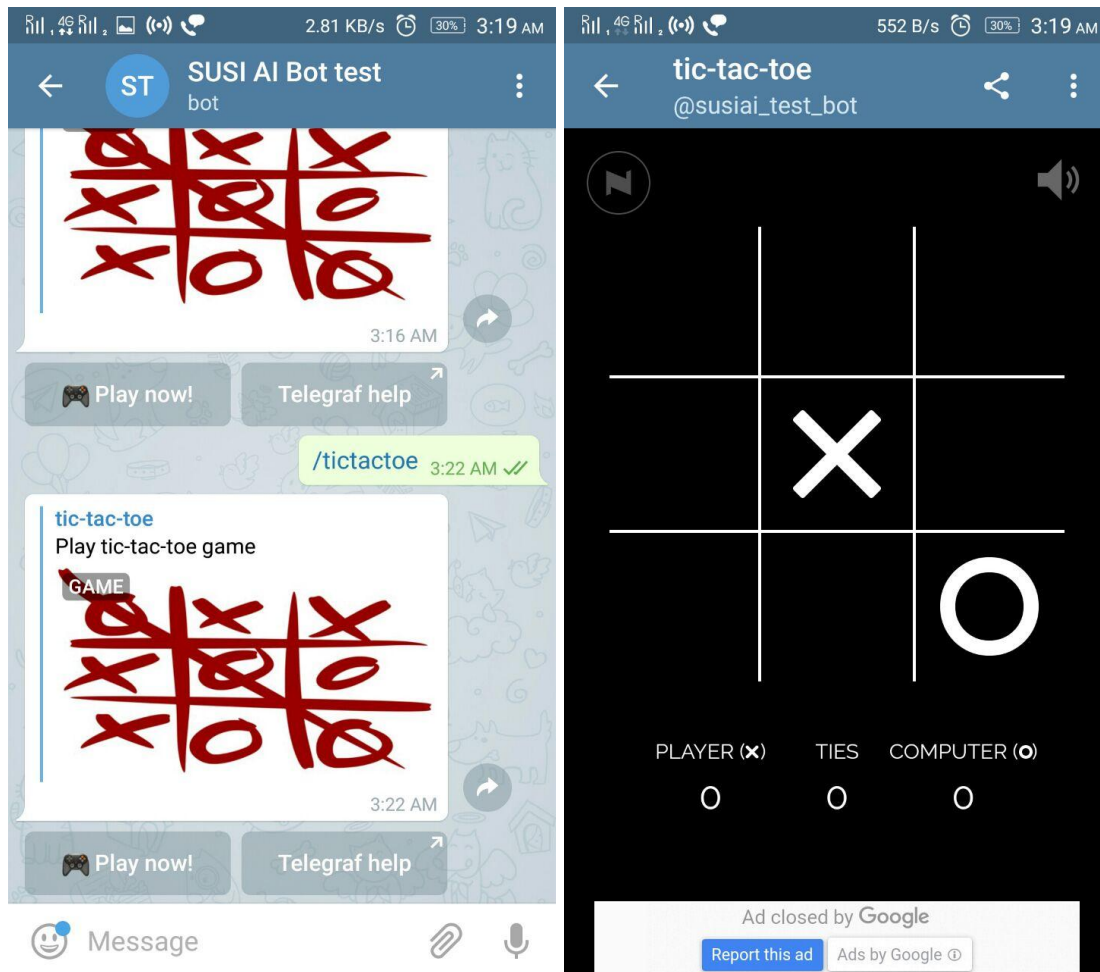
Create Tic-tac-toe game on SUSI Telegrambot

```
const Telegraf = require('telegraf')
const Extra = require('telegraf/extra')
const Markup = require('telegraf/markup')
const bot = new Telegraf(process.env.token)
const gameShortName = 'tictactoe'
const gameUrl = 'https://playtictactoe.org/'

const markup = Extra.markup(
  Markup.inlineKeyboard([
    Markup.gameButton('🎮 Play now!'),
    Markup.urlButton('Telegraf help', 'http://telegraf.js.org')
  ])
)

bot.command('/tictactoe', ({ replyWithGame }) =>
  replyWithGame(gameShortName, markup))
bot.gameQuery(({ answerGameQuery }) => answerGameQuery(gameUrl))
bot.startPolling()
```

Result:



Similarly we can create other games like quizzes, connect four, in the SUSI Telegram bot.

Create reminder feature

```
const Telegraf = require('telegraf');
var schedule = require('node-schedule');
const commandParts = require('telegraf-command-parts');
bot.use(commandParts());
bot.command('remind', (ctx) => {
  var args = ctx.state.command.args;
  //add NLP Later
  var index1 = args.indexOf("to");
  var index2 = args.indexOf("after");
  var job = args.substring(index1+3, index2-1);
  var time = args.substring(index2+6);
  ctx.reply("Will remind you to "+job+" after "+time);
});
```

```

var curDate = new Date();
var mins = time.substr(0,2);
var newDateObj = moment(curDate).add(parseInt(mins), 'm').toDate();
var j = schedule.scheduleJob(newDateObj, function(){
    ctx.reply("Reminder! You have to "+job);
});
})
bot.startPolling()

```

Result:

The screenshot shows a Slack chat interface. On the left, a user named Harshit sends a message: `/remind me to purchase pen after 5 minutes` at 4:57:43 PM. A bot named 'SUSI AI Bot test' (with a green circle icon containing 'ST') responds: 'Will remind you to purchase pen after 5 minutes' at 4:57:41 PM. At 5:02:41 PM, the bot sends a reminder: 'Reminder! You have to purchase pen'. Below this is a message input field with a 'Write a message...' placeholder, icons for attachments, emojis, and a 'SEND' button. On the right, a zoomed-in view of the bot's message shows the text 'SUSI AI Bot test' and 'Reminder! You have to purchase pen' with a 'REPLY' button and a 'VIT Go' logo at the bottom.

Thus a basic reminder functionality has been implemented. We need to add Natural Language Processing for getting the reminder context and formatting the reminder with options buttons like “Remind after 10 minutes”, “Okay”, etc.

Timeline

Community Bonding (April 23 - May 14)		Discuss implementation ideas with mentors. Go through the API documentation of Slack, Gitter, Zulip, Telegram, and examine other SUSI bots as well. Plan the best practices to build the bots. Solve and review issues in chat.susi.ai and other projects.
Week	Date	Tasks
Week 1	May 14 - May 20	<ol style="list-style-type: none"> 1. Solve existing issues in susi_slackbot. 2. Use the correct api and setup developing environment using ngrok. 3. Fix current documentation and installation instructions. 4. Detect SUSI’s mention in the messages and reply only to those. Extract the correct query. Handle error cases. 5. Write a blog on how to setup susi slack bot using ngrok.

Week 2	May 21 - May 27	<ol style="list-style-type: none"> 1. Fix table response. Format message using slack's markdown language. 2. Add support for hyperlink and map type of responses. 3. Setup mongodb database for storing users data. 4. Add todo list feature. Add 'create poll' feature. 5. Detect type of messages and send appropriate response for unsupported type of messages.
Week 3	May 28 - June 3	<ol style="list-style-type: none"> 1. Add task assigning feature. Save data in mongodb database. 2. Add introductory message, setup profile for susi slackbot. 3. Add games in the susi slack bot. 4. Go through the deployment instructions and best practices recommended by slack. 5. Deploy the bot in Slack App directory.
Week 4	June 4 - June 10	<ol style="list-style-type: none"> 1. Go through gitter development api. Solve existing issues in susi_gitterbot. 2. Setup local development environment using ngrok and write a technical blog about it. 3. Support multiple action types. Check for susi mention in messages, and extract the correct query. 4. Fix documentation and installation instructions.
June 11 - June 15		First Evaluation Period
Week 5	June 11 - June 17	<ol style="list-style-type: none"> 1. Setup mongodb database for storing users data. 2. Add todo list feature. 3. Support anchor, map, and table type of responses. Use github markdown specially to create the table. 4. Add introductory, help, and default messages.
Week 6	June 18 - June 24	<ol style="list-style-type: none"> 1. Setup susi gitterbot profile and deploy in a gitter channel for testing and demonstrating. 2. Make a user friendly website to guide users to easily deploy susi bot in their gitter channels. 3. Study more about Zulip bot development process and go through its api. Examine other Zulip bots and outline best practices.
Week 7	June 25 - July 1	<ol style="list-style-type: none"> 1. Solve existing issues in zulip bot. 2. Support anchor, map, image, and table type of responses. 3. Format messages, especially tables response using Zulip's markdown language. 4. Add feature for summarizing previous messages.
Week 8	July 2 - July 8	<ol style="list-style-type: none"> 1. Add todo list feature. Use zulip default storage. 2. Provide subscribed channel list to the user on asking susi. 3. Users can subscribe to specific streams by telling susi. Use

		APIs provided by Zulip. 4. Provide unsubscribe from streams feature.
July 9 - July 13		Second Evaluation Period
Week 9	July 9 - July 15	1. Deploy in production using Zulip bot server. 2. Write test cases for the bot, and follow other best practices recommended by zulip. 3. Deploy in bot testing stream of official zulip stream and wait for testing and approval.
Week 10	July 16 - July 22	1. Study about the Telegram bot development process, Telegraf docs, and lay out best practices. Solve existing issues. 2. Develop inline mode commands. 3. Handle location request feature.
Week 11	July 23 - July 29	1. Make games like tic-tac-toe and quizzes. Also enable inline playing mode. 2. Set up mongodb database for storing user's data. 3. Make todo list feature and save in database.
Week 12	July 30 - Aug 5	1. Add reminder feature in SUSI Telegram bot. 2. Make custom keyboards and add default commands. 3. Add default responses for global commands and customise SUSI's profile. Deploy SUSI Telegram bot. Fix installation documentation.
Aug 6 - Aug 14		Final Evaluation Period

About Me

I am currently a third year student at **VIT University** (Vellore, India), where I am pursuing my Bachelor's Degree in Computer Science and Engineering. I am a full stack web developer. I have good amount of experience using technologies such as **ReactJS**, **Angular 2**, NodeJS, Jekyll, PHP, and MySQL. I love learning new technologies, and keep myself updated with latest trends in technology. I am an active part of [Association for Computing Machinery \(ACM-VIT\)](#) and [Google Developers Group \(GDG-VIT\)](#), where we build various open source projects to help the community. I am contributing to FOSSASIA for the last 6 months. I was selected as a **Google Code-In Mentor** at FOSSASIA. I actively participated in the codeheat contest, where I contributed to various FOSSASIA projects. I got to learn the various best practices followed in FOSSASIA. I participated in various project related discussions with co-developers and mentors. I mostly keep myself active on the gitter channels, helping newcomers to begin to contribute.

Past Contributions to FOSSASIA

fossasia/chat.susi.ai	<p>Pull Requests:</p> <ul style="list-style-type: none">• encode users query - fix #1104• fix #1100 - improve Text-to-Speech Dialog• Improve restoring default theme in theme changer dialog - Fix #887• Fix #879 - set previous theme on exit• Fix #818 - Copy previous messages on pressing 'up' and 'down' arrow keys• Restructure Settings, enable dark theme - Fix #840• Fixes issue #819 - Improve save button actions in settings• Fixes issue #812 - Enhance search function• Fixes issue #809 - Enable click on the toggle label in Login popup <p>Issues Opened:</p> <p>https://github.com/fossasia/chat.susi.ai/issues?utf8=%E2%9C%93&q=is%3Aissue+author%3Ahkedia321</p> <p>PRs and Issues reviewed / commented:</p> <p>https://github.com/fossasia/chat.susi.ai/issues?utf8=%E2%9C%93&q=hkedia321</p>
fossasia/susi_firefox_bot	<p>Pull Requests:</p> <ul style="list-style-type: none">• Fix #107 - hide scroll down button by default• Fix #103 - limit the height of table• Fix #102 - show icon to jump to bottom of chat• Fix #83 - Improve UI in Settings page• Fix #84 - Add Change Password option• Add context menu on selection and right click - fix #67• Show logged in email and logout option - fix #71• Move documentation image - Fix partially #1• Support map and anchor type of responses - fixes #27• Fix issue #6 - support different type of response from server• Fix issue #4 - Show loading icon <p>Issues Opened:</p> <p>https://github.com/fossasia/susi_firefoxbot/issues?utf8=%E2%9C%93&q=is%3Aissue+author%3Ahkedia321</p>

	<p>PRs and Issues reviewed / commented:</p> <p>https://github.com/fossasia/susi_firefoxbot/issues?utf8=%E2%9C%93&q=hkedia321</p>
fossasia/susi_chromebot	<p>Pull Requests:</p> <ul style="list-style-type: none"> • Fix #167 - apply dark theme to loading message • fix #144 - Add context menu on text selection and right click <p>Issues Opened:</p> <p>https://github.com/fossasia/susi_chromebot/issues?utf8=%E2%9C%93&q=is%3Aissue+author%3Ahkedia321</p>
hkedia321/scrum_helper_extension	<p>This is a chrome extension which helps FOSSASIA contributors to write scrums in google groups. It prefills the scrum with the user's PRs and Issues, fetching data from Github API. This extension is used by many active FOSSASIA developers to write scrums.</p> <p>https://github.com/hkedia321/scrum_helper_extension</p>

Other Open-Source Projects

hkedia321/riviera-2018-website	The official website of Riviera 2018, which is the biggest annual cultural fest of VIT University.
hkedia321/gdd-scheduler	This is a web application which helps you plan your sessions at Google Developers Day, India, 2017. You can select the sessions you want to attend and then save a pdf of it.
hkedia321/rice-quality-analysis	Analyse quality of rice grains by using Image processing techniques.
hkedia321/code2create_webapp	Web application in NodeJS for Code2Create - a hackathon organised by ACMVIT.
GDGVIT/myFFCSreLoaded	MyFFCS helps VIT students plan their semester courses. Users can visualize their semester's timetable before creating it.
hkedia321/VITCabShare	VIT Cab Share is a crowd-sourced website which helps VIT students to find co-travellers to share their cab with,

	while travelling from Vellore to airports.
--	--

My Commitments

Do you have any other commitments during the GSoC period, 23rd April to 14th August?

I have my exams till 26th April. I will prepare for it beforehand so that I can give full time for GSoC. Other than that, I don't have any other commitments. My university's summer holidays are there for the rest of the period, so I can fully concentrate on GSoC. I plan to work full time and can easily dedicate 40 - 50 hrs a week. Apart from the mentioned tasks, I plan to be active on other FOSSASIA projects as well, implement new features, solve issues and review PRs.

Do you have plans to have any other job or internship during this period?

No, I will not be doing any other internship or job during this period. I will be fully focused on GSoC only.

My Motivation

Why am I interested in working for the software project that I have selected?

We don't want to live in a world where we have only limited options by giants such as Google with their apps like Google Assistant. We want to have more options. That's what I am excited about working in FOSSASIA and developing open source softwares which can be useful for other people.

I have been working in FOSSASIA for the last 6 months. I have created and contributed to many SUSI bots. I fully understand the requirements and the scope of this project. Working in FOSSASIA will give me an opportunity to experience the development process, the best practices followed in open source development, and interact with the best people.

Would this application contribute to my career? If so, how?

Yes, this project will definitely be a great kickstart to my career. Working in FOSSASIA will give me immense experience and insights into software development. I wish to continue working in FOSSASIA after GSoC and participate in future coding programs.

How do I intend to continue being an active member of this project after the summer is over?

After the successful completion of this project, I will continue working in FOSSASIA and continue development on projects. I will continue as an active developer and participate in future

coding programs in FOSSASIA. I intend to reach out to more developers and bring them on board. I would love to attend the FOSSASIA Summit.

Do you understand this is a serious commitment, equivalent to a full-time paid summer internship or summer job?

Yes, I fully understand that this is a serious commitment. I will dedicatedly work on my project and fulfill my responsibilities. I don't have any other major commitment during this period. I will be active in mailing lists and gitter channels throughout the summer. I will actively participate in the project development process and discussion with mentors.

I am very much enthusiastic about working in FOSSASIA this summer. I am willing to work on different project also if I get assigned by mentors. I can assure you that if I get selected for GSoC, I will give my full efforts to successfully complete this project and continue working in FOSSASIA.

Looking forward to working with you,

Regards,
Harshit Kedia