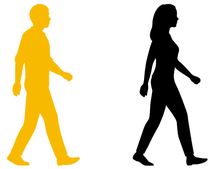
[redacted]

[redacted]

[redacted] Lauren Poon [redacted]

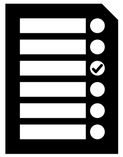
Agenda

Walkthrough of design thinking

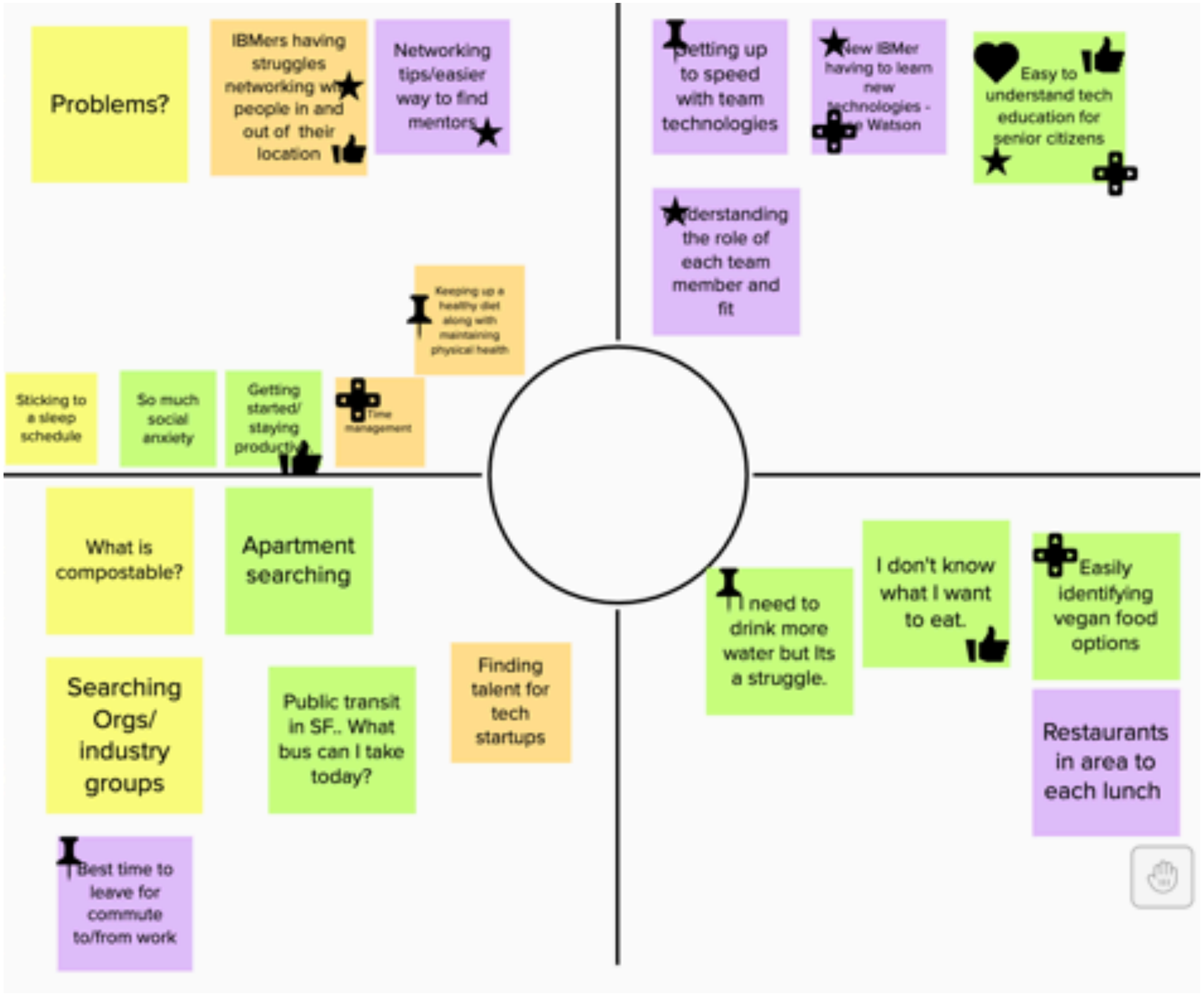


DEMO Solution

Project Timeline

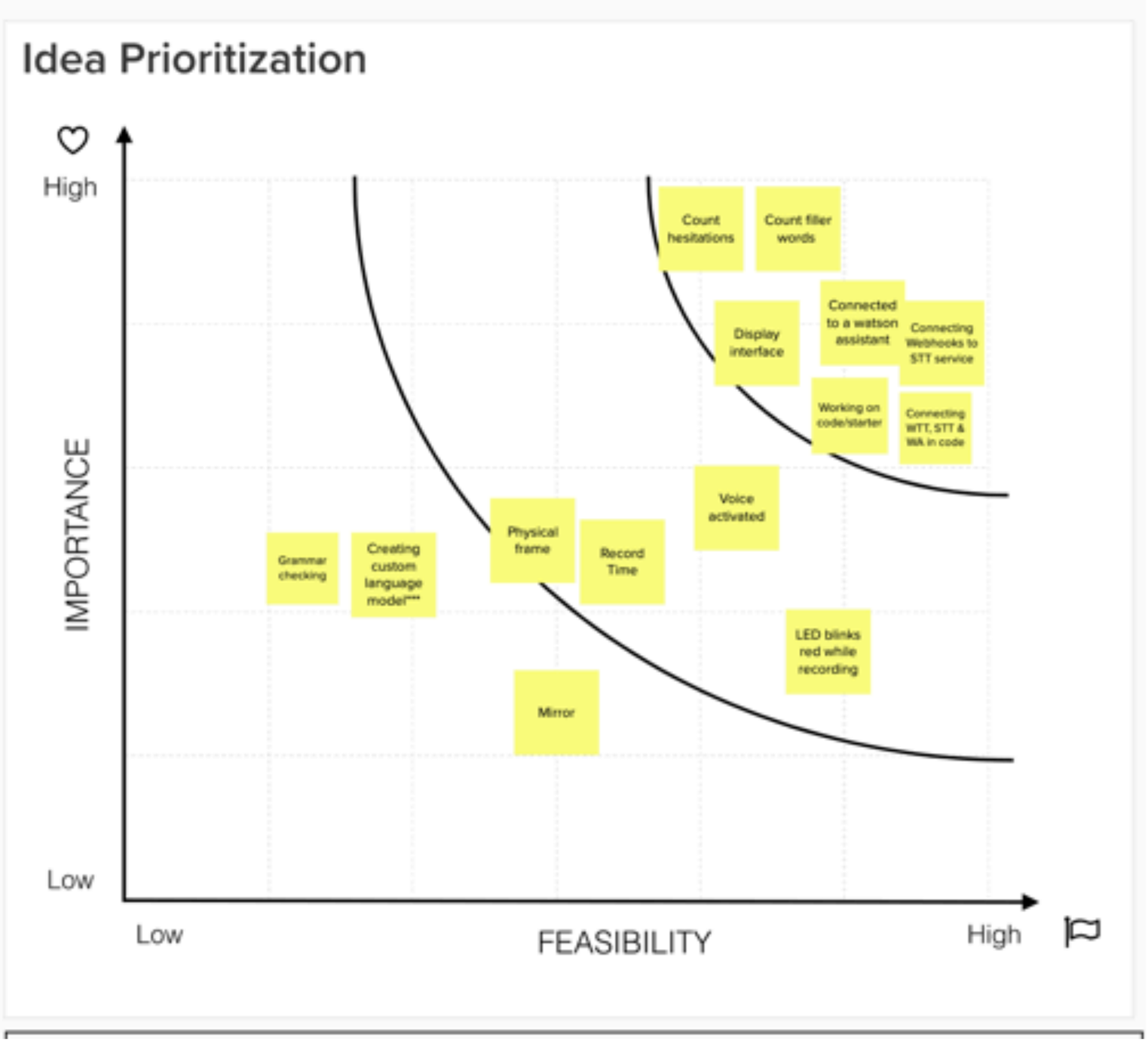


Next steps…



Design Thinking Session #1: Idea Board

Design Thinking Session #2: Final Decision

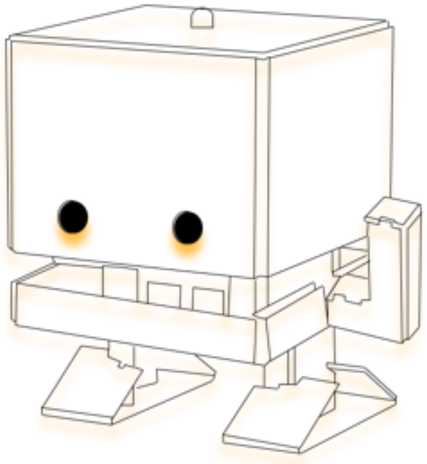


Design Thinking Session #3: Idea Prioritization Board

TOASTER BOT

Toast Masters Simulation Bot

PUBLIC SPEAKING



TJ bot will simulate Toast Masters that can run anywhere, so that anyone could strive to perfect their presentation skills.

Hesitation Recipe

```
30
31  // Setting up variables
32  let recording = false;
33  let name = tj.configuration.robot.name.toLowerCase()
34  let start = name + " start recording"
35  let stop = name + " stop recording"
36  let stopped = name + " stopped recording"
37  let transcript = 'transcript.txt'
38
```

```
38
39  // creates transcript file and replaces the existing one
40  fs.open(transcript, 'w+', function(err) {
41      if (err) {
42          throw "could not open file: " + err
43      }
44  });
45
```

```
46  // Main listening callback function
47  tj.listen(function(input) {
48      input = input.toLowerCase()
49      if (recording) {
50          stop_record = input.includes(stop) || input.includes(stopped) //tj often hears stopped so we ad
51          if (stop_record) {
52              console.log("stopping recording")
53              recording = false;
54              let hesitationCounter = 0
55              var words = fs.readFileSync(transcript);    //reading words in transcript file
56              words = words.toString();
57              words = words.split(" ");
58              for (var i = 0; i < words.length; i++) {
59                  if (words[i] == '%hesitation'){
60                      hesitationCounter++;            //increment counter when um/hesitation is heard
61                  }
62              }
63              console.log("Number of 'ums': " + hesitationCounter)
64          } else {
65              fs.appendFile(transcript, input, function (err) {
66                  if (err) throw err;
67                  console.log("wrote to transcript: " + input)
68              });
69          }
70      } else {
71          start_record = input.includes(start)
72          stop_record = input.includes(stop)
73          console.log(input)
74          console.log("start: " + start_record)
75          console.log("stop: " + stop_record)
76          if (start_record) {
77              console.log("starting recording")
78              recording = true;
79          } else {
80              console.log("Try saying, \"" + tj.configuration.robot.name + ", start recording\"")
81          }
82      }
83  })
```

```
Lauren.Poon [redacted]
~ — pi@Karl: ~ — -zsh        ~ — -zsh

pi@Karl:~/Desktop/botty-builders/hesitation $ sudo node hesitation.js
verbose: TJBot initializing microphone
verbose: TJBot initializing LED
verbose: TJBot initializing assistant service
verbose: TJBot initializing speech_to_text service
verbose: TJBot initializing text_to_speech service
verbose: TJBot initializing visual_recognition service
info: Hello from TJBot! My name is Watson.
verbose: TJBot library version v1.5.1
Hello. I'm the ToastMasters Bot. If you'd like to practice presenting, I can give you some feedback.
Try saying, "Watson, start recording" or "Watson, stop recording"
verbose: TJBot initializing microphone
info: TJBot heard: Watson start recording
watson start recording
start: true
stop: false
starting recording
info: TJBot heard: so now I'm gonna show you how I add a card to my to do list so I'm gonna go right
 down to the bottom of the list and press be adding another card but in we'll call this the X. Y. Z.
 loops yeah X. Y. Z. card
wrote to transcript: so now i'm gonna show you how i add a card to my to do list so i'm gonna go rig
ht down to the bottom of the list and press be adding another card but in we'll call this the x. y.
z. loops yeah x. y. z. card
```

Results

```
wrote to transcript: %hesitation i can not a label so label
s you have from each
info: TJBot heard: Watson stop recording
stopping recording
Number of 'ums': 3
```

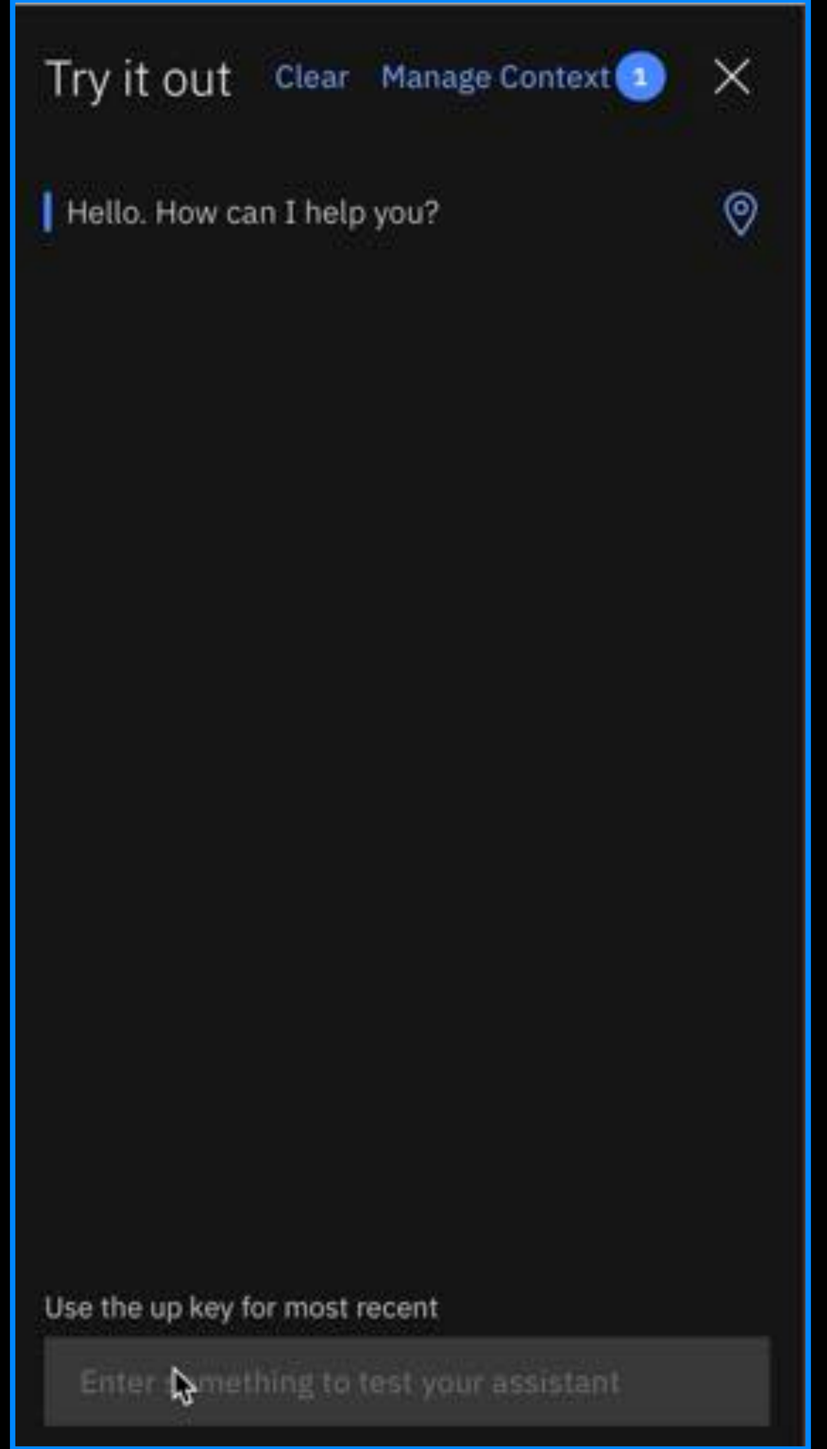
```
pi@Karl:~/Desktop/botty-builders/hesitation $ cat transcript.txt
so now i'm gonna show you how i add a card to my to do list so i'm gonna go right down to the bottom of t
he list and press be adding another card but in we'll call this the x. y. z. loops yeah x. y. z. card we'
re gonna click it to edit it just like this and we're gonna add a description like you need to do is %hes
itation you can definitely add a much lengthier description this helps me keep track of things that i nee
d to do sometimes i'll just copy and paste emails or slack messages into the description so i remember ex
actly where to reference i also like comment notes as i'm doing this card or if i have links that i need
to do it i usually comments in in here it saves the time and date four when you add it %hesitation i can
also add members who card this is my personal board so i can just ask myself or not i know that it's my j
ob %hesitation i can not a label so labels are really cool to visually see how many tasks you have from e
```

Running the Hesitation Recipe

```
start: true
stop: false
starting recording
info: TJBot heard: so now I'm gonna show you how I add a card to my to do list so I'm gonna go right do
wn to the bottom of the list and press be adding another card but in we'll call this the X. Y. Z. loops
 yeah X. Y. Z. card
wrote to transcript: so now i'm gonna show you how i add a card to my to do list so i'm gonna go right
down to the bottom of the list and press be adding another card but in we'll call this the x. y. z. loo
ps yeah x. y. z. card
info: TJBot heard: we're gonna click it to edit it just like this and we're gonna add a description lik
e you need to do is
wrote to transcript: we're gonna click it to edit it just like this and we're gonna add a description l
ike you need to do is
info: TJBot heard: %HESITATION you can definitely add a much lengthier description this helps me keep t
rack of things that I need to do sometimes I'll just copy and paste emails or slack messages into the d
escription so I remember exactly where to reference I also like comment notes as I'm doing this card or
 if I have links that I need to do it I usually comments in in here it saves the time and date
wrote to transcript: %hesitation you can definitely add a much lengthier description this helps me keep
 track of things that i need to do sometimes i'll just copy and paste emails or slack messages into the
 description so i remember exactly where to reference i also like comment notes as i'm doing this card
or if i have links that i need to do it i usually comments in in here it saves the time and date
info: TJBot heard: four when you add it
wrote to transcript: four when you add it
info: TJBot heard: %HESITATION I can also add members who card this is my personal board so I can just
ask myself or not I know that it's my job
wrote to transcript: %hesitation i can also add members who card this is my personal board so i can jus
t ask myself or not i know that it's my job
info: TJBot heard: %HESITATION I can not a label so labels are really cool to visually see how many tas
ks you have from each
wrote to transcript: %hesitation i can not a label so labels are really cool to visually see how many t
asks you have from each
info: TJBot heard: Watson stop recording
stopping recording
Number of 'ums': 3
```

Chatbot

- Randomize a topic
- Re-iterate that randomization if the user refuses the topic
- Repeat the sequence until approval
- Start Listening to user



API Connections

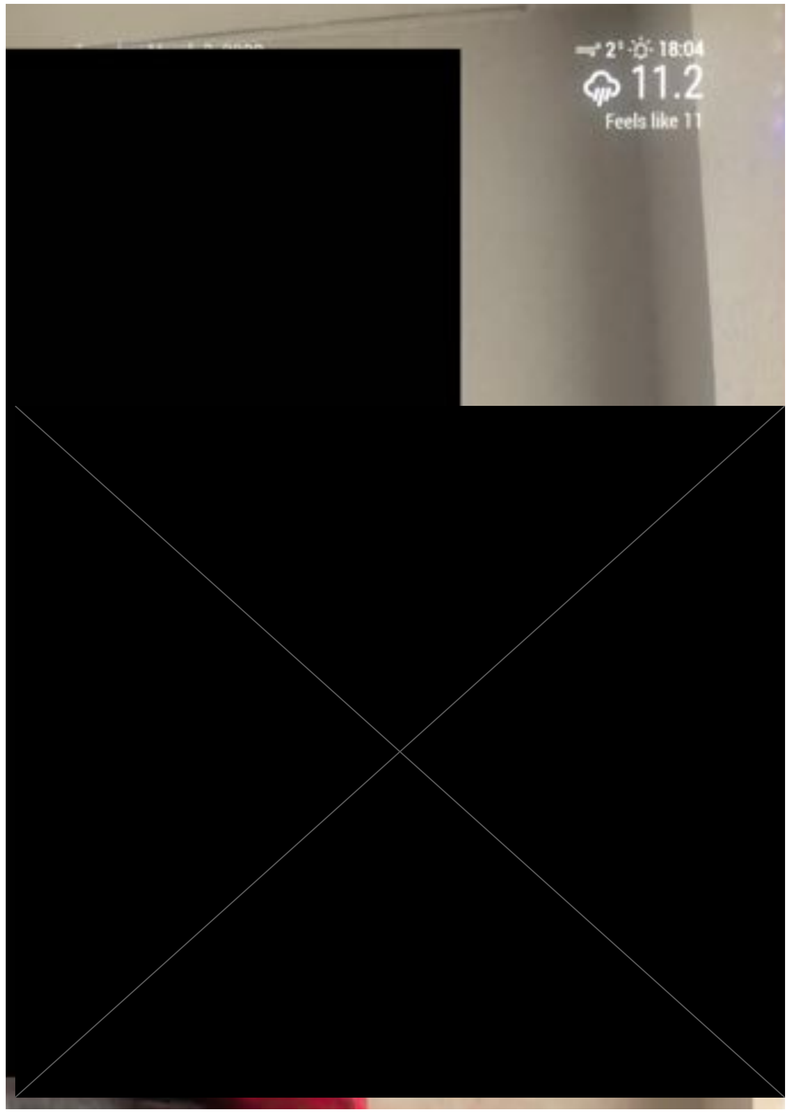
- Connections for routes in app
- 15 Minute Intervals

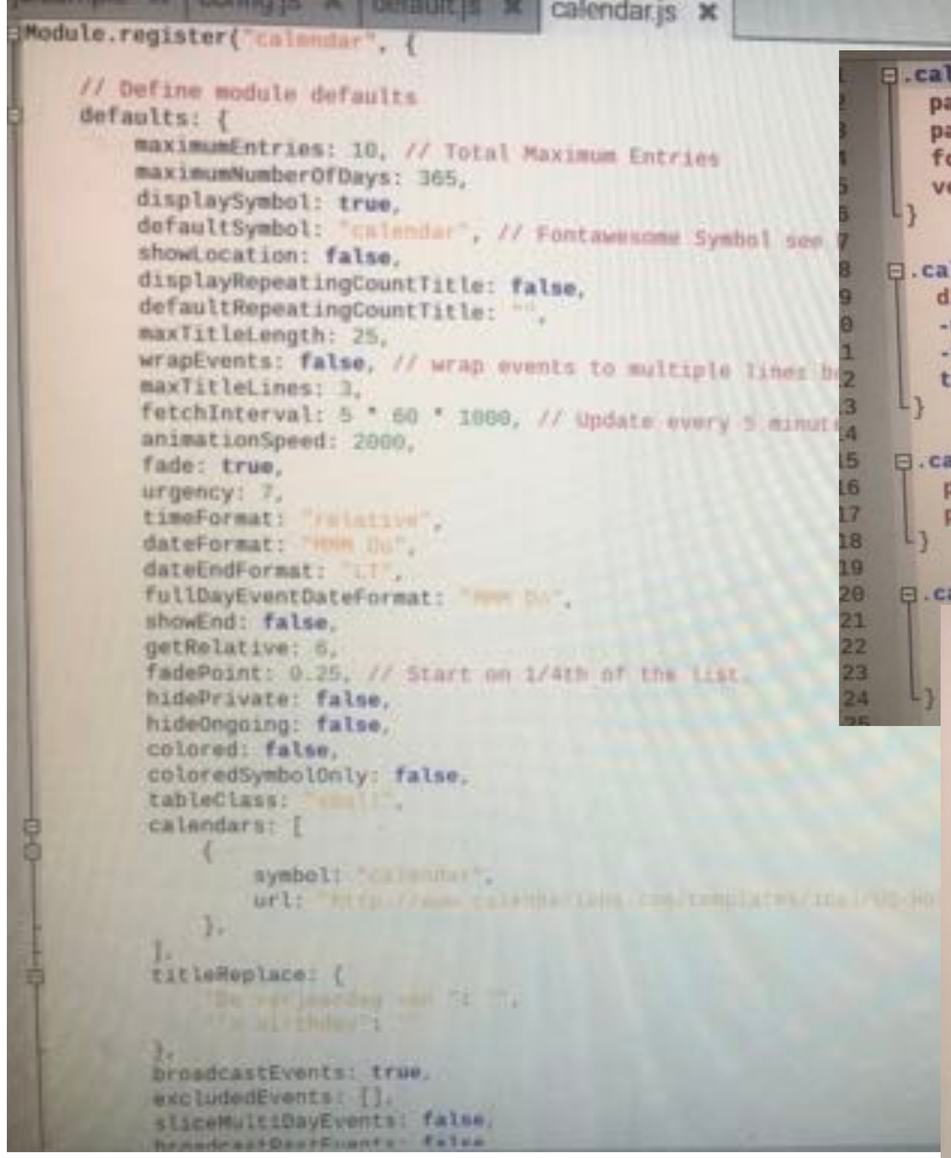
app.js > ...

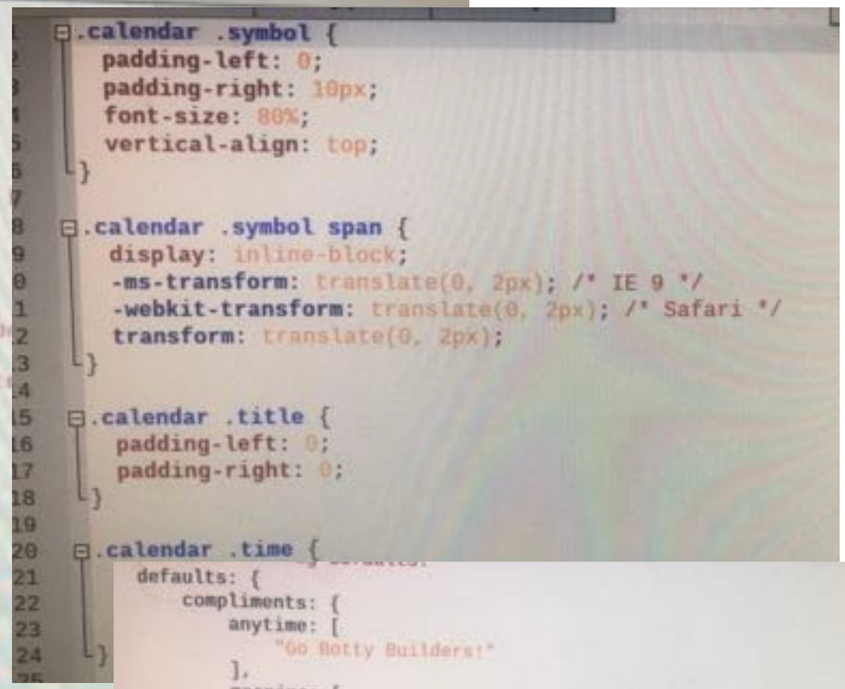
```
var now = moment(); // new Date().getTime();
var then = moment().add(900, 'seconds'); // new Date(now + 900 * 1000);

$(".now").text(moment(now).format('h:mm:ss a'));
$(".then").text(moment(then).format('h:mm:ss a'));
$(".duration").text(moment(now).to(then));
(function timerLoop() {
  $(".difference > span").text(moment().to(then));
  $(".countdown").text(countdown(then).toString());
  requestAnimationFrame(timerLoop);
})();
```

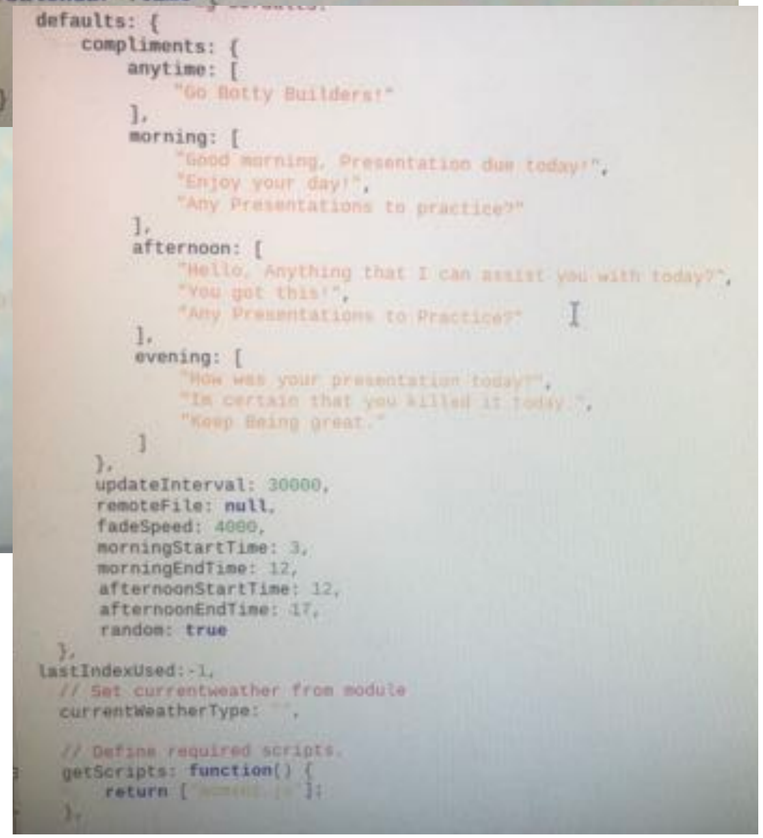
```
@app.route('/api/toast_masters', methods=['POST', 'GET'])
def getConvResponse():

    convText = request.form.get('convText')
    convContext = request.form.get('context', "{}")
    jsonContext = json.loads(convContext)

    response = assistant.message(workspace_id=workspace_id,
                                 input={'text': convText},
                                 context=jsonContext)

    response = response.get_result()
    reponseText = response["output"]["text"]
    responseDetails = {'responseText': '... '.join(reponseText),
                       'context': response["context"]}
    return jsonify(results=responseDetails)


@app.route('/api/approval', methods=['POST'])
def getSpeechFromText():
    inputText = request.form.get('text')
    ttsService = TextToSpeechV1()

    def generate():
        if inputText:
            audioOut = ttsService.synthesize(
                inputText,
                accept='audio/wav',
                voice='en-US_AllisonVoice').get_result()

            data = audioOut.content
        else:
            print("Empty response")
            data = "I have no response to that."

        yield data

    return Response(response=generate(), mimetype="audio/x-wav")


@app.route('/api/disapproval', methods=['POST'])
def getTextFromSpeech():

    sttService = SpeechToTextV1()

    response = sttService.recognize(
            audio=request.get_data(cache=False),
            content_type='audio/wav',
            timestamps=True,
            word_confidence=True,
            smart_formatting=True).get_result()
```

Smart Mirror





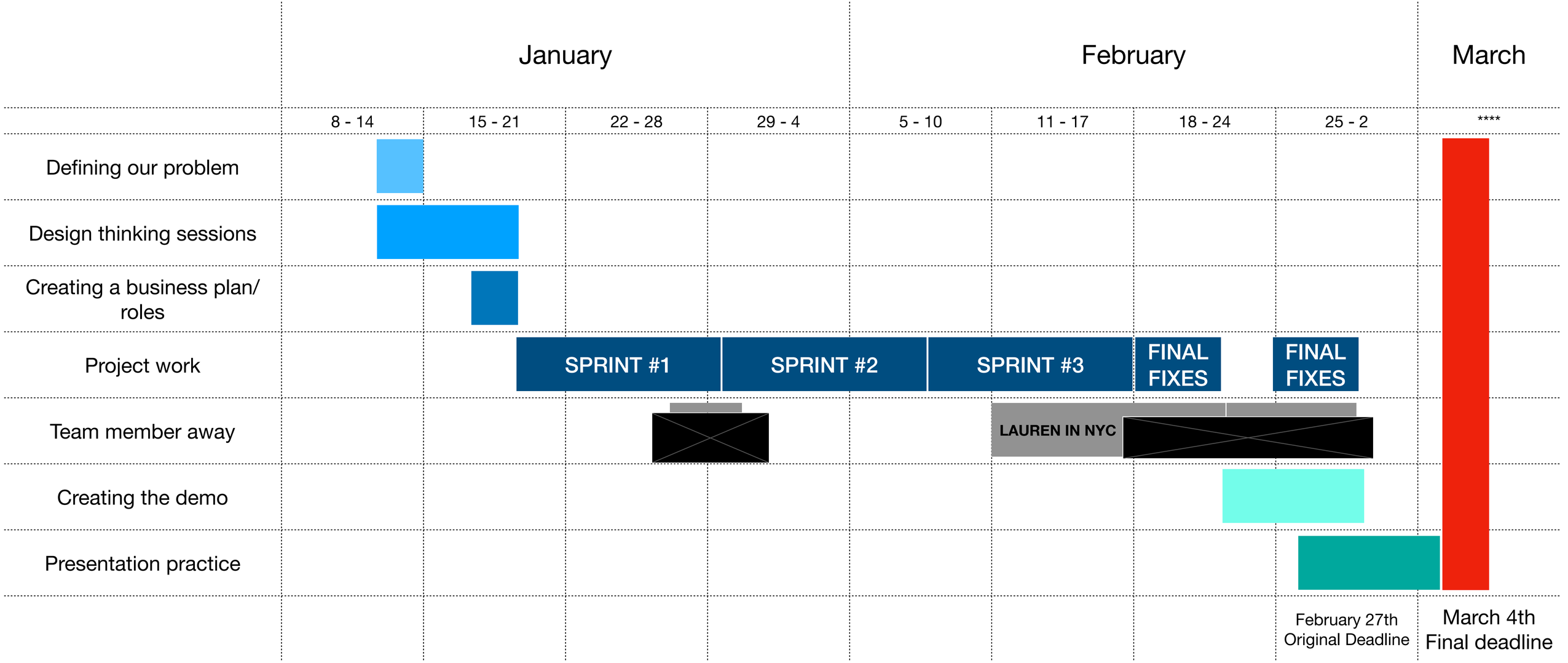




Materials = $150
- Monitor
- Raspberry pi
- HDMI>VGA
- 2-way mirror

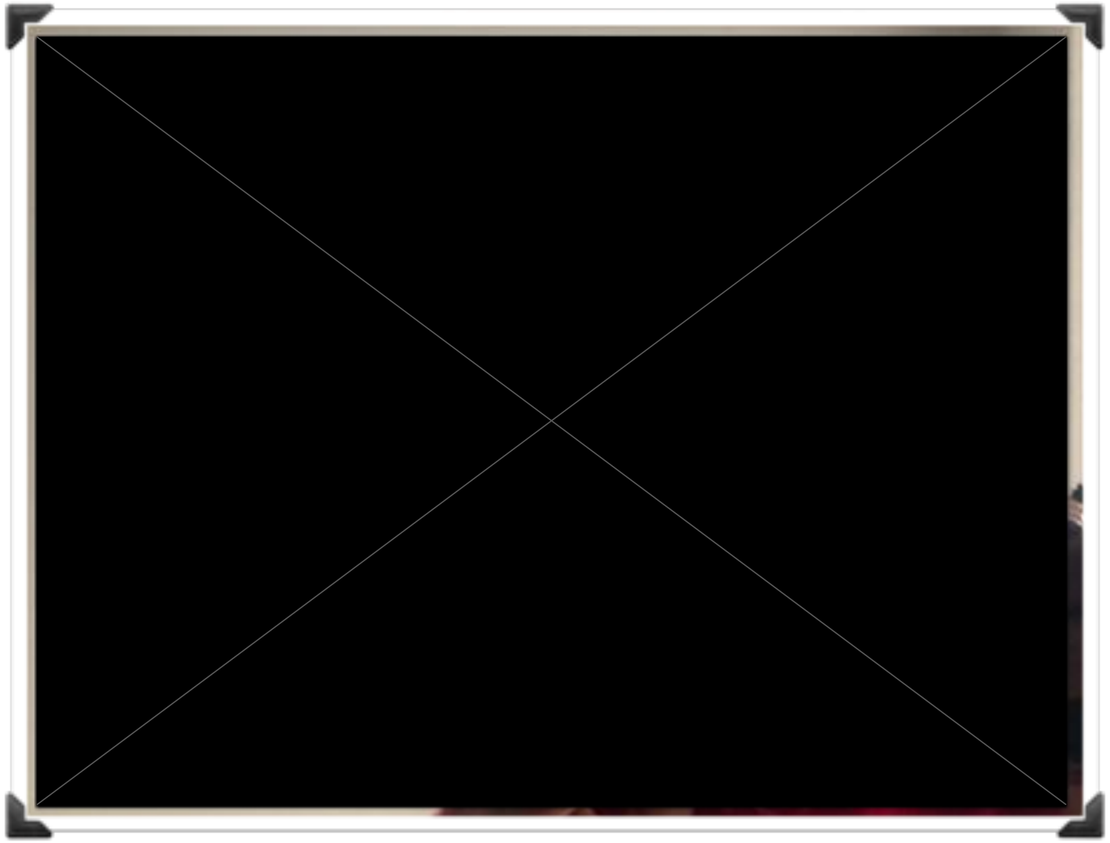
Features:
- Affirmation quote
- Daily weather
- Calendar events

Botty Builders Project Timeline



Future Features

- Customizable Widgets
- Timing Capabilities
- User Personalization
- Alexa/Google/Siri Assistant
- Fully Integrated Mirror



THEME	CRITERIA	DESCRIPTION	SCORE
Idea	Innovation	How creative or innovative is the idea behind the solution?	10
	Value	How relevant is the solution to a real market or business problem?	10
Solution	Application of new skills	How well did the team incorporate or apply technical kills learned during their training?	10
	Feasibility	Does it actually work? Could this scale as a real solution with multiple users?	10
Delivery	Teaming	Did every team member have a roll?	10
	Quality	How was the overall presentation and delivery ?	10
Overall Showcase			60