Appendix

Works Cited

"Sheets API." *Google Developers*, https://developers.google.com/sheets/api.
Accessed 20 Mar. 2021.

Twilio Python Quickstarts for SMS, Voice and More. https://www.twilio.com/docs/quickstart/python?utm_source=docs&utm_medium=social&utm_campaign=guides_tags. Accessed 20 Mar. 2021.

How to Connect Form Data to Google Sheets. www.youtube.com, https://www.youtube.com/watch?v=qA_QzQrmTzQ. Accessed 20 Mar. 2021.

HTML Forms. https://www.w3schools.com/html/html_forms.asp. Accessed 20 Mar. 2021.

Datetime — Basic Date and Time Types — Python 3.9.2 Documentation. https://docs.python.org/3/library/datetime.html. Accessed 20 Mar. 2021.

Interview

- Key

M: me (Svadrut Kukunooru)

C: client (Sreedhar Kukunooru)

- Initial Conversation

This transcript is mostly recorded from memory, transcribed a few hours after the actual conversation occurred. Please excuse any errors.

C: What's up? Why do you look so worried?

M: I just have an assignment due for Computer Science. It's a big project, and I don't really know where to start.

C: What's the assignment about?

M: Basically, we have to find a client and create a software solution for him -- wait. You're the president of WATG, right? Do you guys have anything that I could help with by creating a program? What about that big talent show -- what was it called?

C: It's called Dhoom Dhaam. And yeah, we could use some help with that.

M: What do you need?

C: (*laughing*) I can't come up with stuff we need help with on the spot right now. Could you give me a day or two?

M: Yeah, sure. Let me know when you do.

- Second Conversation

M: Do you have an idea for what I can do?

C: I do, in fact. For our talent show, we have around 40 people. What we struggle with is our website design for registering people and reminding people that their performance is soon. If that's enough for you, then you can just take care of those two tasks. Please make a web form with the same capabilities as the Google Form, and that sends the form data to a Google Sheet for us.

M: What do you guys do with the data in the Google Sheet?

C: Well, we first see if we have enough people in each category, and if there are too many people in one category, we ask some of them to switch. Next, once we have the desired number of people, we can decide times for each person and put them in a column next to the people.

M: How do you remind people about their performance?

C: Right now, we use around 15 pagers and manually page people a few minutes before their performance starts. This is always very time-consuming since we're also involved in organizing a bunch of other stuff, like food and the vendors and emceeing and--

M: Yeah, I understand. So you want me to create a program -- a script that does this for you?

C: Yeah, that would be nice.

M: Alright, I'll let you know about the requirements for the program soon, and then I can start working on it. Sound good?

C: Sounds good. I'll talk to you later.

- Third Conversation

M: Hey, I finished drafting the success criteria. Could you let me know if it's good or not?

copied text

- Allow for people to register for the talent show and record their team name, a summary of what they're performing, their email address, and their phone numbers in a form and click submit

- If the form data they entered is wrong, notify the users that they need to fill out the form again.

- Have the data from that form directly transfer to a spreadsheet to more easily facilitate manipulations of the data by the organizers.

  - This is already in place with a Google Form, but my client would like to have everything on one page and not link to external services. For more information on this topic, please refer to Appendix A.

- After the organizers determine times for each of the performers in the spreadsheet and put the times next to each performer, have an algorithm that calls the data in the spreadsheet and formats it into an appropriate dictionary with nested lists.

  - I don't want to use arrays since Python doesn't support that without me importing another library and I already am importing too many libraries. Even though performance doesn't matter, I would still like to keep the programs' memory use as low as possible. (I realize using a low-level language would help with that, but I do not have the time to learn C.)

- Have an algorithm that checks the current time and compares it with the times in the formatted dictionary to determine whether to start the notifier program.

- After checking them in on the day of the event, show the participants what times they are performing and provide them with two reminders to get ready for the performance.

  - In previous years, my client had to spend hundreds on pagers to each team and remind them manually when to prepare for the performance, leaving less time for organization of other parts of the event. For more information about our conversation, please refer to Appendix A.

C: Yeah, this looks great! This is exactly what I wanted!

M: Sweet! I'll let you know when I'm done so we can review the app!

- Final Conversation

M: Hey, I'm done with the app. Do you want to review it?

C: Sure! I'm in a meeting though, so give me an hour or so.

M: Sounds good. Let me know when you're ready to review.

one hour later

C: Are you able to show me your app now?

M: Yeah, sure!

shows him the web form and script and demoes its various features

M: So, what do you think? Have I met all the success requirements?

C: It looks like to me that you have! Sweet, thanks a lot for this! Administrating the talent show will be much easier thanks to your help.

M: Thanks for your vote of approval! Do you have any ideas on how the app could be improved?

C: You could a front-end for the texting application to show the people texted and the time left until the next step. Also, you could have a utility that pauses the program for a while and that instantly allows you to move the program to the next person in the queue. You could also maybe use the Google Sheets API instead of the hacky Ajax solution if you have more time.

M: Thanks, that's all I need! Talk to you soon!

C: Thanks, see you later!