How to make a reservation system for a small to medium facility for free

Roland Himmelhuber, Wyant College of Optical Science, The University of Arizona, 2024

If you are managing a facility with machines and instruments that your users want to reserve and you want to be able to create usage data, either for utilization analysis or billing. This guide is for you.

If you need to manage 100 or more machines or items (like lenses, microphones, cameras, ect) this will probably not that helpful to you.

There are many commercial lab management/reservation systems available, like iLabs or Clustermarket. There is also a slew of applications that are pure lab management with inventory systems and project management. None of those we found fitting for our lab with around 100 users and around 20 tools and instruments.

Requirements for the reservation system should be as follows:

1. Users must be able to create reservations, but not edit or delete them without permission. This is to ensure billing is done properly and people don’t just delete their reservation when they are done.

2. Users must be able to see all reservations.

3. All reservation data (user, time, tool, ect) must be saved for billing and statistics.

Before we dive into the details lets have a look at the general flow of how a reservation is made and what we will need to make it work. We will be using 4 tools for this.

1. Google Forms

2. Google Sheets

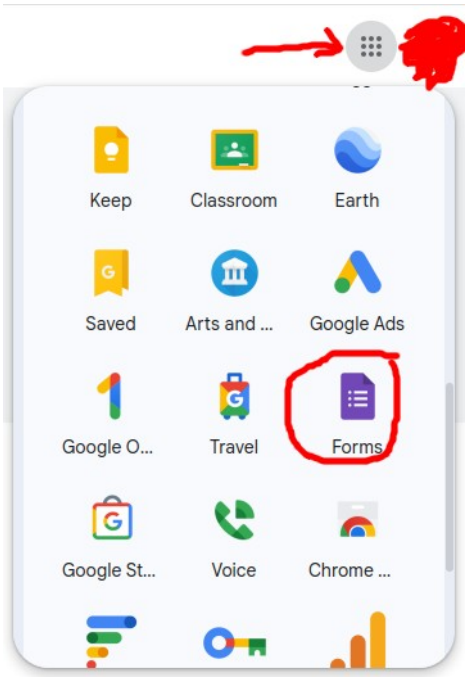
3. appscript

4. Google Calendar

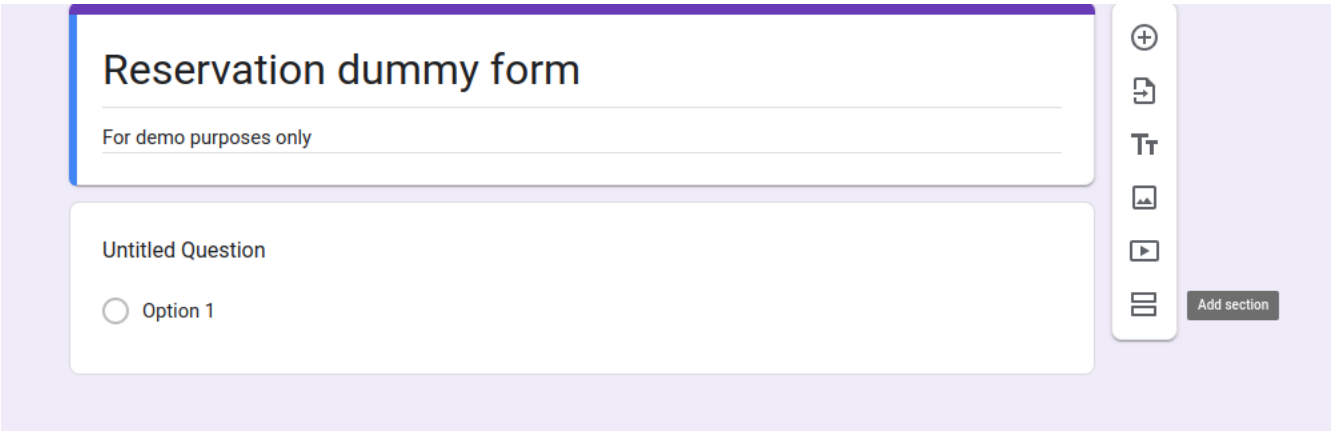
A user fills out a Google form with the reservation details (name, tool, time, ect). The data is written into a Google sheet. Then an appscript is started that takes takes the data and creates a Google calendar entry with the relevant information (who is using what when).

Section 1: The Form

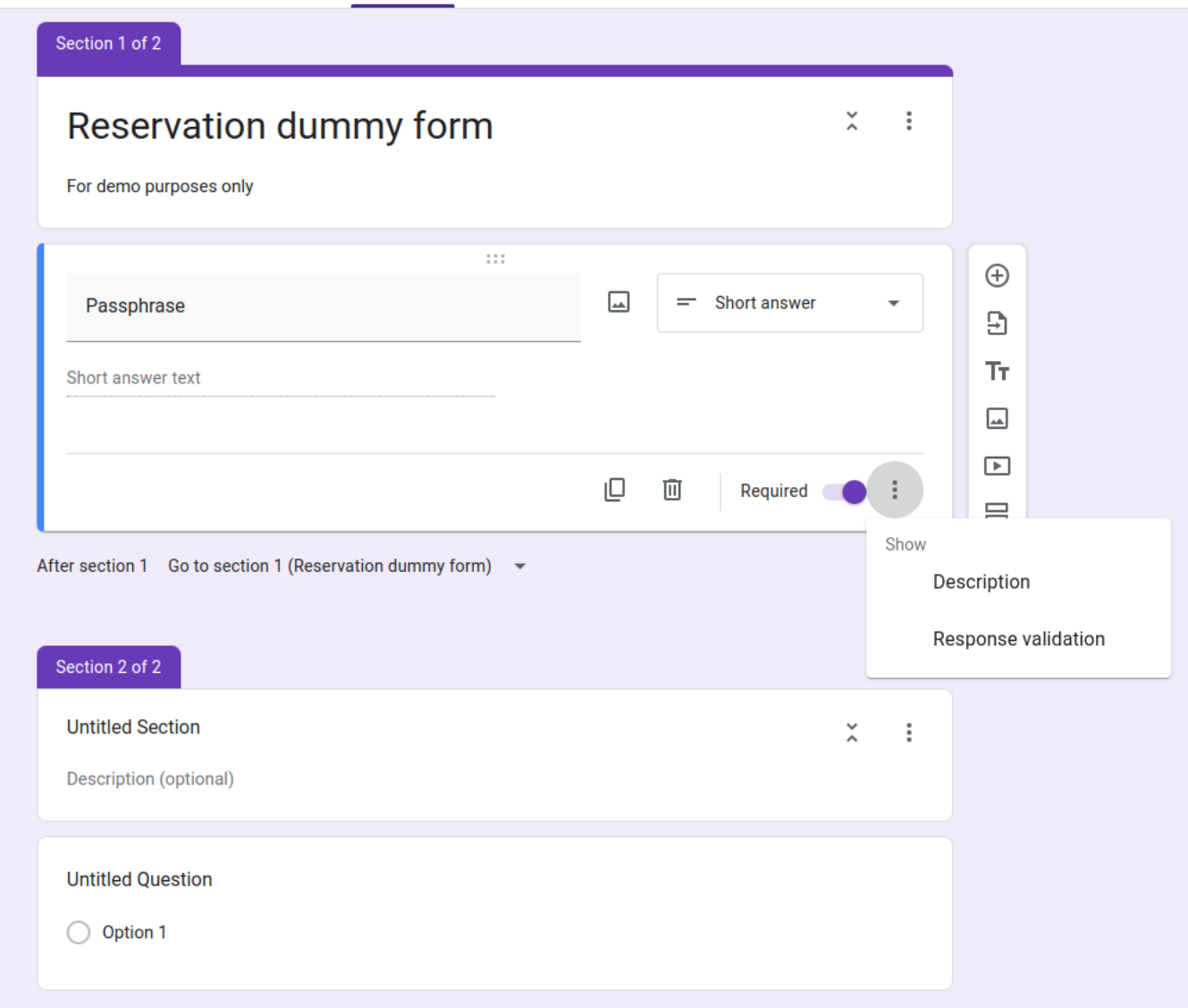
If you don't know what a Google form is, log into your Google account and click on the 3x3 dot array next to your icon:



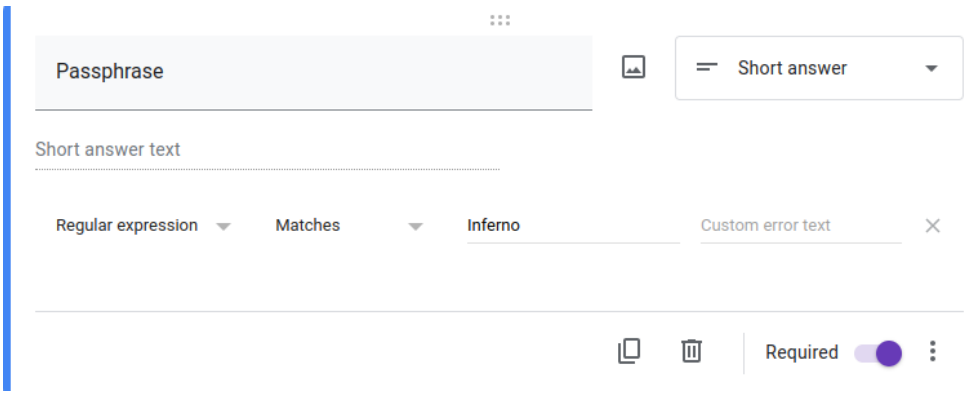
Click on new form and you can begin. I think it is a good idea to password protect the form. An easy way of doing this is to create a section:



In section 1 create the first question by clicking the plus sign. Make sure to make the question mandatory and validate the response:



The response should be regular expression the matches your password of choice, in this case “Inferno”.

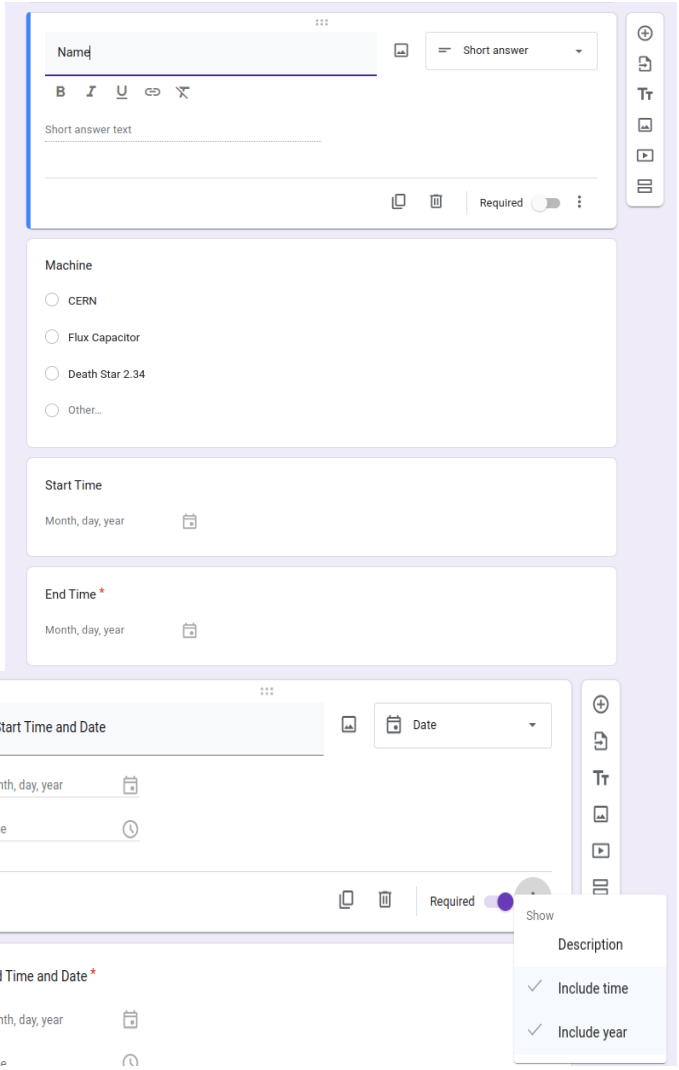


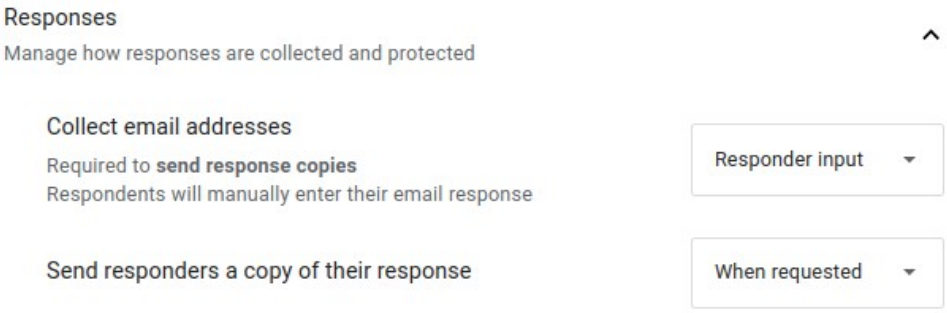
This way the person who is trying to fill out the form won't see the rest of it unless the correct password is entered. Make sure to select "go to section 2" from the drop down menu of section 1.

Then continue to section 2 and create all the questions you find necessary. Don't worry about an email address quite yet, we will deal with this later.

To include time and date in a question. You have to make it a date question first and then include time. At the very end, include a multiple choice question that only has one answer. The number 1 is a good choice. We will need that later.

Now that you have the form, we need to have the data written into a sheet. For that, on the top of the page click on responses and link to sheet. I also think it's a good idea to have emails with responses send to users if they want that. I am changing the sheet once a month, so it does not get too unwieldy.





Note that section 1 now asks for an email address. You can test the form by clicking "Send" and grabbing the link.

Section 2: The Sheet, the Script and the Calendar

If you fill out a few forms you will see your sheet getting populated like this:

	A	B	C	D	E	F	G	H
	Timestamp	Email Address	Passphrase	Name	Machine	Start Time and Date	End time and Date	Select the 1
	7/23/2024 13:02:32	muh@mah.gov	Inferno	Muh	CERN	6/6/2066 6:00:00	6/12/2066 6:00:00	1
	7/23/2024 13:03:47	peter@puff.org	Inferno	Peter	Death Star 2.34	5/5/2055 6:00:00	5/5/2055 17:00:00	1
	7/23/2024 13:04:54	Bob@Bobberson.bob	Inferno	Bob Bobberson	Death Star 2.34	4/4/2044 20:00:00	4/5/2044 8:00:00	1
	7/23/2024 13:23:17	marco.polo@italy.tv	Inferno	Marco	Flux Capacitor	8/8/2070 8:07:00	8/8/2070 9:07:00	1
	7/23/2024 13:26:57	m@b.com	Inferno	M	Microwave	2/2/2024 10:00:00	2/2/2024 12:00:00	1
	7/23/2024 13:36:05	a@b.com	Inferno	Abc	CERN	6/6/2026 6:06:00	6/6/2026 8:00:00	1

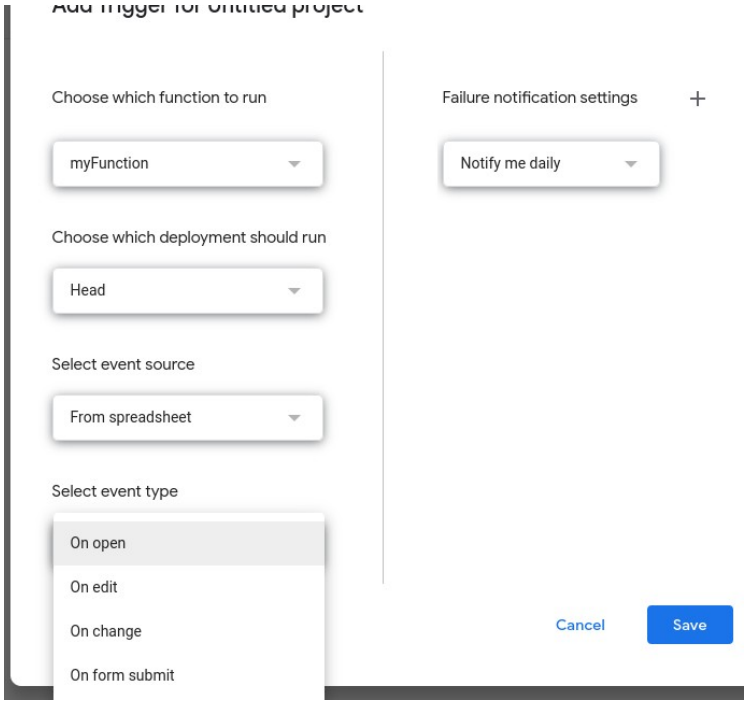
Now, this is already pretty good, but you are the only one who knows who has reserved what. We are now going to use Google's appscript scripting language to read the data from the sheet and create calendar events from it. Open the extension drop down menu, then open appscript. Editing the form will change the layout of the sheet. With appscript, we have to know in which cells are the relevant data bits. So make sure you are perfectly happy with the questions before you continue.

Now we need to create a new Google calendar that will hold the events. Once you have that, click on the little dots next to it and go to "Settings and Sharing". Make it public and copy the calendar ID. You can paste it somewhere in the sheet for now. Below, is the code that will make all of this work. Just put in your calendar ID instead of `your@group.calendar.Google.com`. If your values are at different positions you will have to adjust that as well.

```
function myFunction() {
var spreadsheet = SpreadsheetApp.getActiveSheet(); // We will be using the sheet we just looked at
var calendarId = "your@group.calendar.Google.com" // The script needs to know what caledar to use
var eventCal = CalendarApp.getCalendarById(calendarId);
var bookings = spreadsheet.getRange("b2:g500").getValues(); // We are going to read in all the cells that we might need


for (x=0; x<bookings.length; x++) { // the loop will be executed for each line that we just read in
var shift = bookings[x]; // We are now grabbing the first line from the bookins matrix
var pos = x
var cell_check = spreadsheet.getRange(pos+2,8) // This is the CELL where our little "1" was put, Note the the syntax is (y,x) for some reason
var check = spreadsheet.getRange(pos+2,9).getValue(); //This the value in the CELL of the "1", namely "1"
```

```
//console.log (check) console.log is usefull for debugging
if (check == 1){ // This IF is only excecuted if there is a 1 in the 1 cell
var startTime = shift[4]; //Here we just grab the data we need for the calendar
event from the lines, or 1D matrix. The positions are not what you see in the sheet
because we did not read in the whole sheet but started at collum B, which is at
position 0
var endTime = shift[5];
var user = shift[2];
var tool = shift[3];
var userandtool = user + " - "
var userandtool = userandtool + tool //Just making it look a little nicer
//console.log (userandtool)

if (startTime < endTime){ // if the times line up a event is created and the "1"
becomes a "2". This ensures that when the script runs again, no duplicate events are
created
eventCal.createEvent(userandtool, startTime, endTime);
cell_check.setValue(2);
}
if (startTime > endTime){ //nothing happens if the dates a wrong.
cell_check.setValue("Time mismatch");
}
}
}

}
```

The last thing to do is to make sure the script runs every time a form is submitted. For that in the appscript editor, on the left, click on “triggers” and set it to run upon form submission:



Congratulations!You now should have a working calendar app that lets people reserve things and you can run statistics on the usage.