

MSA Iftar Google Form Set-up Guide

1.0 Introduction:

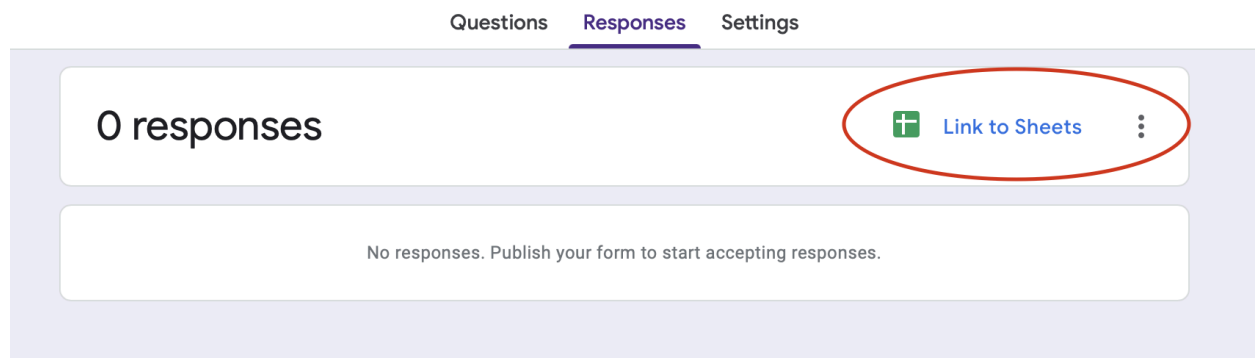
This Google Form Script automatically creates QR codes for people when they come to iftars. When a user submits a form, the script generates a unique ticket code, saves it in the sheet, and emails the user a QR code linked to their ticket page. After scanning the QR code, we can view the ticket's details and verify that they only signed up once for that day. If valid, a "Check In" button is displayed, allowing us to mark the ticket as "Used" with a single tap. The system ensures that tickets can only be used once per event and prevents duplicate submissions, providing a seamless and efficient automated check-in process.

2.0 Set Up Instructions:

Below are the steps required to add this functionality to the MSA's Google Form.

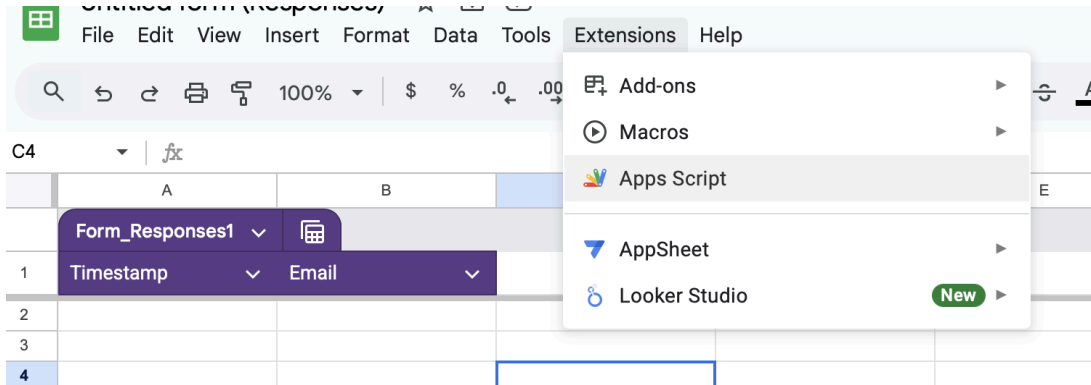
2.1 Link the Google Form to Sheets and View It

Connect the Google form with Sheets to get the respondents and create the new custom script. After it's linked, open the new sheet.



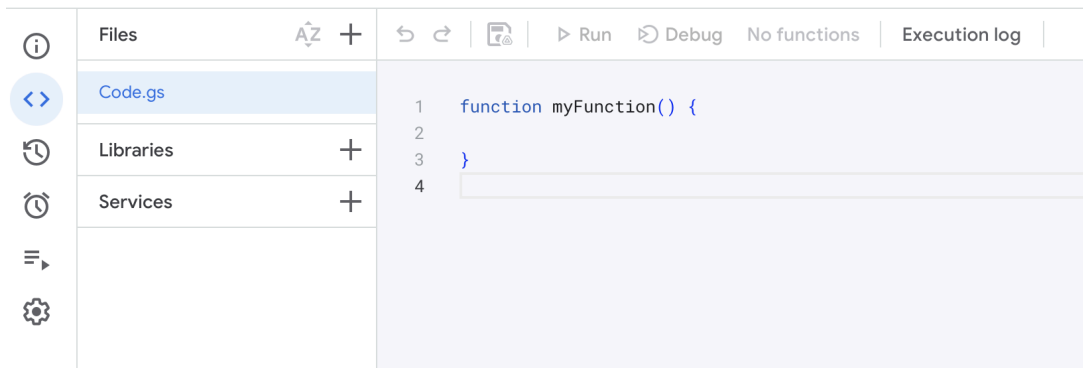
2.2 Open a New AppScript inside Google Sheets

With the opened Sheet open AppScript by going to **Extension > App Script**



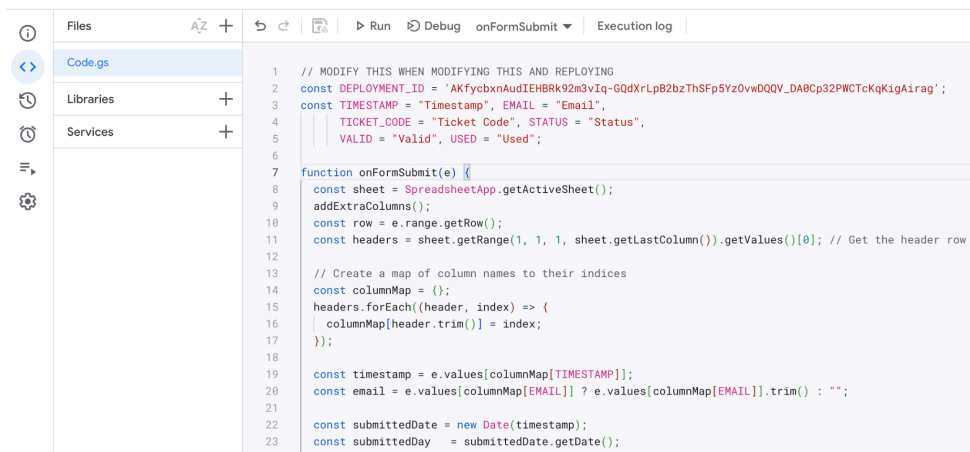
2.3 Add the Custom Code to App Script

When you open App Script you should be automatically inside a Code.gs file with an empty function.



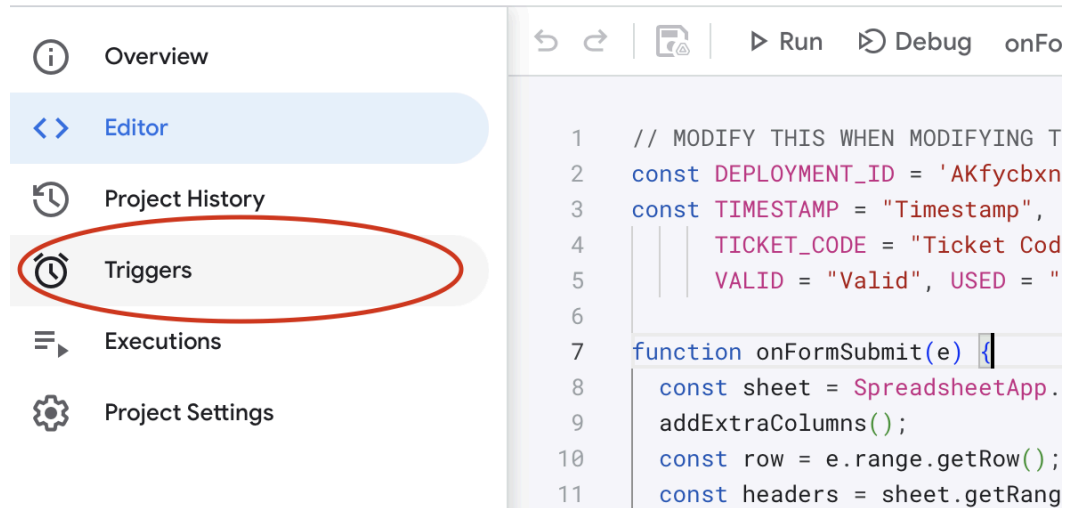
Visit this [link](#) and copy-and-paste the code inside. Replace the default function with the copied code in the Code.gs file. Save after you're done.

Github Link: <https://github.com/saifaldin14/MSA-Iftar/blob/main/GoogleFormSignIn.js>

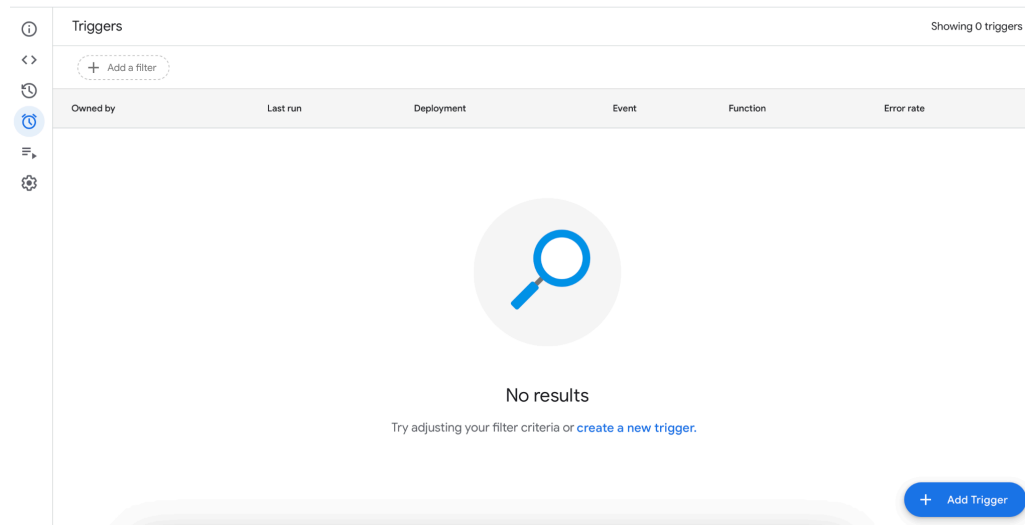


2.4 Add a New Trigger for When Users Submit the Form

Go to the **Triggers** tab and create a new trigger to be called when users submit the form.



You should see a page similar to this:



Click on the **Add Trigger** button in the bottom right corner. For the “Choose which function to run” option select “onFormSubmit”, for the “Select event source” option select “From spreadsheet” and for the “Select event type” option select “On form submit”. Your dialog should look like this:

Choose which function to run

onFormSubmit

Choose which deployment should run

Head

Select event source

From spreadsheet

Select event type

On form submit

Failure notification settings



Notify me daily

Cancel

Save

Click on **Save** and create the new trigger.

2.4.1 Script Authorization Error

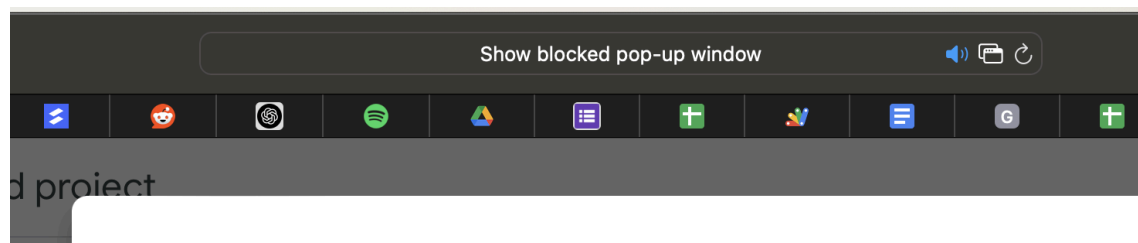
If you get an error like this at the bottom of the dialog:

Cancel

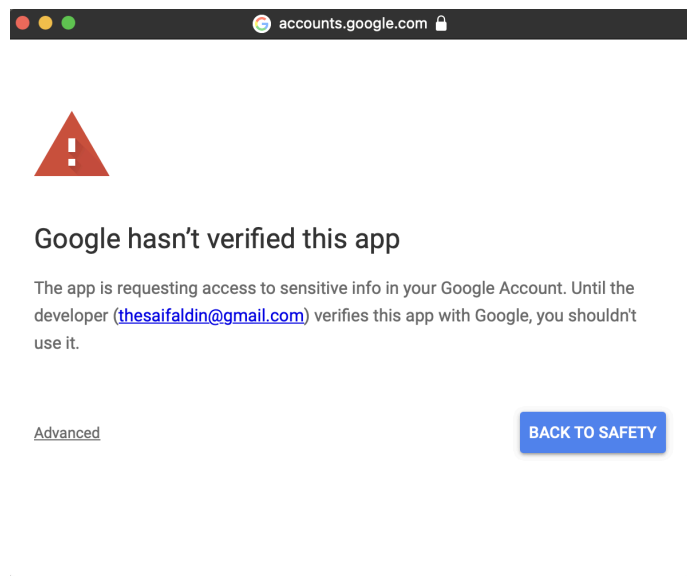
Save

Script authorization failed. Please check your pop-up blocker settings and try again.

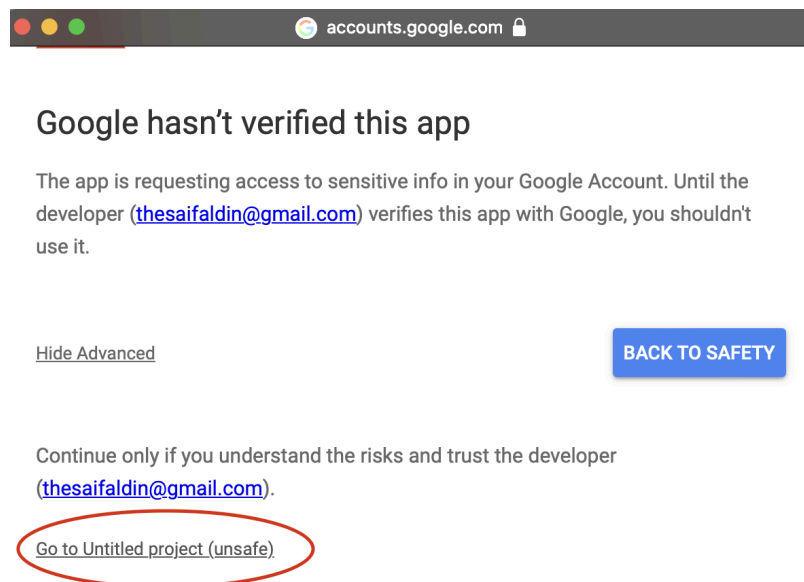
Then show the blocked pop-up in your browser to open the Google Authorization screen.



After signing in if you see a page like this:



Then click on **Advanced > Go to [PROJECT_NAME]**. Allow Google to access the settings for this app. This is a custom script we are using for the Google form and not an official app released on the Google app store, that's why this error shows up.



You should now see a screen similar to this:

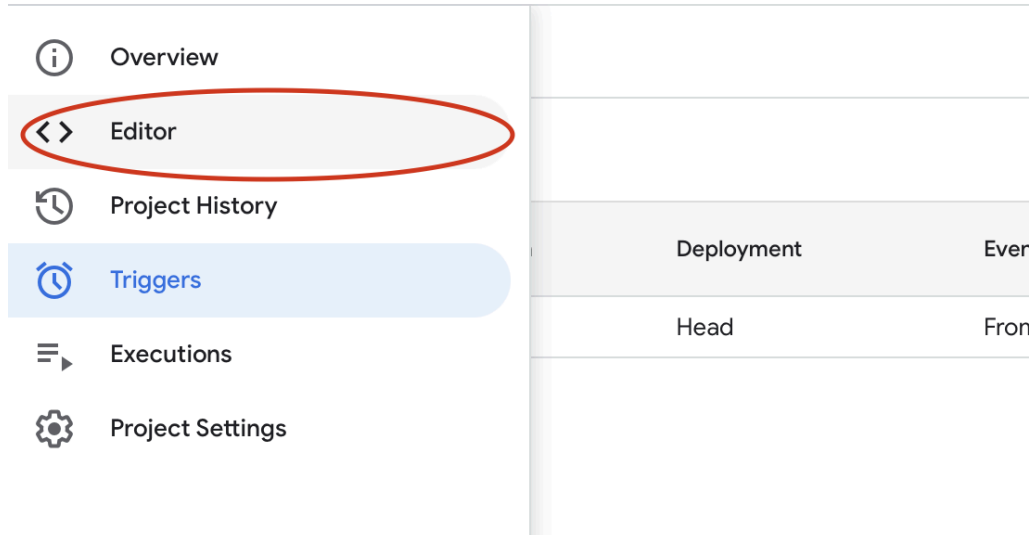
Triggers Showing 1 trigger

[+ Add a filter](#)

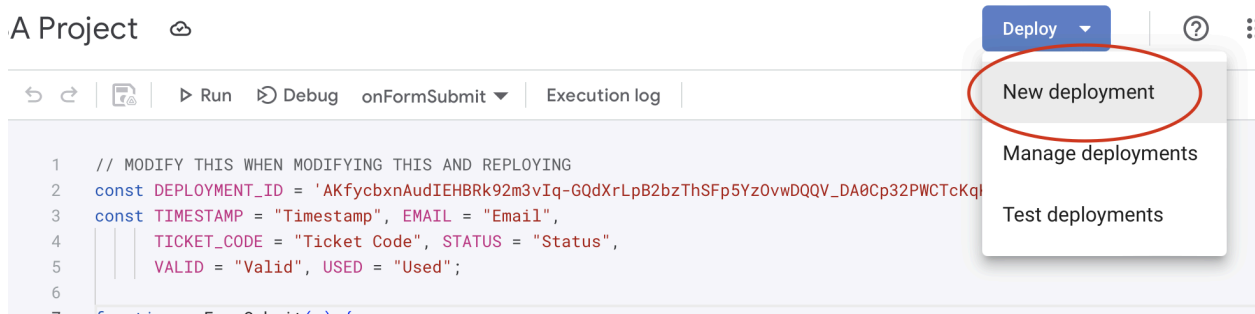
Owned by	Last run	Deployment	Event	Function	Error rate
Me	-	Head	From spreadsheet - On form submit	onFormSubmit	-

2.5 Deploy the Temporary Web Page

For the attendants to verify people for the Iftar when they scan the QR code they should see a web page allowing them to sign people in. Navigate back to the **Editor** tab when the code is:




Create a new deployment by going to **Deployment > New Deployment**:



You should now see a dialog like this:

New deployment

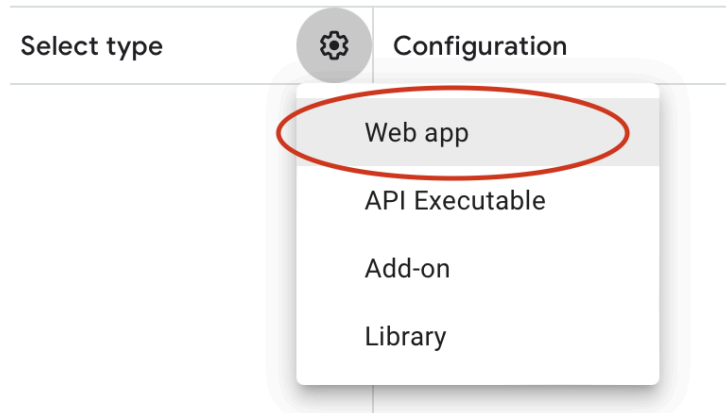
Select type	Configuration
	<div><div></div><div>Please select a deployment type</div></div>

Cancel

Deploy

Create a Web Page deployment by going to **Gear Icon** next to **Select Type > Web app**:

New deployment



Now you should see a configuration modal screen. Add a description (optional), and set the access level to “Anyone” to allow people to sign-in using any email:

The image shows the 'Configuration' modal screen for a 'Web app'. The 'Description' field contains 'Ifatar Sign In'. The 'Web app' section shows 'Execute as' set to 'Me (thesaifaldin@gmail.com)' and 'Who has access' set to 'Anyone'. A blue banner at the bottom states: 'This can also be used as a library. Learn more'.

Click on **Deploy** and copy the Deployment ID:

The image shows the 'New deployment' page after deployment. It shows a success message: 'Deployment successfully updated.' Below it, the version is 'Version 1 on Mar 8, 2025, 2:37 PM'. The 'Deployment ID' is highlighted with a red oval: 'AKfycbwWR2tZmoWBRkNwGFc9lI8Z8iMrY1VLmwqYKeZH6L3fn4kD5GvO3GxfnkmLncfOUcSb'. A 'Copy' button is next to the ID. Below this, the 'Web app' section shows the 'URL' as 'https://script.google.com/macros/s/AKfycbwWR2tZmoWBRkNwGFc9lI8Z8iMrY1VLmwqYKeZH6L3fn4kD5GvO3GxfnkmL...' and a 'Copy' button.

In the code in the **Code.gs** replace the value in the **DEPLOYMENT_ID** variable on **Line 2** with the copied Deployment ID

```
1 // MODIFY THIS WHEN MODIFYING THIS AND REPLYING
2 const DEPLOYMENT_ID = 'AKfycbwWR2tZmoWBRkNwGfc9l18Z8iMrY1VLmwqYKeZH6L3fn4kD5Gv03GxfnkmlNcf0UcSb';
3 const TIMESTAMP = "Timestamp", EMAIL = "Email",
4   TICKET_CODE = "Ticket Code", STATUS = "Status",
5   VALID = "Valid", USED = "Used";
6
7 function onFormSubmit(e) {
8   const sheet = SpreadsheetApp.getActiveSheet();
9   addExtraColumns();
```

Boom! Now you're done and it's applied to the Google Form!