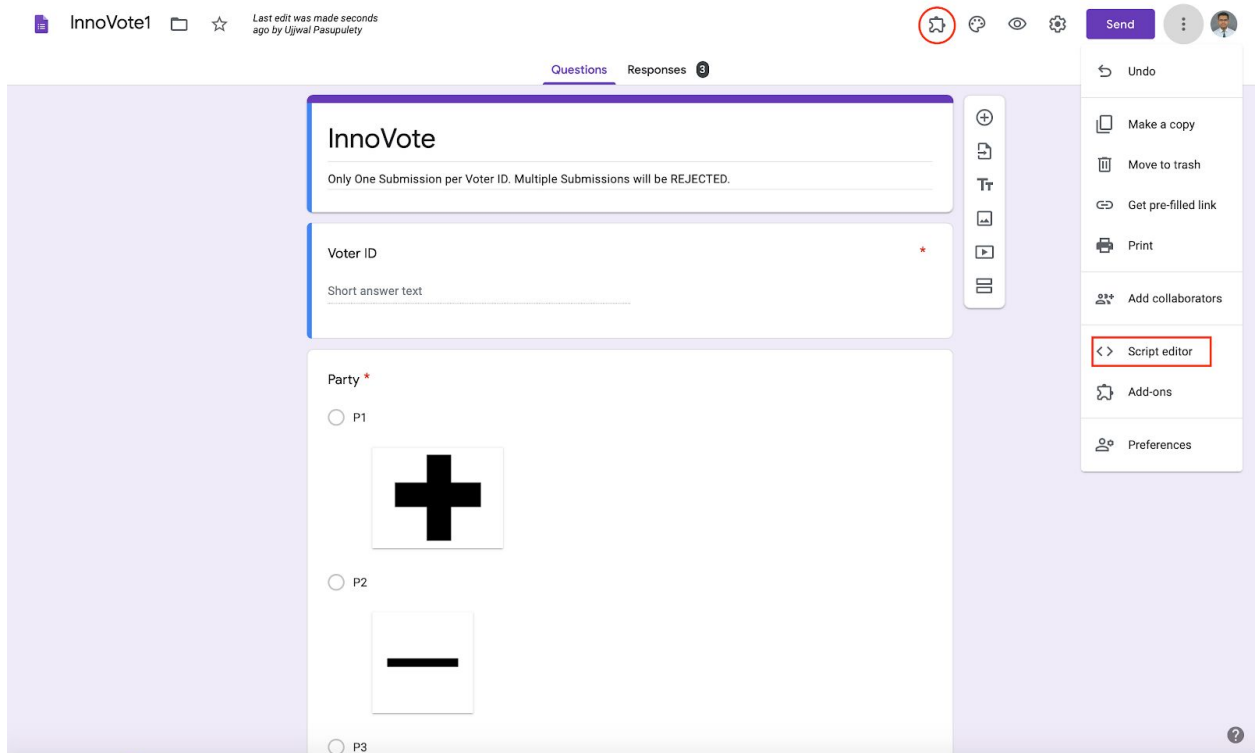


InnoVote Support Document - by Ujjwal Pasupulety

This document contains detailed instructions on how to implement the working prototype of InnoVote

Enable Form CAPTCHA

1. In the Google Form, click the 3 dot menu at the upper right hand corner and open the Script editor(outlines in the red box).

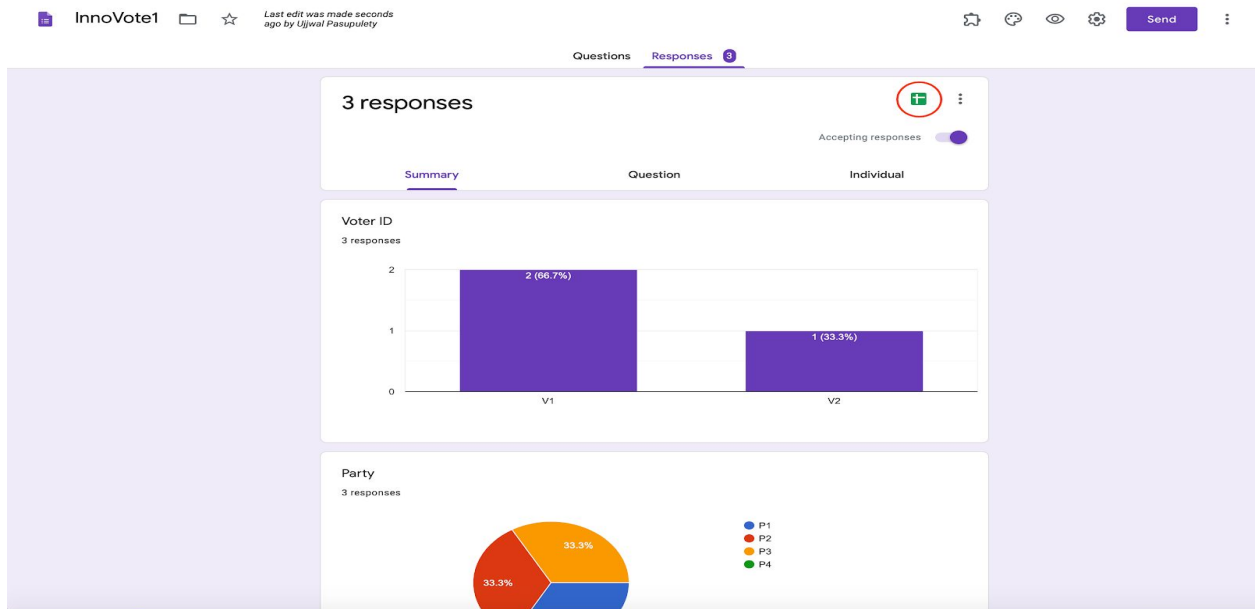


The screenshot shows a Google Form titled "InnoVote" with the subtitle "Only One Submission per Voter ID. Multiple Submissions will be REJECTED." The form contains three questions: "Voter ID" (Short answer text), "Party" (Multiple choice), and "Party" (Multiple choice). The "Party" question has three options: "P1" with a plus sign icon, "P2" with a minus sign icon, and "P3" with a minus sign icon. The "Script editor" option in the top right menu is highlighted with a red box.

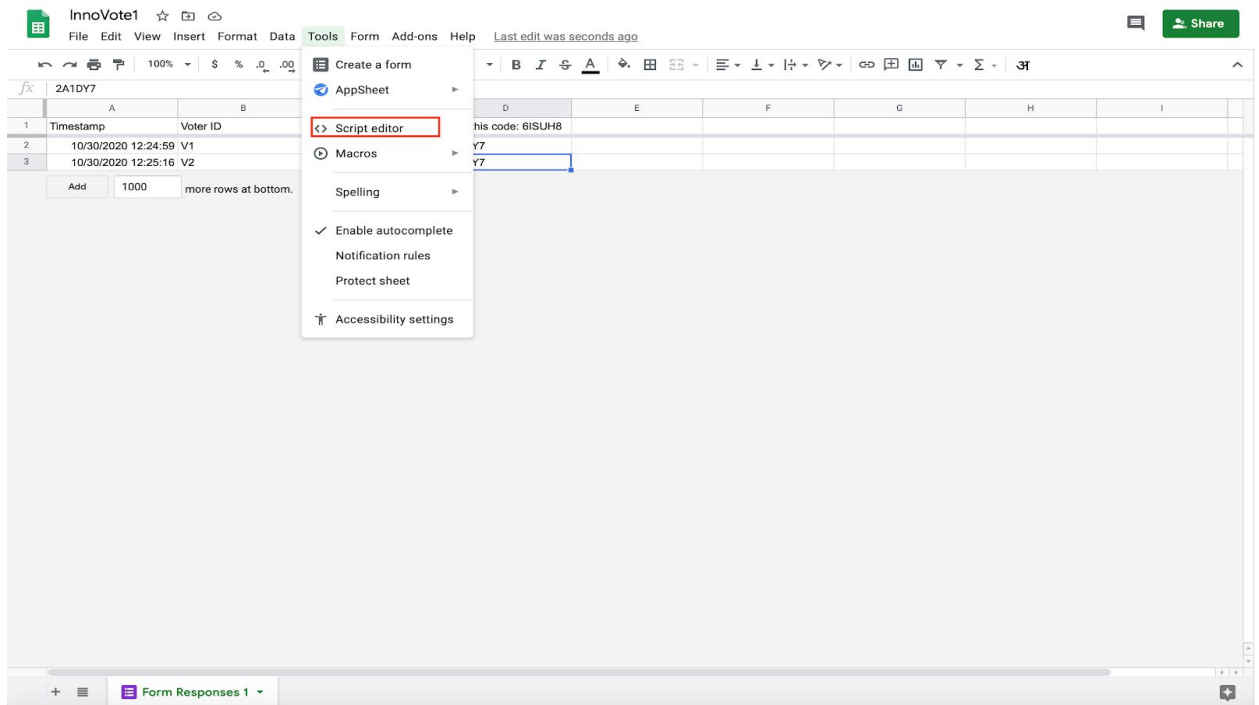
2. Copy the code from this [blog post](#) and paste in the script window. It will get saved automatically.
3. Refresh the Google Form, you will notice a new icon in the shape of a jigsaw piece(circled red) will appear. This will allow you to enable the CAPTCHA functionality in the form.

Duplicate removal from Google Sheet

1. Go to the responses section of the form and click the Green icon(circled in red). This will redirect you to a Google Sheet containing all the voter responses.



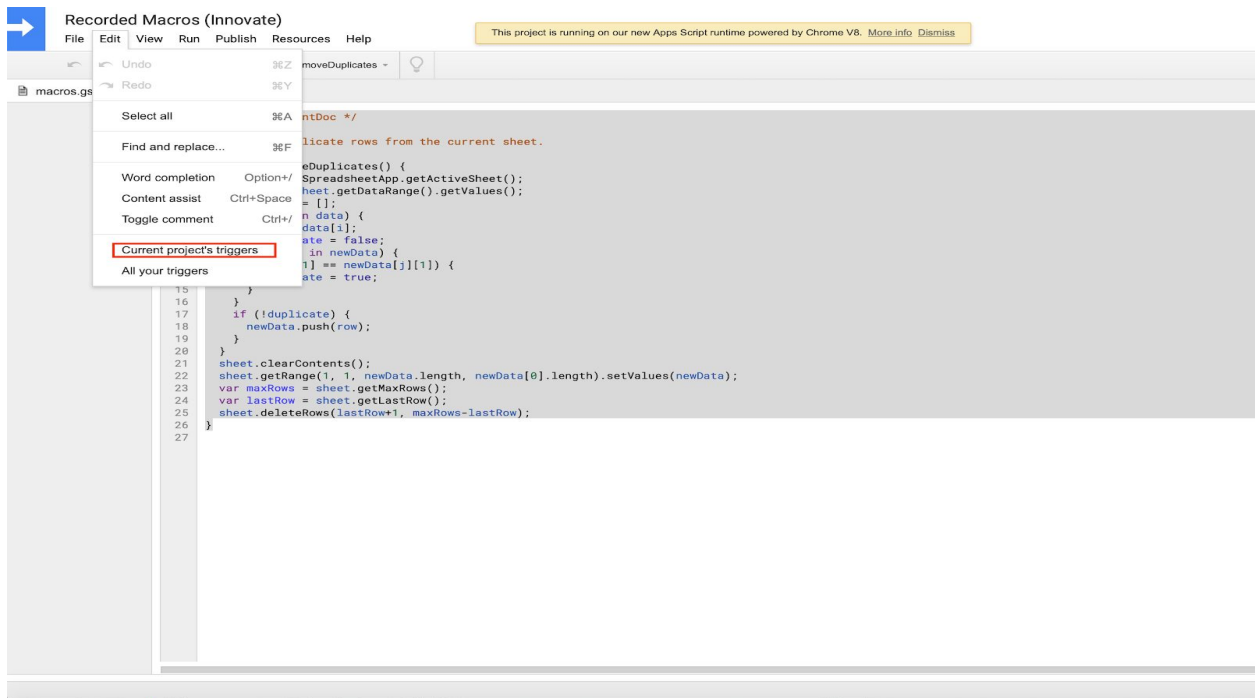
2. The Google Sheet's script editor can be accessed through the Tools menu.



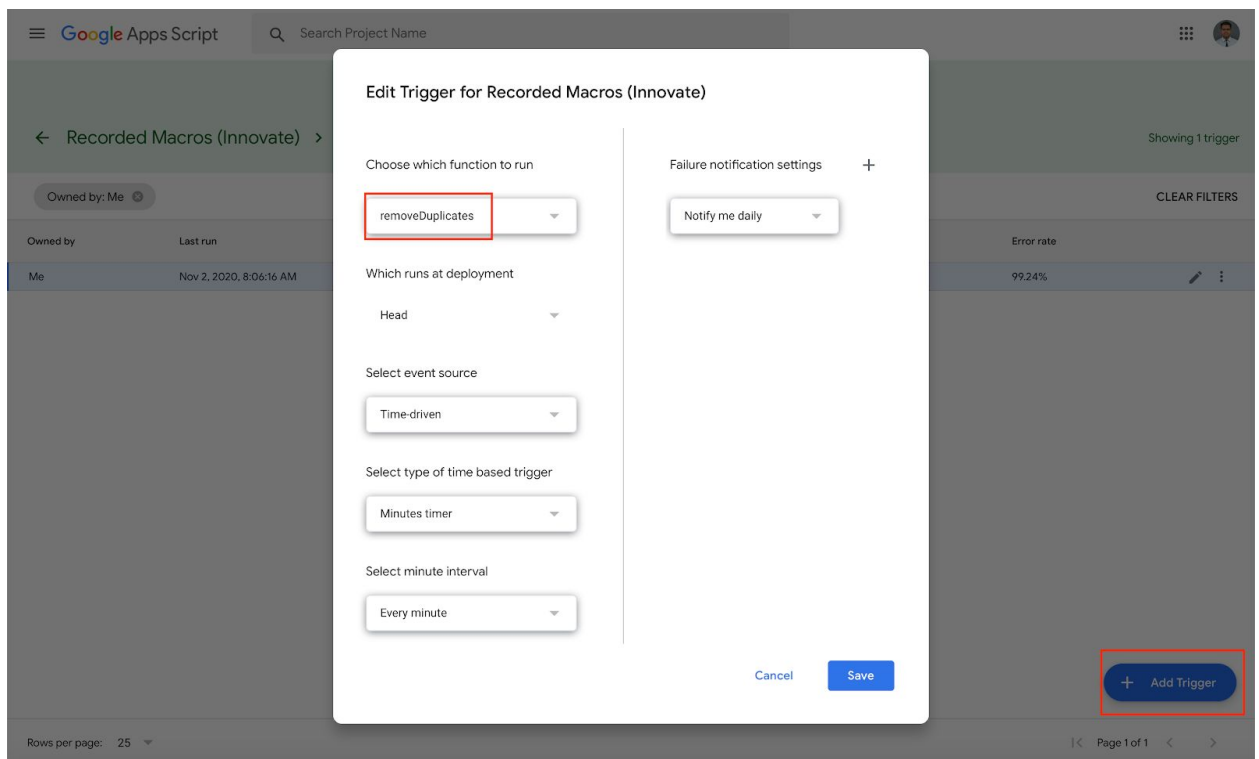
3. Copy the following script into the Apps Script window

```
/** @OnlyCurrentDoc */
/**
 * Removes duplicate rows from the current sheet.
 */
function removeDuplicates() {
  var sheet = SpreadsheetApp.getActiveSheet();
  var data = sheet.getDataRange().getValues();
  var newData = [];
  for (var i in data) {
    var row = data[i];
    var duplicate = false;
    for (var j in newData) {
      if (row[1] == newData[j][1]) {
        duplicate = true;
      }
    }
    if (!duplicate) {
      newData.push(row);
    }
  }
  sheet.clearContents();
  sheet.getRange(1, 1, newData.length,
newData[0].length).setValues(newData);
  var maxRows = sheet.getMaxRows();
  var lastRow = sheet.getLastRow();
  sheet.deleteRows(lastRow+1, maxRows-lastRow);
}
```

4. Now a trigger must be added to enable this script to run every minute. To do this, go to the Edit tab in the Apps script window and select “Current Project’s Triggers”.

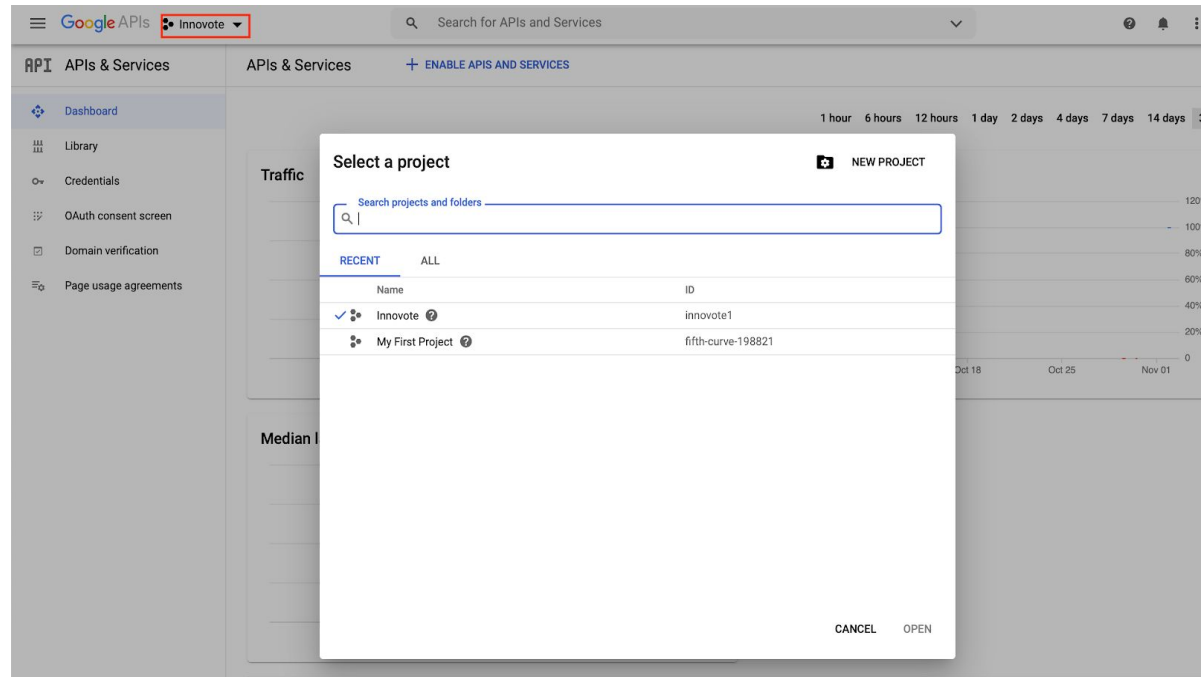


5. Click “Add Trigger”. This will result in the following pop-up window where the trigger can be selected to run in a periodic manner. “removeDuplicates” is the function name that was used in Step 3.

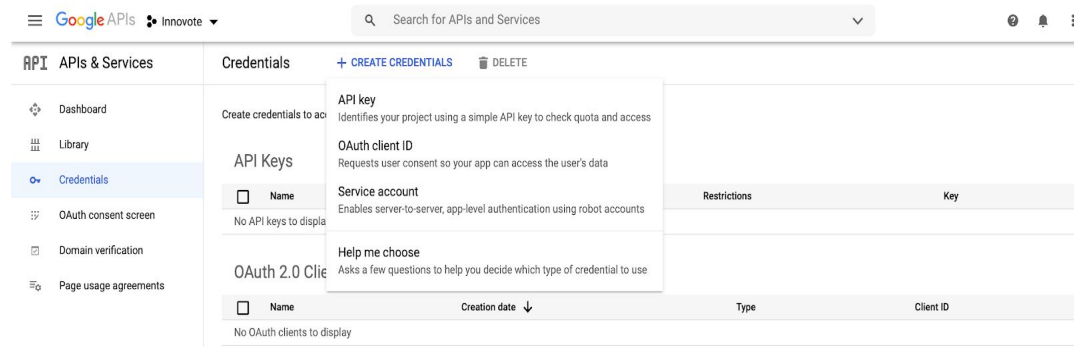


Setting up a service account

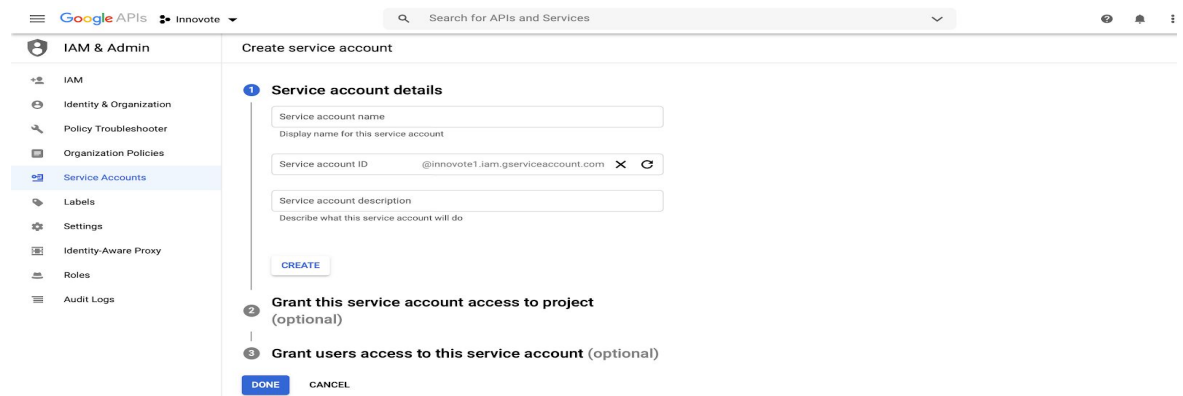
1. Go to <https://console.developers.google.com> and create a new Project



2. Go to "Credentials" and under the "Create credentials" select "Service Account"



3. Fill the form and ignore the optional choices



- ```

1 {
2 "type": "service_account",
3 "project_id": "innovotel",
4 "private_key_id": "[REDACTED]",
5 "private_key": "[REDACTED]",
6 "client_email": "[REDACTED]@innovotel.com",
7 "client_id": "[REDACTED]",
8 "auth_uri": "https://accounts.google.com/o/oauth2/auth",
9 "token_uri": "https://accounts.google.com/o/oauth2/token",
10 "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
11 "client_x509_cert_url": "[REDACTED]"
12 }
13

```

- FormAdd-onsHelpLast edit was seconds ago

Default (Arial)10B I G A

|   |                       |   |   |   |   |   |
|---|-----------------------|---|---|---|---|---|
| C | D                     | E | F | G | H | I |
|   | Type this code: DZ92S |   |   |   |   |   |
|   | 2A1DY7                |   |   |   |   |   |
|   | 2A1DY7                |   |   |   |   |   |

Share

Share with people and groups

Ujjwal Pasupulety (you)

ujjwalpasupulety@gmail.com

Owner

innovote

[Redacted email address]

Editor

[Send feedback to Google](#)

Done

Get link

Restricted Only people added can open with this link

[Change to anyone with the link](#)

[Copy link](#)

## Running the Python scripts from the Github repository

```
[UPASUPUL-M-M0PH:Personal upasupul$ python3 genvoterdb.py]
[UPASUPUL-M-M0PH:Personal upasupul$ python3 viewdb.py]
('V1', 0, None)
('V2', 0, None)
('V3', 0, None)
[UPASUPUL-M-M0PH:Personal upasupul$ python3 innovote.py]
['V1', 'P2']
UPDATE voterlist SET vote = 'P2', voted = 1 WHERE id = 'V1' AND voted = 0;
['V2', 'P3']
UPDATE voterlist SET vote = 'P3', voted = 1 WHERE id = 'V2' AND voted = 0;
[UPASUPUL-M-M0PH:Personal upasupul$ python3 viewdb.py]
('V1', 1, 'P2')
('V2', 1, 'P3')
('V3', 0, None)
[UPASUPUL-M-M0PH:Personal upasupul$ python3 tally.py]
{'P1': 0, 'P2': 1, 'P3': 1, 'P4': 0}
[UPASUPUL-M-M0PH:Personal upasupul$]
```

1. genvoterdb.py - Creates a toy SQL database file called “voterlist.db”
2. viewdb.py - Displays the voter list table stored in the voterlist.db file. Initially, no one has voted and the *Voted* flag is set to 0. Also the *Party* is set to None.
3. innovote.py - pulls data from the Google sheet using the JSON key and constructs SQL queries that directly modify the SQL database.
4. tally.py - tallies the votes by scanning every entry in the SQL database and displays the final vote count.