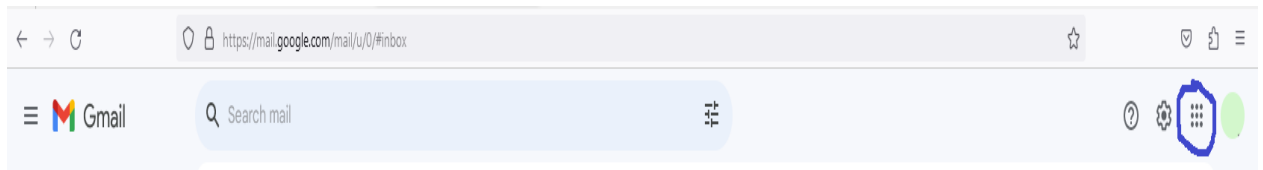
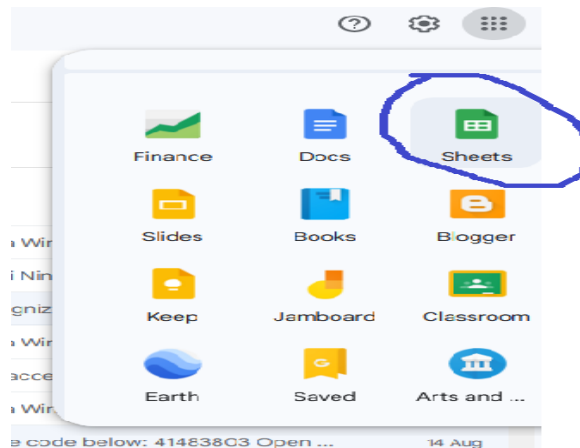


HOW TO LINK YOR GOOGLE SHEETS TO THE FRONT-END CODE:

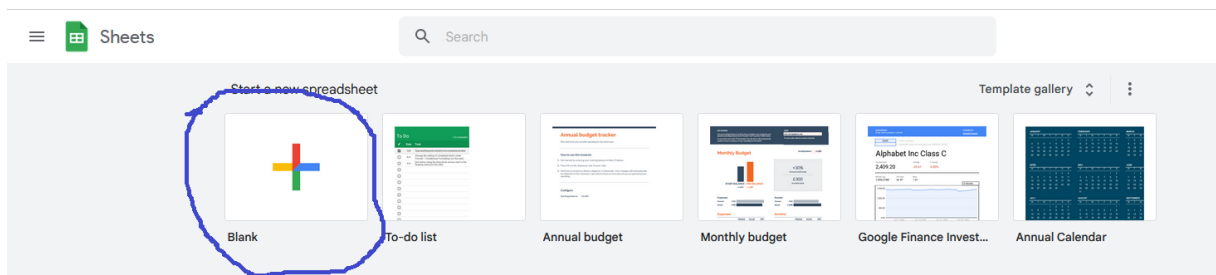
Step 1. Click on the dots as shown in the below image



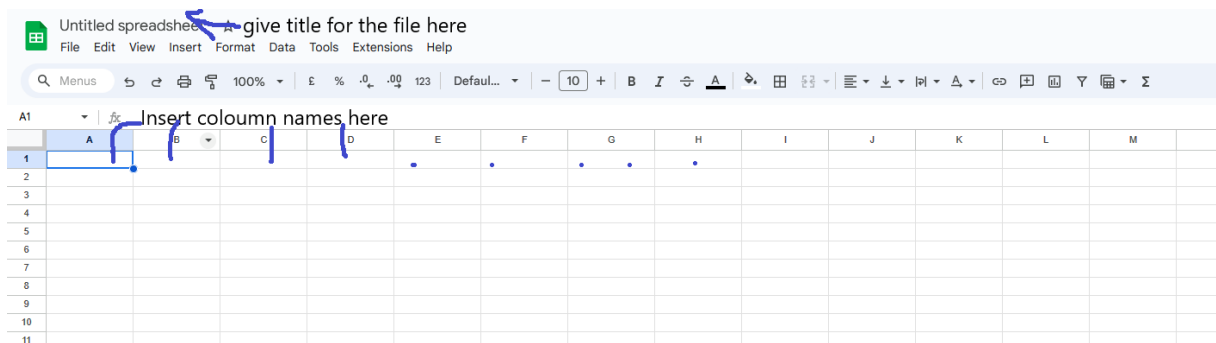
Step 2. After clicking on the dots then select the sheets option



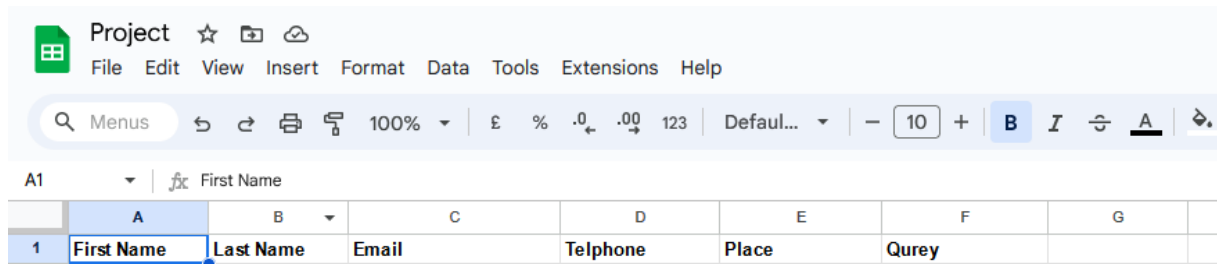
Step 3. After selecting the sheets then select blank sheet.



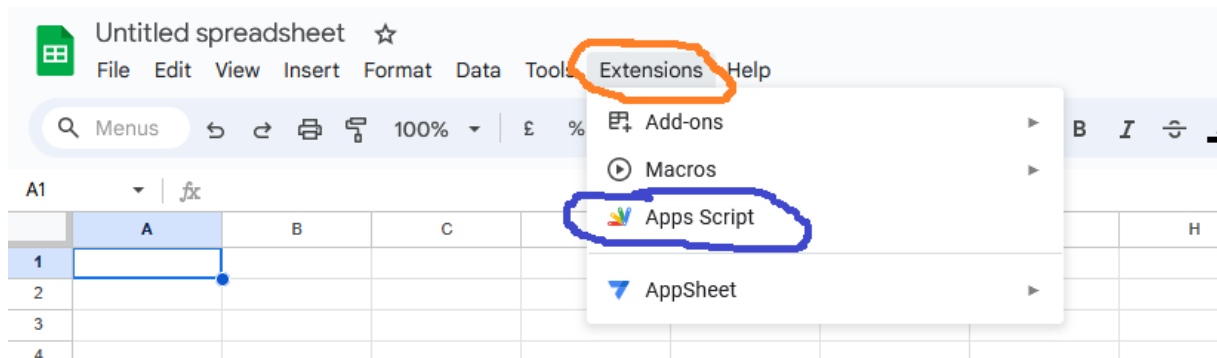
Step 4. After selecting blank sheet then rename the sheet and give the column name in the sheets which are same as the name in the code and showed in the next image.



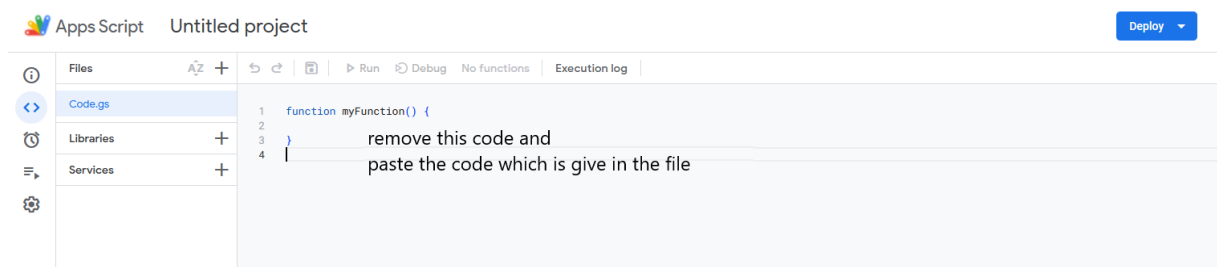
Step 5. Rename the sheet and give the column name as shown in the image



Step 6. After renaming select extensions and then select app script.



Step 7. After selecting app script remove the code and then copy paste the code which is given in this file.



Code to copy paste: -

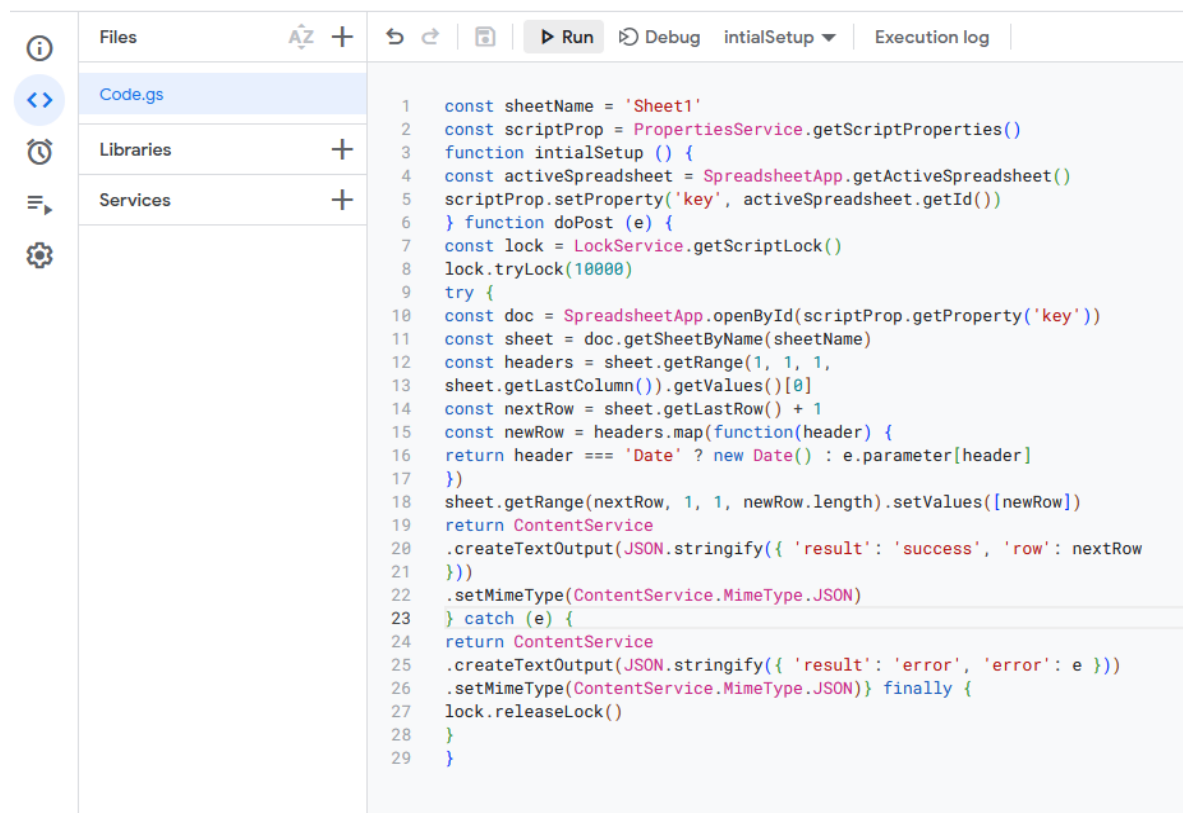
```
const sheetName = 'Sheet1'  
  
const scriptProp = PropertiesService.getScriptProperties()  
  
function intialSetup () {  
  
const activeSpreadsheet = SpreadsheetApp.getActiveSpreadsheet()  
  
scriptProp.setProperty('key', activeSpreadsheet.getId ())  
  
}  
  
function doPost(e) {  
  
const lock = LockService.getScriptLock()  
  
lock.tryLock(10000)  
  
try {  
  
const doc = SpreadsheetApp.openById(scriptProp.getProperty('key'))
```

```

const sheet = doc.getSheetByName (sheetName)
const headers = sheet.getRange(1, 1, 1,sheet.getLastColumn()).getValues()[0]
const nextRow = sheet.getLastRow()+1
const newRow = headers.map(function(header) {
return header === 'Date'?new Date(): e.parameter[header]
})
sheet.getRange(nextRow, 1, 1, newRow.length).setValues([newRow])
return ContentService
    .createTextOutput (JSON.stringify({'result':'success','row':nextRow
}))
    .setMimeType(ContentService.MimeType.JSON)
}
catch (e) {
    return ContentService
        .createTextOutput(JSON.stringify({'result':'error','error':e}))
        .setMimeType(ContentService.MimeType.JSON )
}
finally {
    lock.releaseLock()
}
}

```

Step 8. After copy pasting the code as shown in figure then save the code and run the code.



```
1  const sheetName = 'Sheet1'
2  const scriptProp = PropertiesService.getScriptProperties()
3  function initialSetup () {
4  const activeSpreadsheet = SpreadsheetApp.getActiveSpreadsheet()
5  scriptProp.setProperty('key', activeSpreadsheet.getId())
6  } function doPost (e) {
7  const lock = LockService.getScriptLock()
8  lock.tryLock(10000)
9  try {
10 const doc = SpreadsheetApp.openById(scriptProp.getProperty('key'))
11 const sheet = doc.getSheetByName(sheetName)
12 const headers = sheet.getRange(1, 1, 1,
13 sheet.getLastColumn()).getValues()[0]
14 const nextRow = sheet.getLastRow() + 1
15 const newRow = headers.map(function(header) {
16 return header === 'Date' ? new Date() : e.parameter[header]
17 })
18 sheet.getRange(nextRow, 1, 1, newRow.length).setValues([newRow])
19 return ContentService
20 .createTextOutput(JSON.stringify({ 'result': 'success', 'row': nextRow
21 }))
22 .setMimeType(ContentService.MimeType.JSON)
23 } catch (e) {
24 return ContentService
25 .createTextOutput(JSON.stringify({ 'result': 'error', 'error': e }))
26 .setMimeType(ContentService.MimeType.JSON)} finally {
27 lock.releaseLock()
28 }
29 }
```

Step 9. After running the code, it will ask for the permission give the permission then in execution log you will see that execution completed message as shown in the code.

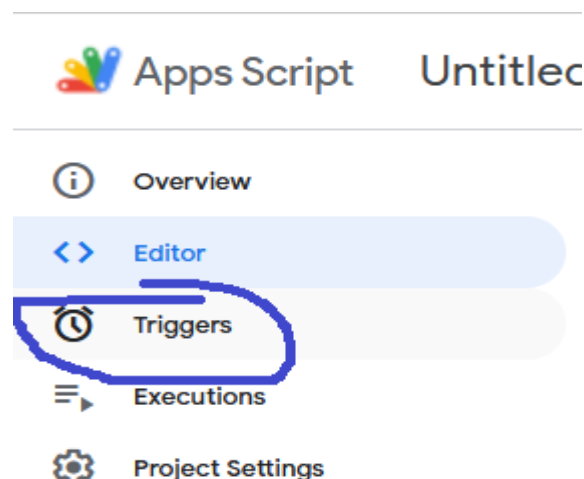


```
24 return ContentService
25 .createTextOutput(JSON.stringify({ 'result': 'error', 'error': e }))
26 .setMimeType(ContentService.MimeType.JSON)} finally {
```

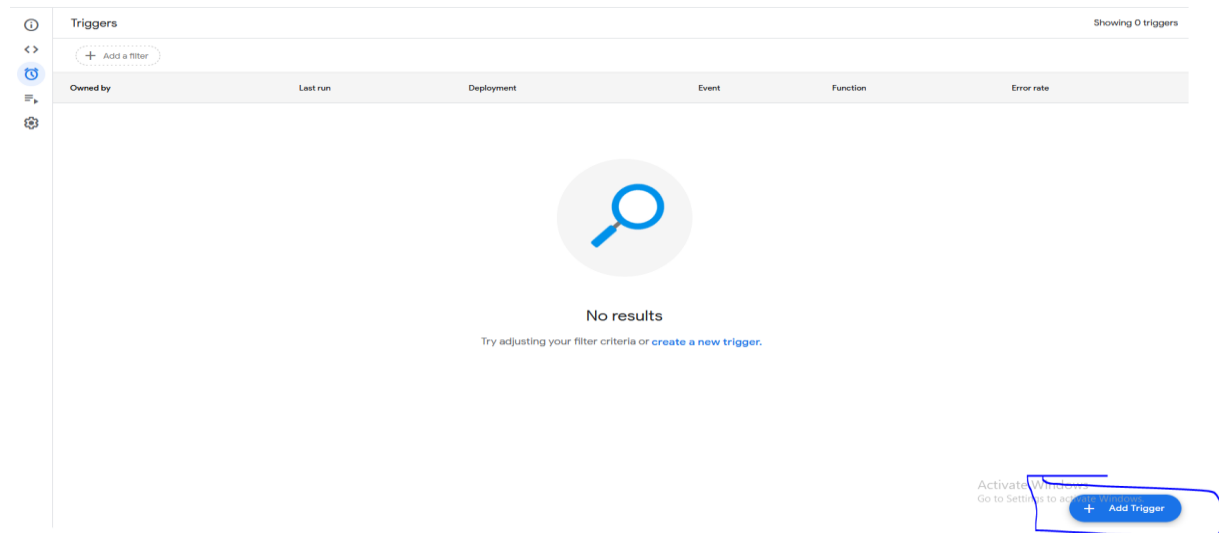
Execution log

9:41:47 PM	Notice	Execution started
9:41:48 PM	Notice	Execution completed

Step 10. After execution completed, select triggers



Step 11. After selecting triggers then select add trigger as shown in figure.



Step 12. After selecting add trigger then new window will appear then make the changes as shown in below figure, after that click on save then it ask for the permission, give the permission.

Add Trigger for Untitled project

Choose which function to run

doPost ▼

Choose which deployment should run

Head ▼

Select event source

From spreadsheet ▼

Select event type

On form submit ▼

Failure notification settings +

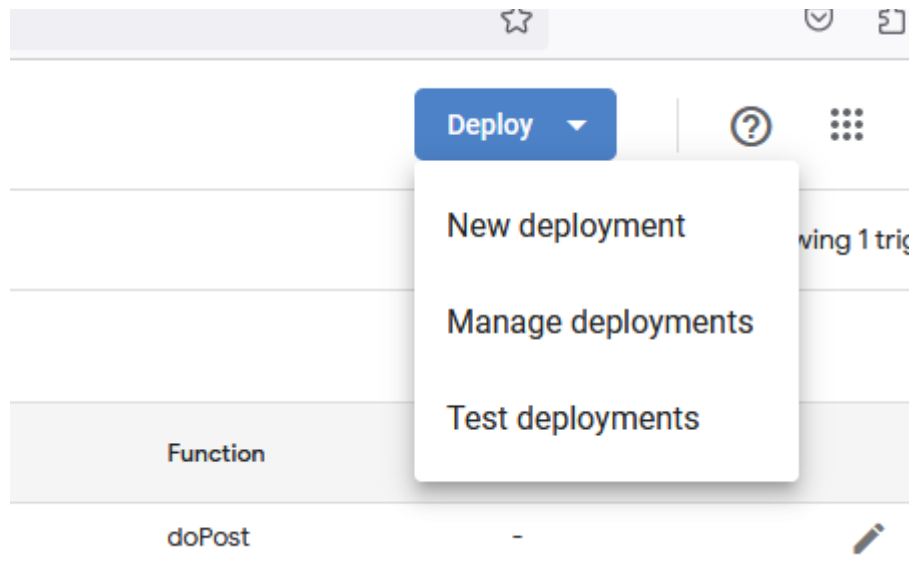
Notify me immediately ▼

[Cancel](#) [Save](#)

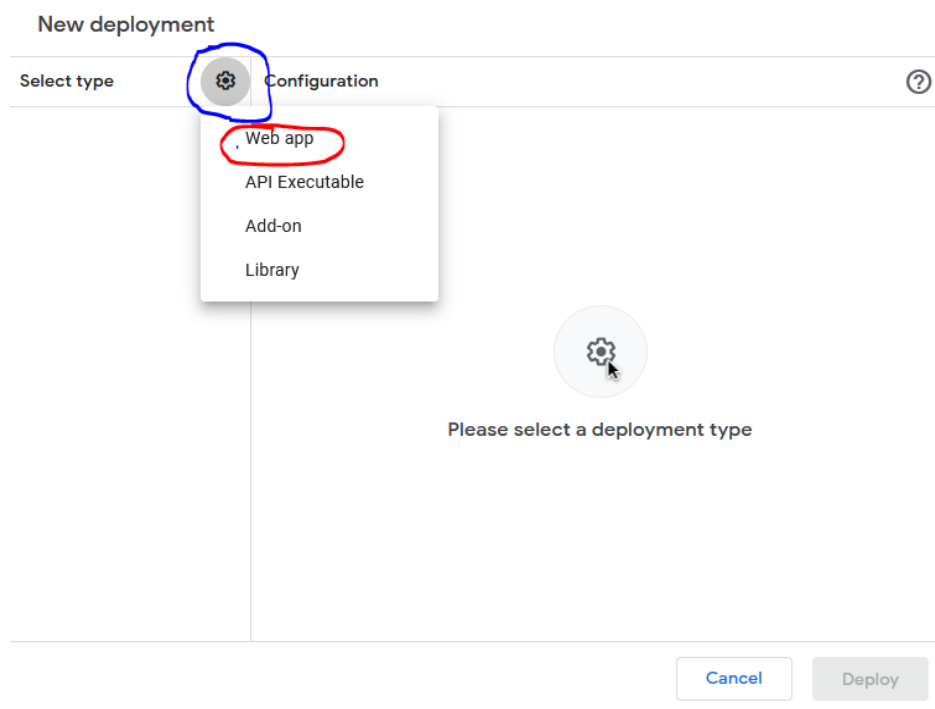
Step 13. After saving the changes then you will get the page lick the below image.

<

Step 14. After that click on deploy and then click on new deployment.



Step 15. After clicking on new deployment, you will get a window as shown below in that click on setting then click on web app.



Step 16. After clicking on web app, you have to make some changes as shown in below figure and then click on deploy.

New deployment

Select type

Web app

Configuration

Description

New description
project title

Web app

Execute as
Me

The web app will be authorized to run using your account data.

Who has access
Anyone

This can also be used as a library. [Learn more](#)

Cancel

Deploy

Step 17. After clicking on deploy you will get a new window which consists of the URL link then copy the link and past the link in place of API address in the code.

New deployment

Deployment successfully updated.

Version 1 on Aug 24, 9:47 PM

Deployment ID
AKfycbzM_kQzlnYJA34r8iiQhFox_1fSFjGgWH0H1_X40Gp8tsNUhDMkHk0aYejKZ_aYcm6cWw
[Copy](#)

Web app

URL
https://script.google.com/macros/s/AKfycbzM_kQzlnYJA34r8iiQhFox_1fSFjGgWH0H1_X40Gp8tsNUhDMkHk0aYejKZ_aY...
[Copy](#)

Done