

ENVIRONMENTAL DATA CAPTURE AND IMAGE STORAGE

List of TextFiles in the SD Card:

File 1. ssid.txt

File 2. password.txt

File 3. scriptDriveUrl.txt

File 4. spreadsheetUrl.txt

File 5. delay.txt

Data to Store in the Text Files in the SD Card:

File 1. Your WiFi SSID

File 2. Your WiFi password

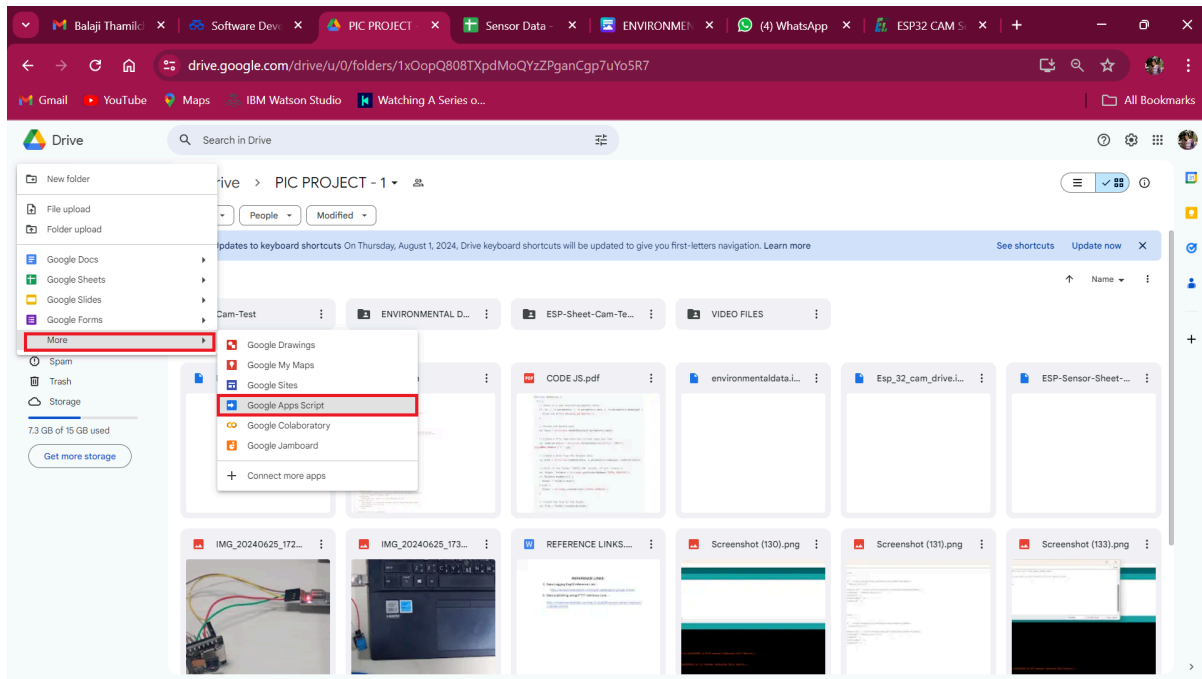
File 3. The WebApp URL Link generated at the end of step 1

File 4. The GSheet URL Link generated at the end of step 4

File 5. The time delay user wants to capture data

Step 1: Google App Script Creation for Drive Access:-

1. Create a folder in Google Drive and name it .
2. Create a Google App Script from the drive, which you want to save the captured images.



3. Paste the following code in the script. In the script, replace this "Replace_with_your_drive_folder_name" with your folder name.

```
function doPost(e) {  
  try {  
    // Check if e and required parameters exist  
    if (!e || !e.parameters || !e.parameters.data ||  
    !e.parameters.mimetype) {  
      throw new Error('Missing parameters');  
    }  
    // Decode the base64 data  
    var data = Utilities.base64Decode(e.parameters.data);  
    // Create a file name with the current date and time  
    var nombreArchivo = Utilities.formatDate(new Date(), "GMT-3",
```

```

"yyyyMMdd_HH:mm:ss") + ".jpg";

    // Create a blob from the decoded data
    var blob = Utilities.newBlob(data, e.parameters.mimetype,
nombreArchivo);
    // Check if the folder "ESP32_CAM" exists, if not, create it
    var folder, folders =
DriveApp.getFoldersByName("Replace_with_your_drive_folder_name");
//Change the folder name with yours
    if (folders.hasNext()) {
        folder = folders.next();
    } else {
        folder =DriveApp.createFolder("Replace_with_your_drive_folder_name");
    }
    // Create the file in the folder
    var file = folder.createFile(blob);
    // Return a success message
    return ContentService.createTextOutput("Completed.");
} catch (error) {
    // Return an error message if something goes wrong
    return ContentService.createTextOutput("Error: " + error.message);
}
}

// Function to handle GET requests
function doGet(e) {
    return ContentService.createTextOutput("This endpoint accepts POST
requests to save images.");
}

// Function for local testing within Google Apps Script Editor
function testDoPost() {
    var e = {
        parameters: {
            data:
Utilities.base64Encode(Utilities.newBlob('dummy_image_data').getBytes()),
            mimetype: 'image/jpeg'
        }
    };
}

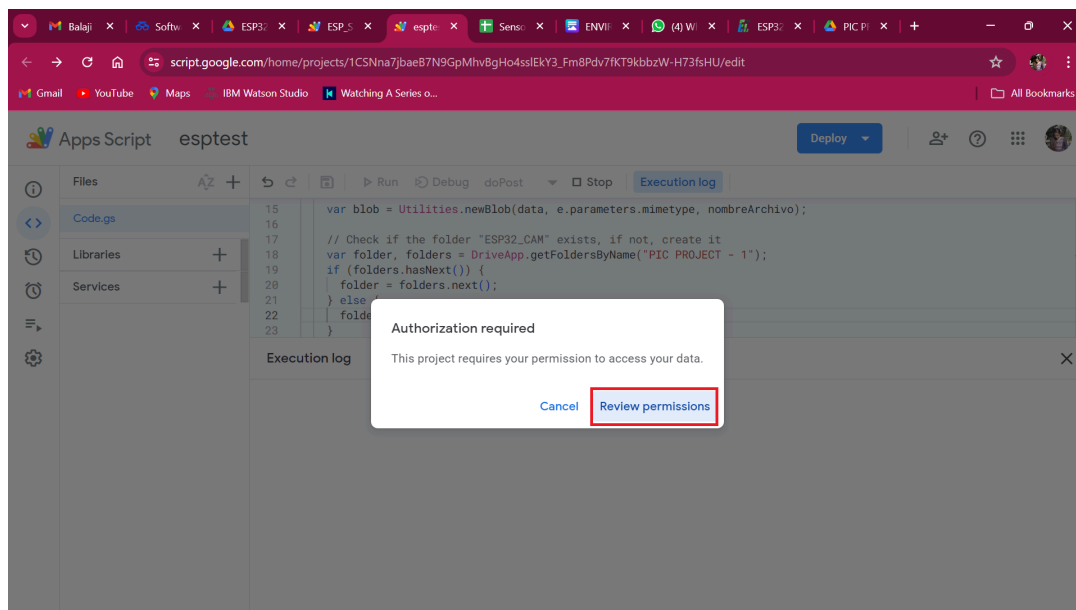
```

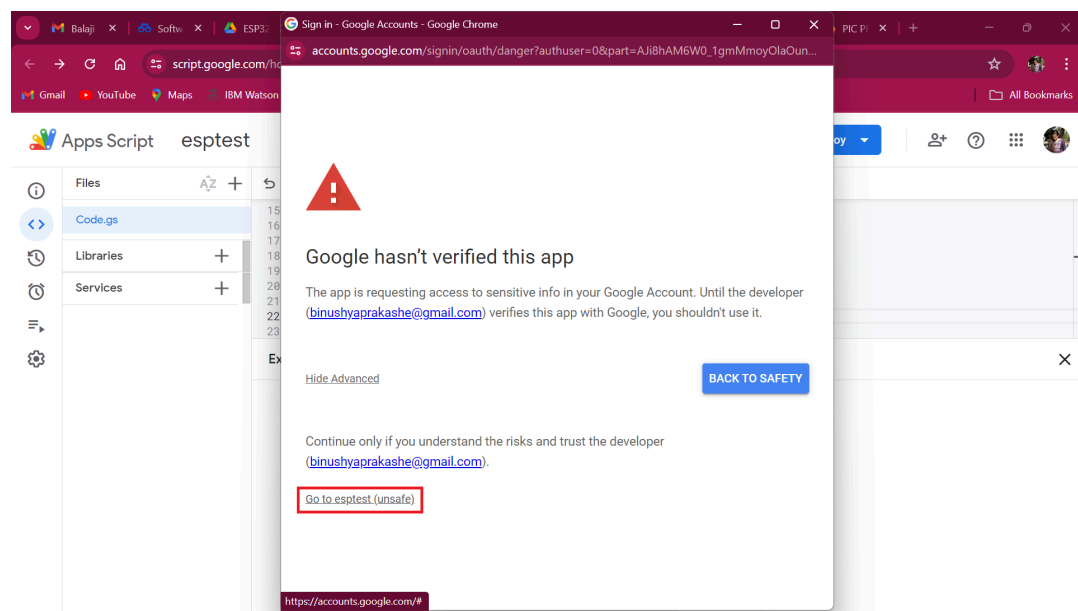
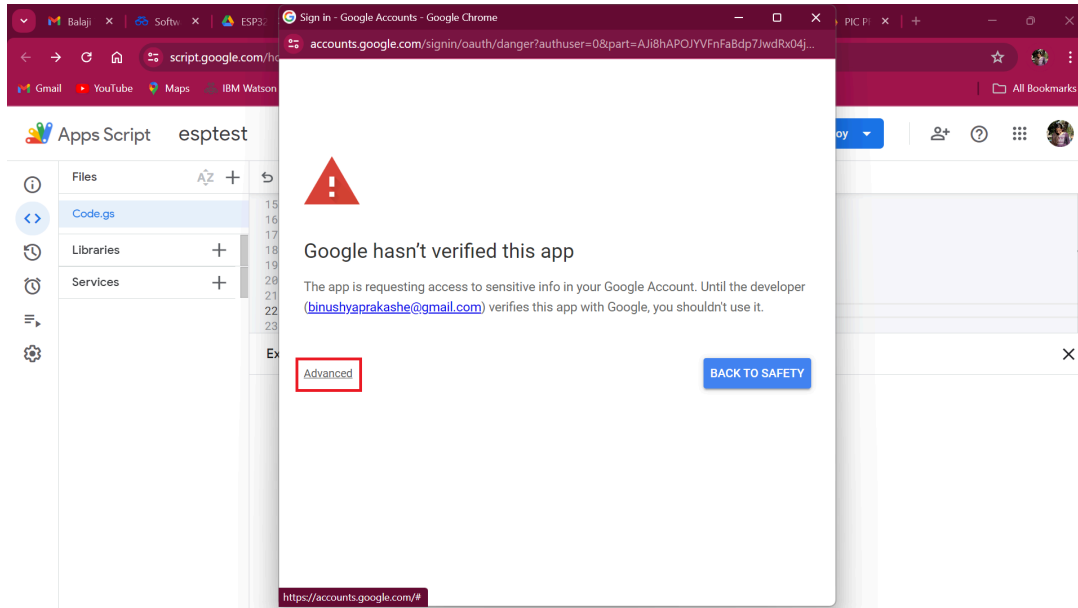
```

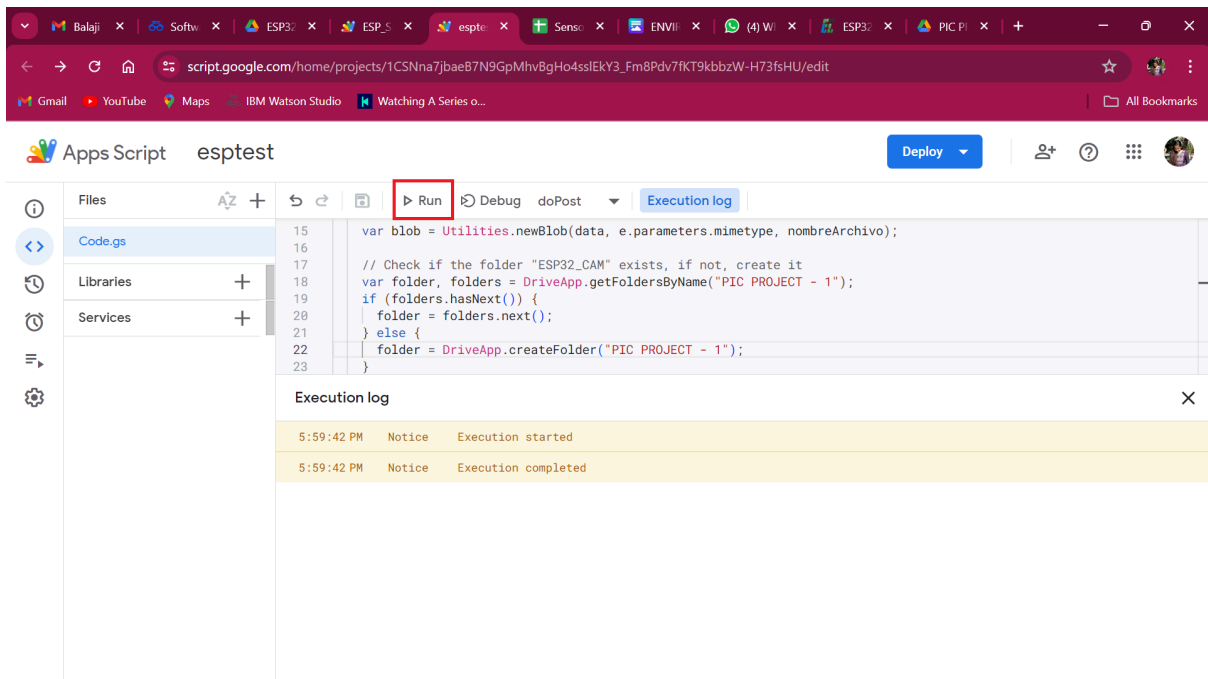
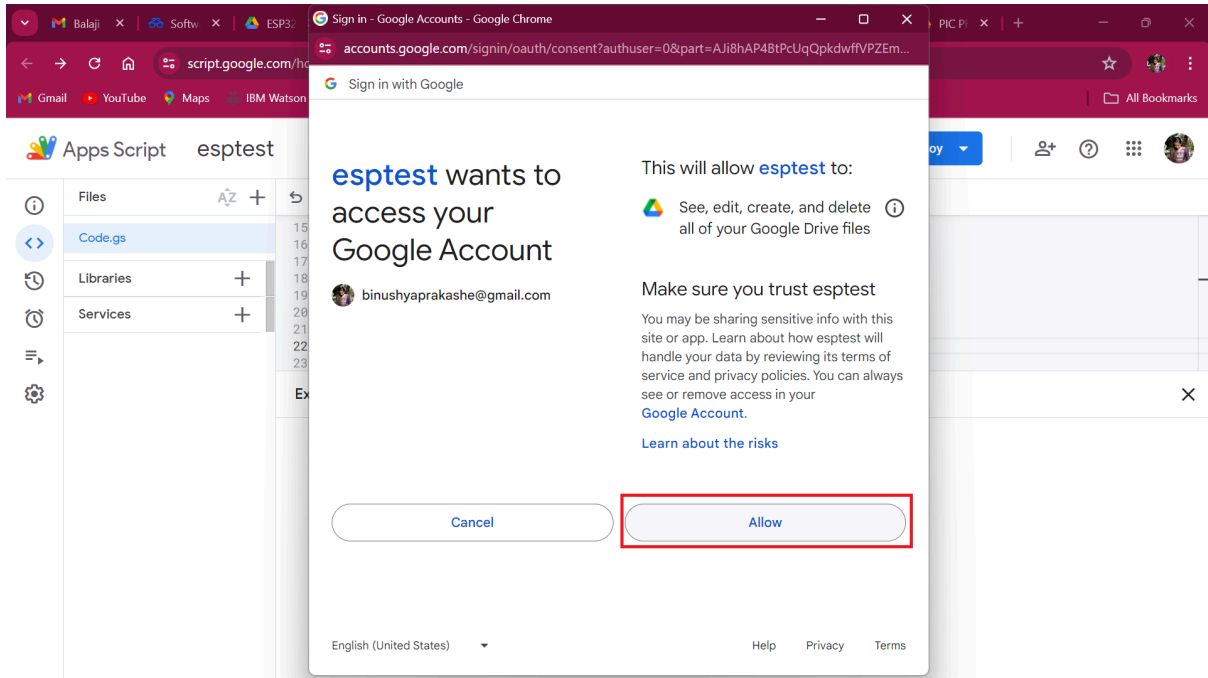
    }
};
Logger.log(doPost(e).getContent());
}

```

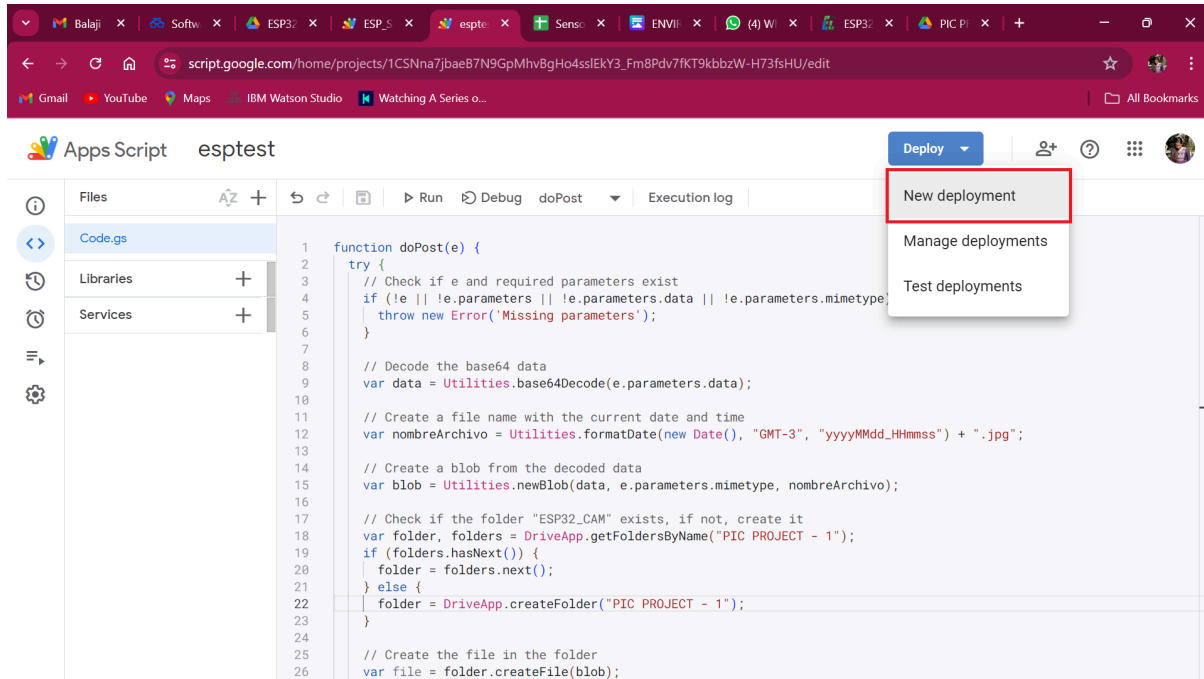
4. Save and Run the code. During the Execution process, provide the drive with the necessary permissions to access the cloud.



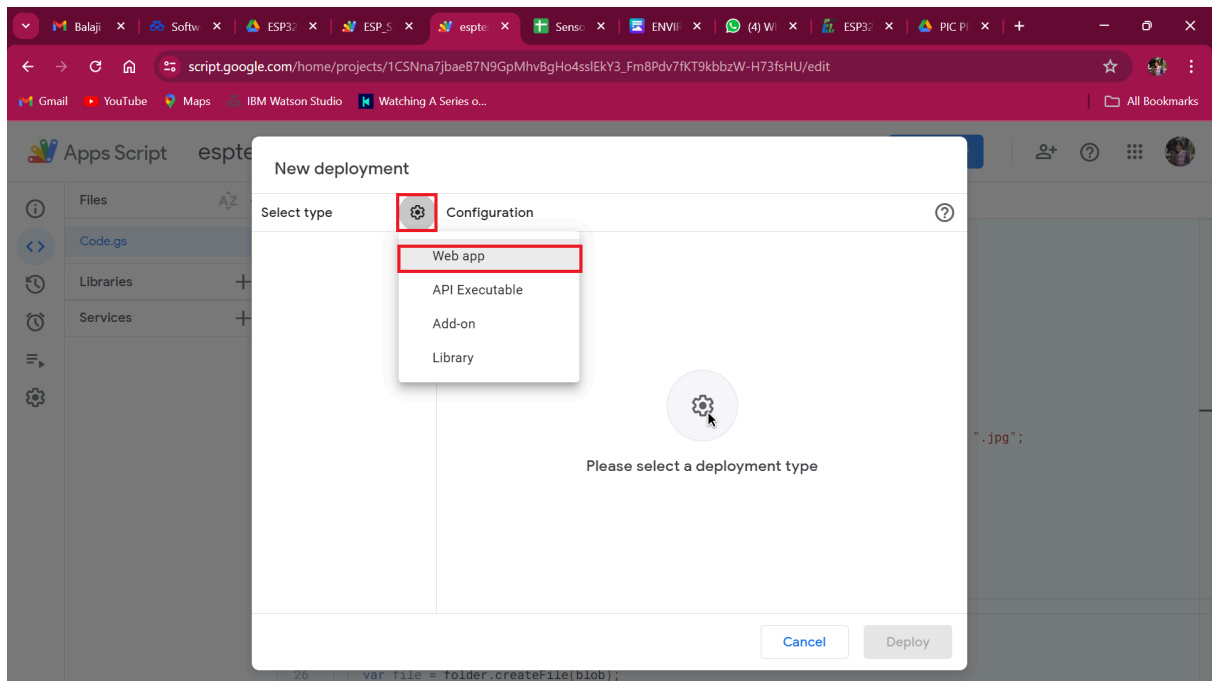




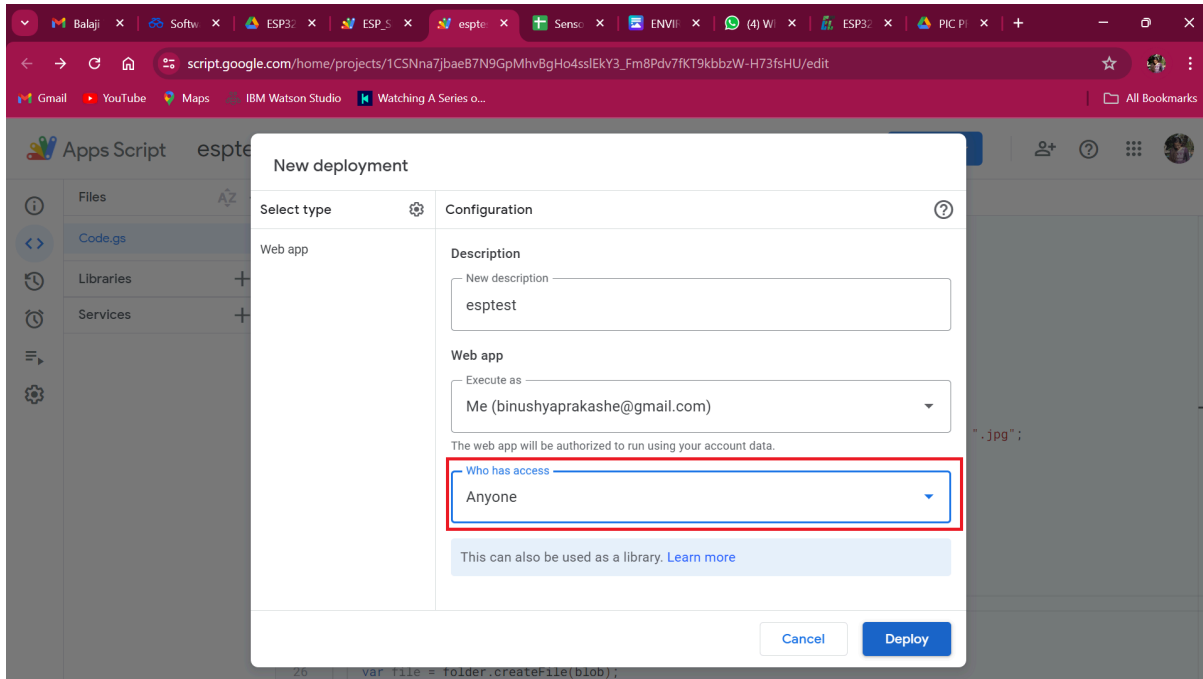
5. Deploy the following Code. Make a new deployment.



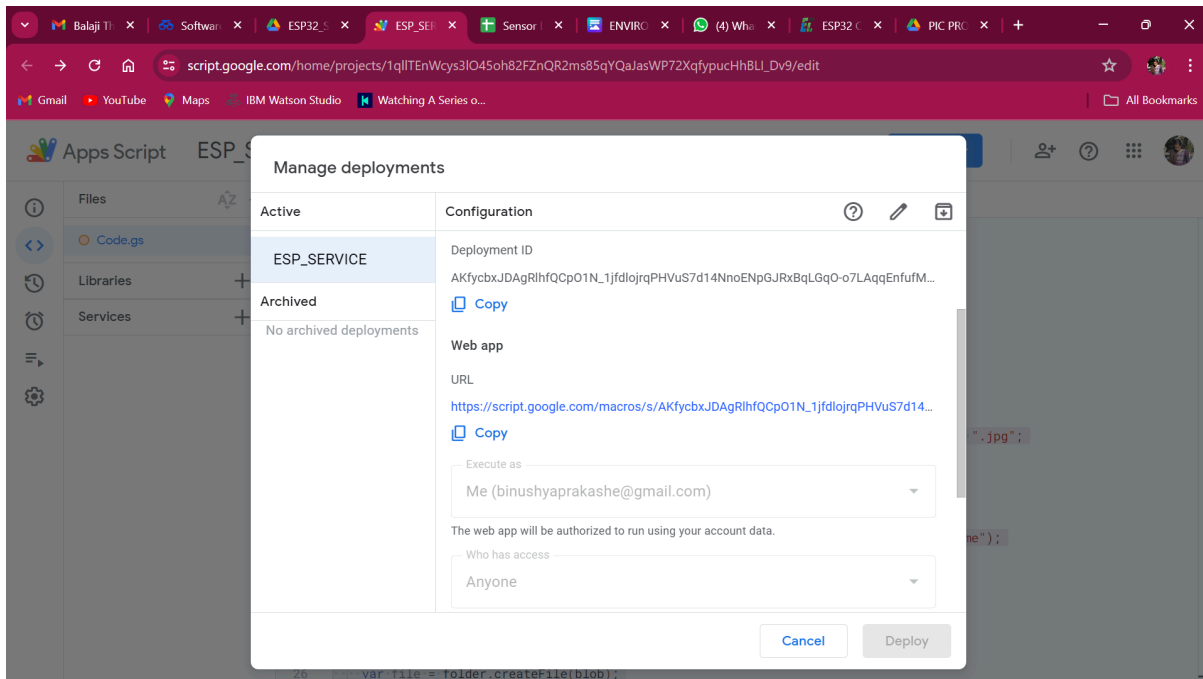
6. Now create a new deployment as a Web App.



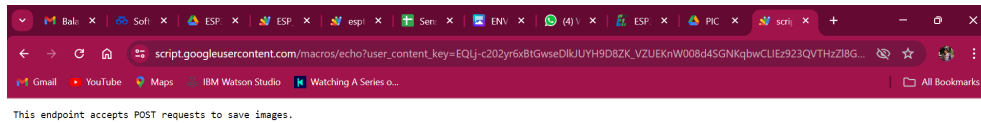
7. Modify the Changes as given and deploy.



8. Now Copy the Web app Script link and check for endpoint access.



9. If you get the text saying image ready to access the endpoint.



10. you're ready to go. Copy the script link and save it in the "ScriptDriveUrl.txt" file using SD Card.

Note: The WebApp URL generated in the above step(Step-8) needs to be stored in the SD Card.

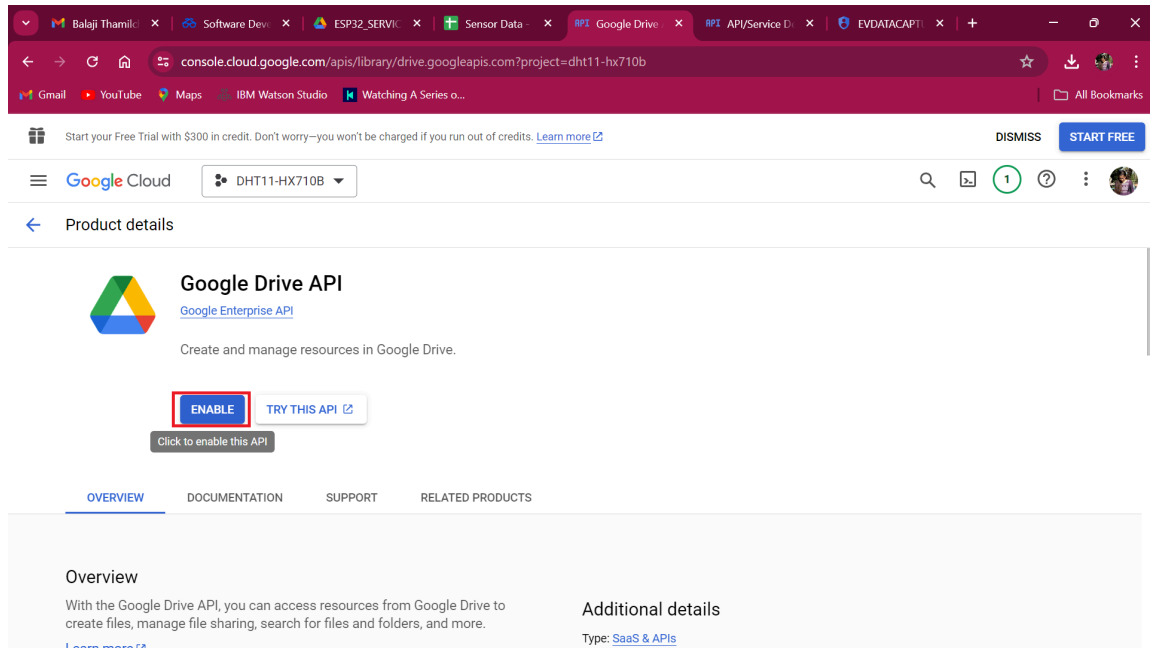
Step 2: Enable Google Drive API

1. Click the below link to navigate to Google Drive API

<https://console.cloud.google.com/apis/library/drive.googleapis.com>

* To enable Google Drive API, you have to use the same Google account.

2. Click **Enable** icon to enable Google Drive API



The screenshot shows the Google Cloud console interface for the Google Drive API. The browser tabs at the top include 'Balaji Thamil', 'Software Dev', 'ESP32_SERV', 'Sensor Data', 'RPI Google Drive', 'RPI API/Service D', and 'EVDATAAPT'. The address bar shows the URL 'console.cloud.google.com/apis/library/drive.googleapis.com?project=dht11-hx710b'. The page header includes a 'Start your Free Trial' message and a 'DISMISS' button. The main content area displays the 'Google Drive API' with a description: 'Create and manage resources in Google Drive.' Below this, the 'ENABLE' button is highlighted with a red box, and a 'TRY THIS API' link is also visible. A tooltip below the button says 'Click to enable this API'. The navigation tabs at the bottom include 'OVERVIEW', 'DOCUMENTATION', 'SUPPORT', and 'RELATED PRODUCTS'. The 'Overview' section is active, showing a description of the API and a link to 'Learn more?'. The 'Additional details' section shows the type as 'SaaS & APIs'.

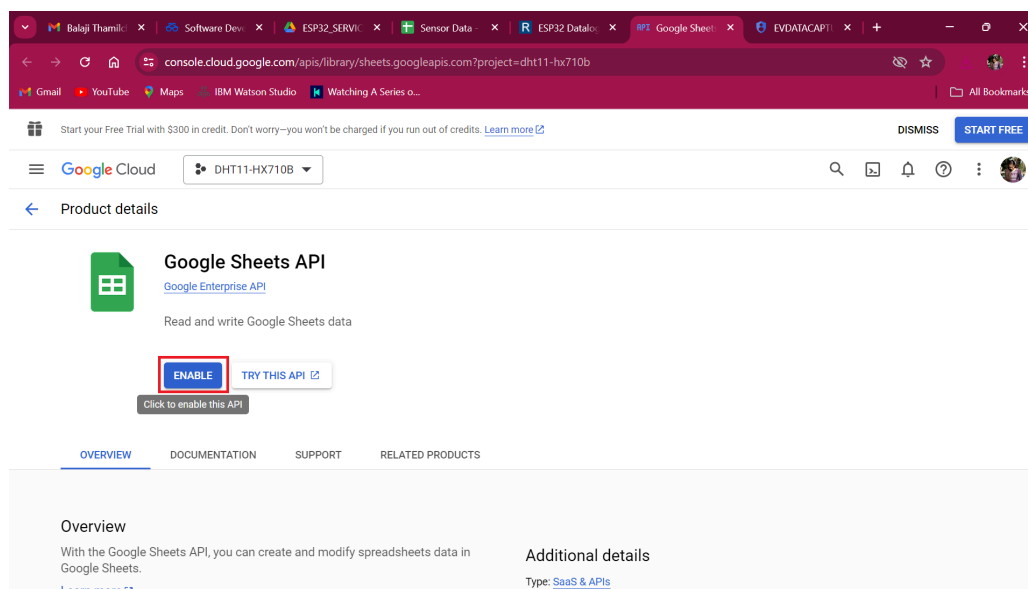
Step 3: Enable Google Sheet API

1. Click the below link to navigate to Google Sheet API

<https://console.cloud.google.com/apis/library/sheets.googleapis.com>

* To enable Google Sheets API, you have to use the same Google account.

2. Click **Enable** icon to enable Google Sheet API



The screenshot shows the Google Cloud console interface for the Google Sheets API. The browser tabs at the top include 'Balaji Thamil', 'Software Dev', 'ESP32_SERV', 'Sensor Data', 'R ESP32 Data', 'RPI Google Sheet', and 'EVDATAAPT'. The address bar shows the URL 'console.cloud.google.com/apis/library/sheets.googleapis.com?project=dht11-hx710b'. The page header includes a 'Start your Free Trial' message and a 'DISMISS' button. The main content area displays the 'Google Sheets API' with a description: 'Read and write Google Sheets data'. Below this, the 'ENABLE' button is highlighted with a red box, and a 'TRY THIS API' link is also visible. A tooltip below the button says 'Click to enable this API'. The navigation tabs at the bottom include 'OVERVIEW', 'DOCUMENTATION', 'SUPPORT', and 'RELATED PRODUCTS'. The 'Overview' section is active, showing a description of the API and a link to 'Learn more?'. The 'Additional details' section shows the type as 'SaaS & APIs'.

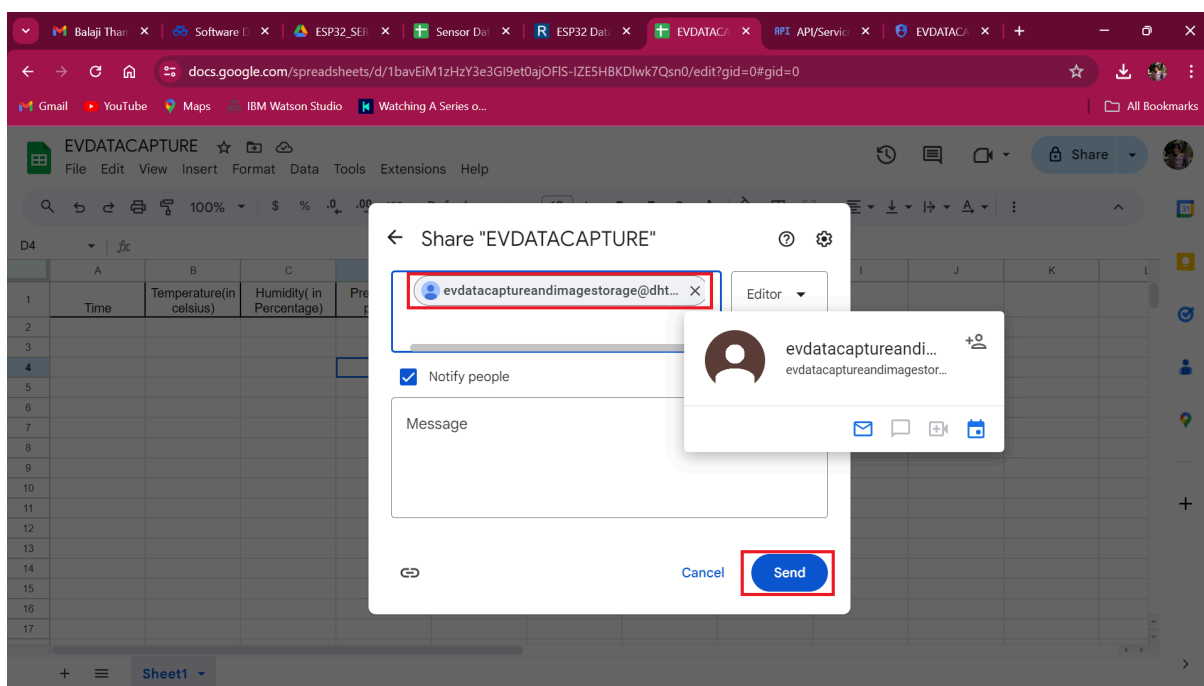
Step 4: Create a Google Spreadsheet

1. Go to Google Sheets and [create a new spreadsheet](#).
2. Assign any name to your spreadsheet.
3. In the first row of the spreadsheet, type the fields as shown below.

Timestamp	Temperature (in celsius)	Humidity (in percentage)	Pressure (in pascal)	Pressure (in ATM)	Pressure (in mmHg)	Pressure (in PSI)
-----------	--------------------------	--------------------------	----------------------	-------------------	--------------------	-------------------

4. Share the Spreadsheet with the service account email(client_email). Give access to the following client email id.

test-environmentdatacapturing@test-environmentdatacapturing.iam.gserviceaccount.com



5. Get the Spreadsheet Link and paste the following in "spreadsheetUrl.txt" in the SD Card.

