**GOOGLE APP SCRIPT**

**What is google app script?**

Google Apps Script is a rapid application development platform that makes it fast and easy to create business applications that integrate with Google Workspace.You can access google product like sheets,drive etc.. and nothing need to install google gives code editor you write on here and you can run on server

**What you can do with app script?**

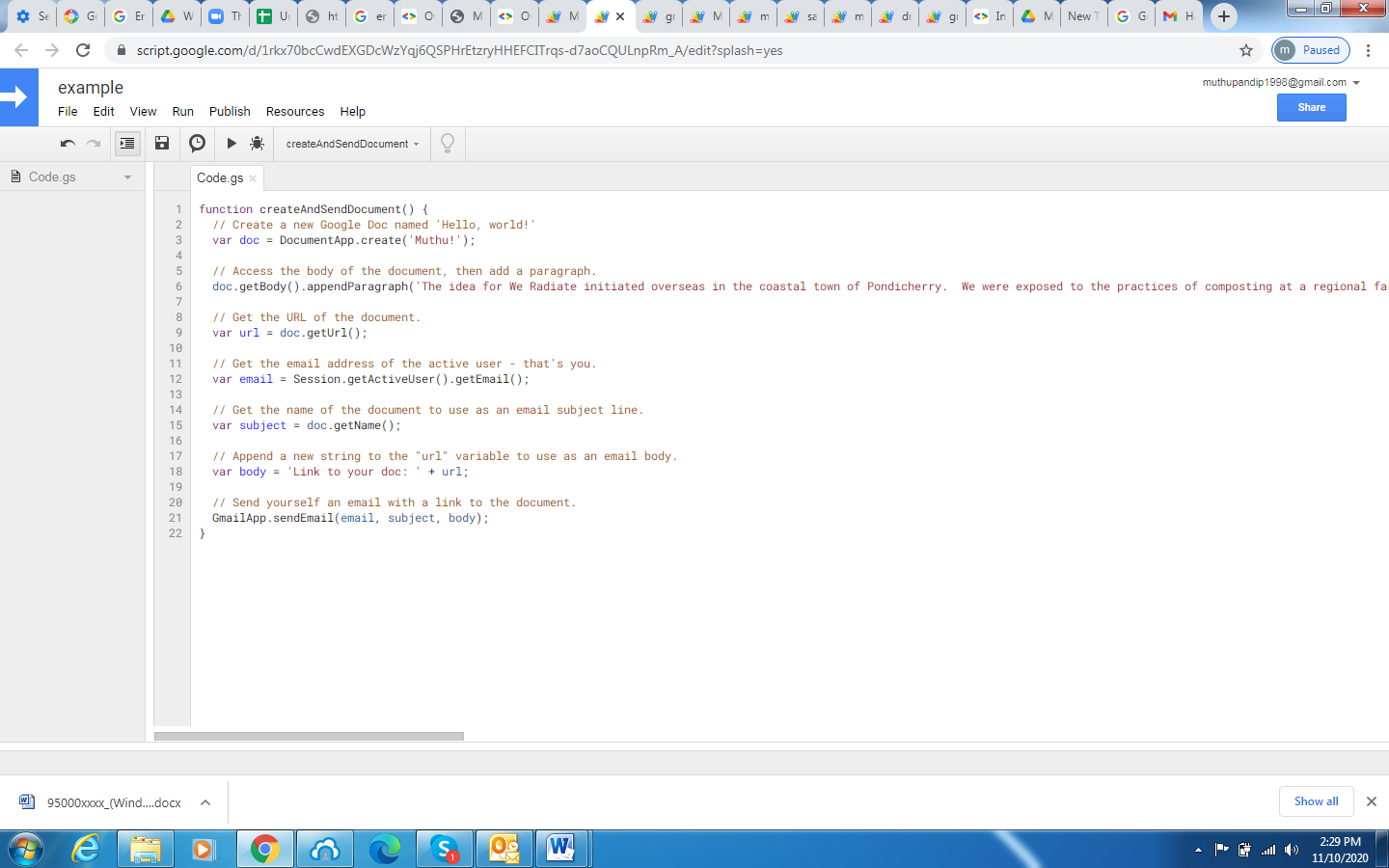
* Add [custom menus](https://developers.google.com/apps-script/guides/menus), [dialogs, and sidebars](https://developers.google.com/apps-script/guides/dialogs) to Google Docs, Sheets, and Forms.
* Write [custom functions](https://developers.google.com/apps-script/execution_custom_functions) and [macros](https://developers.google.com/apps-script/guides/sheets/macros) for Google Sheets.

**Set up:**

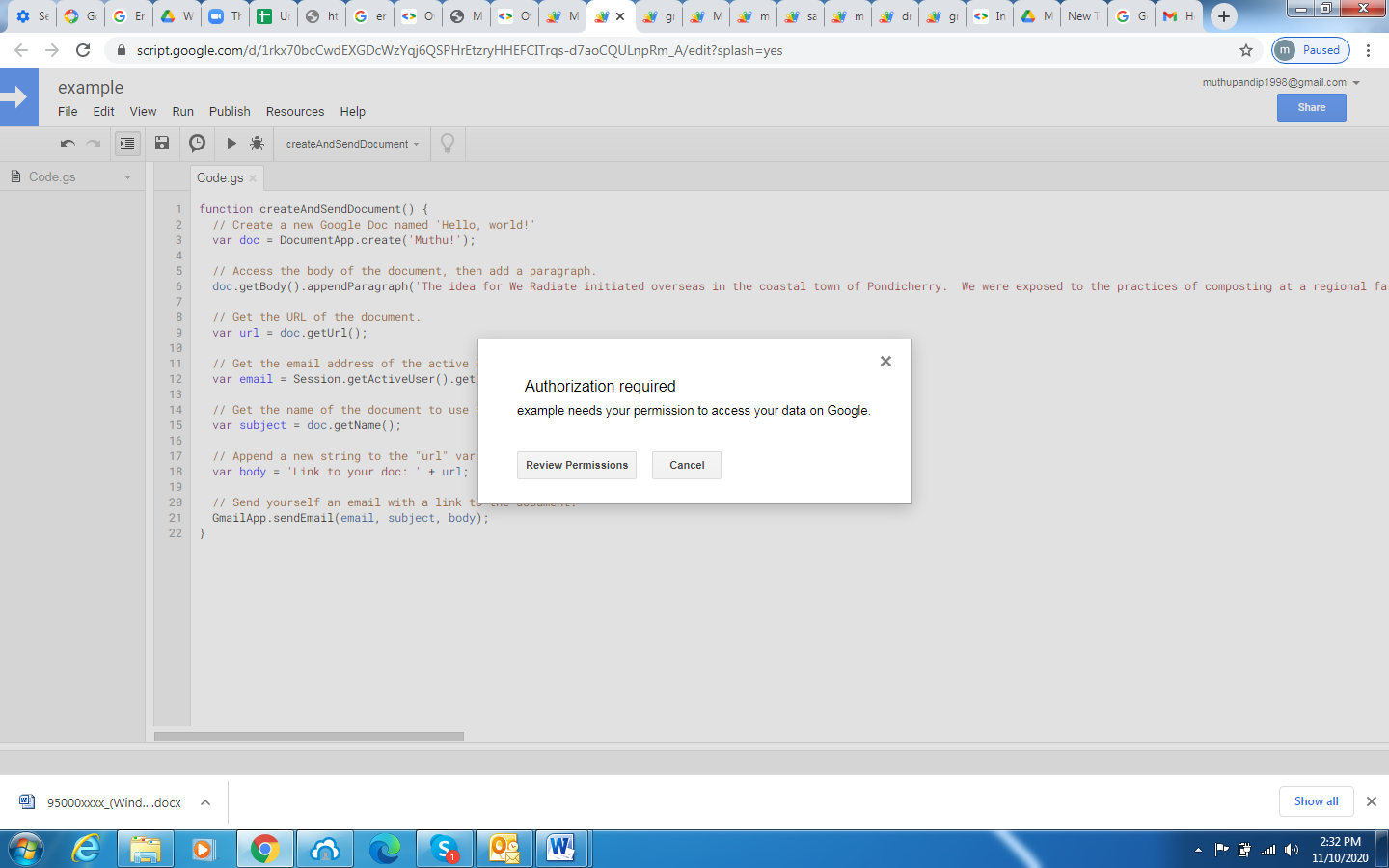
Visit [script.google.com](https://script.google.com/?authuser=2) to open the script editor.Here you create new project and code write code on here

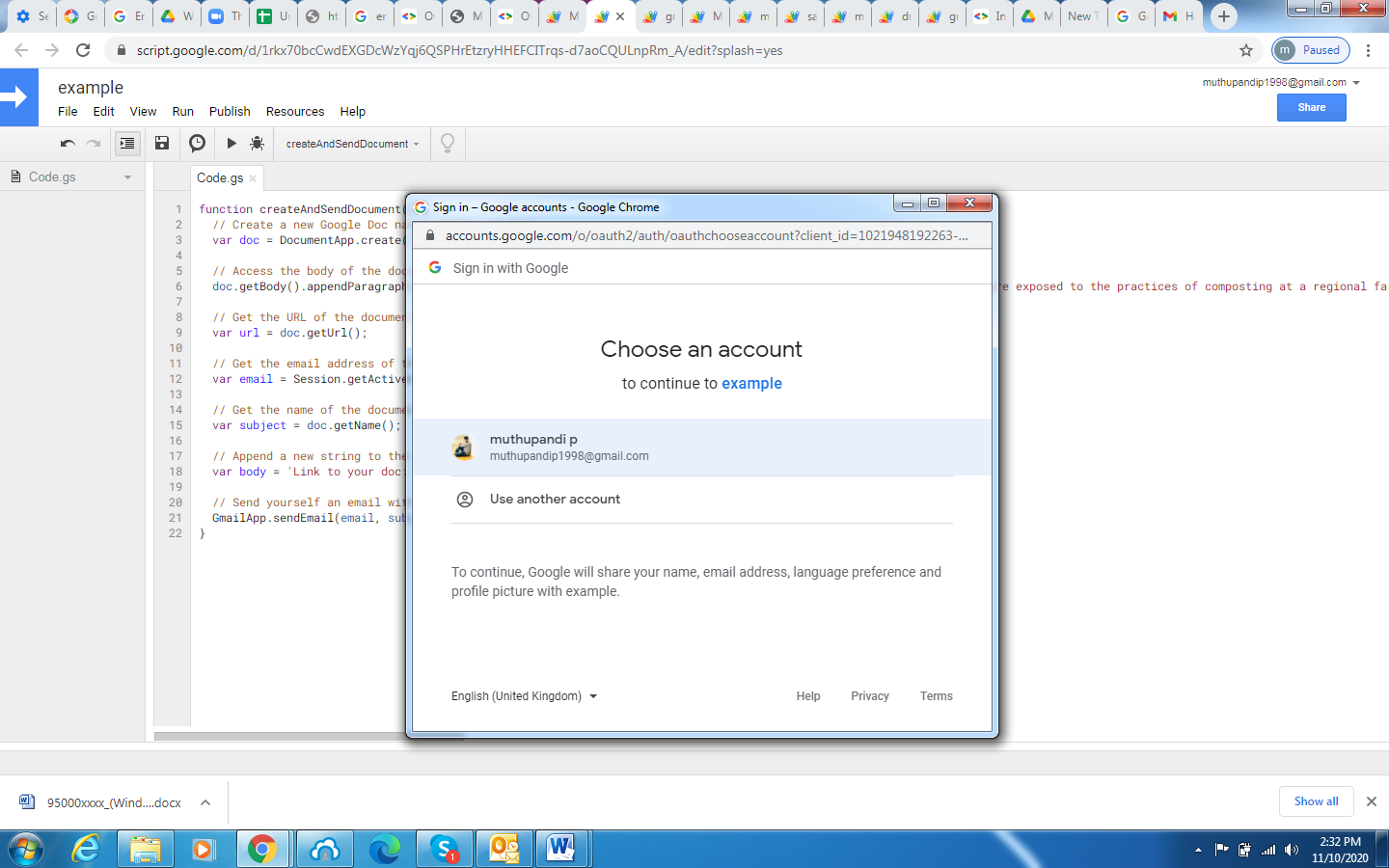
1)Create one document in drive then send mail that document using google app script

\* Write code on code editor file then save that file

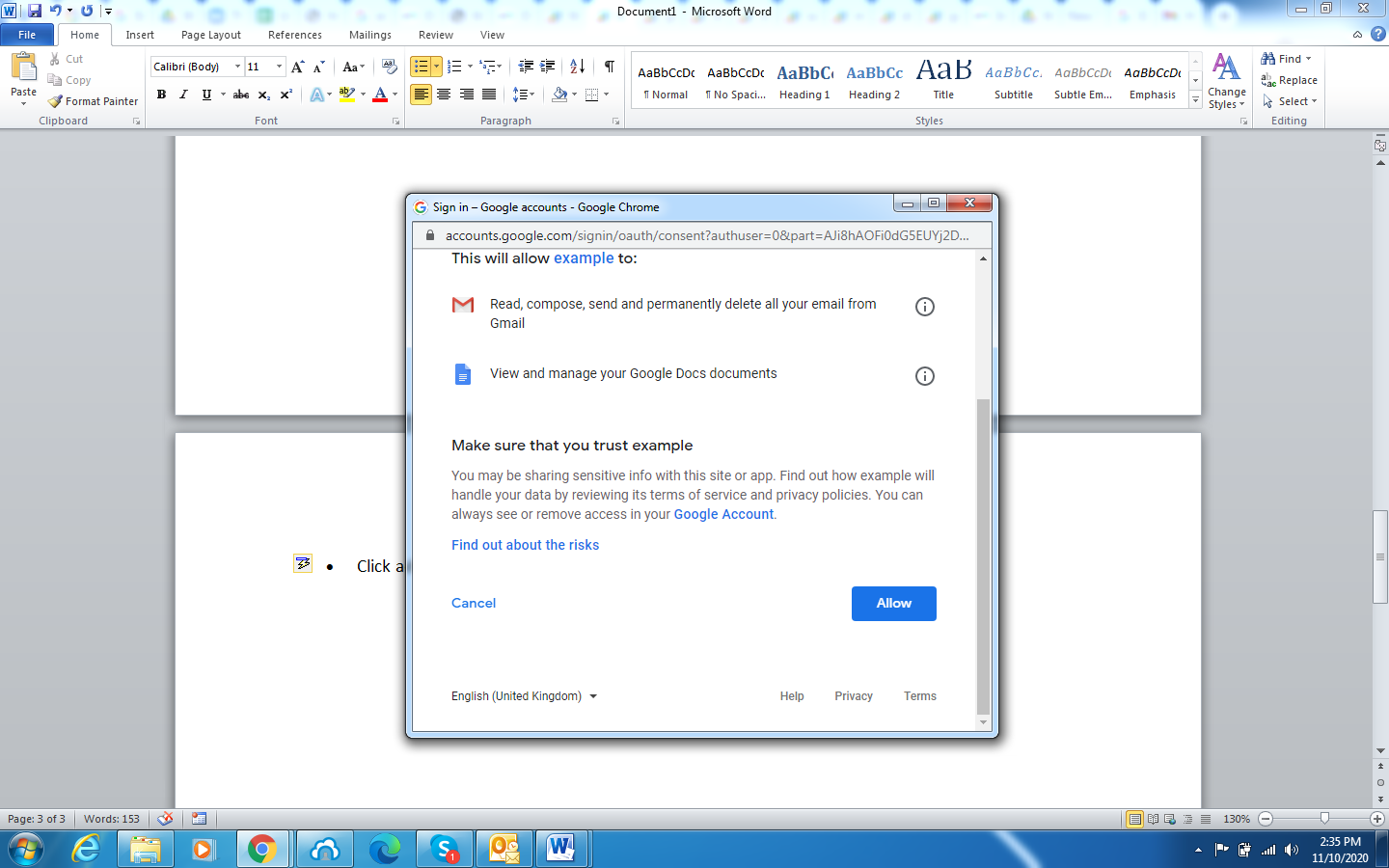


* Then click top run button it ask sign in your mail id then gives permission allow your gmail

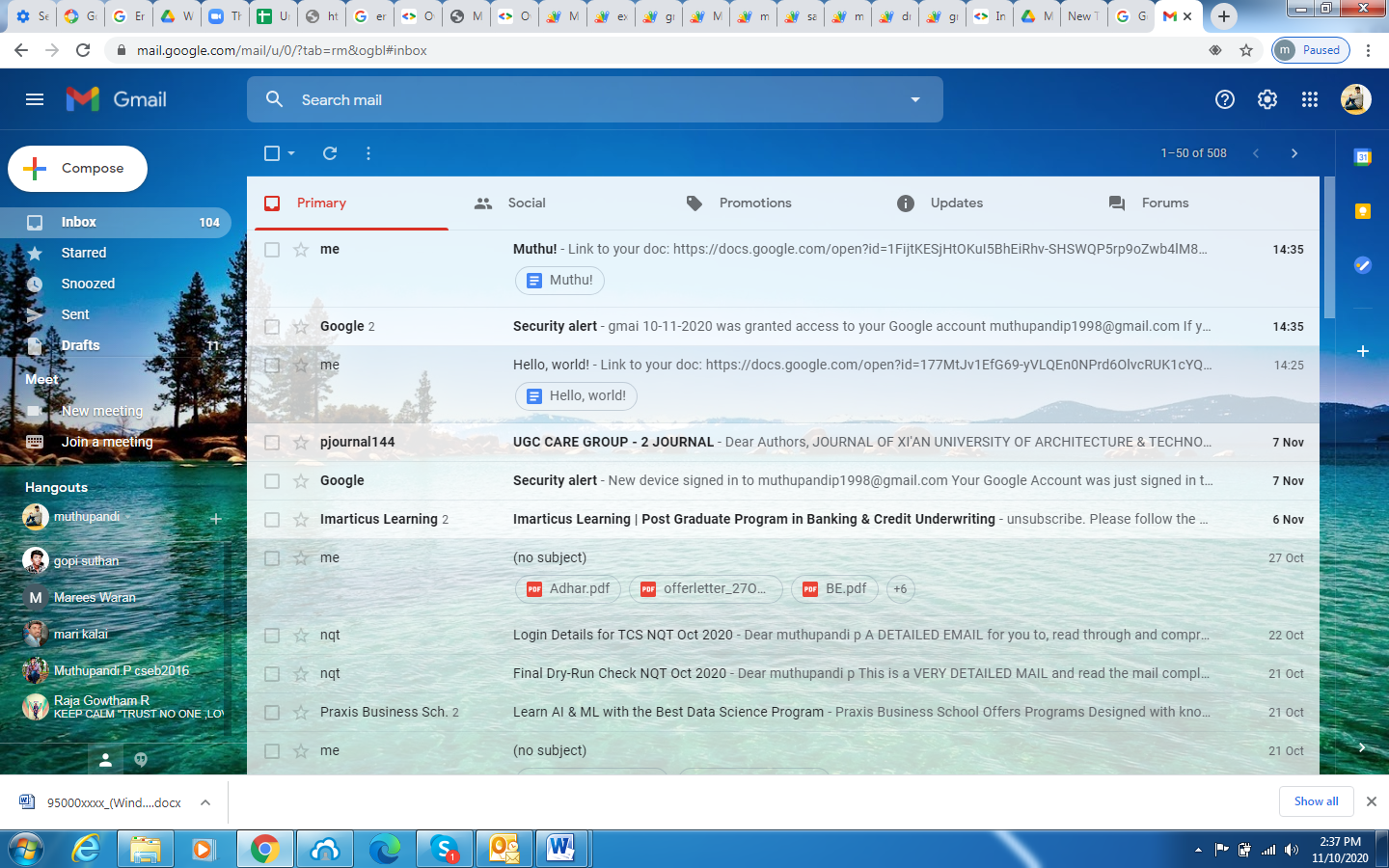




* Click advance then goes to example and click allow



* Now check your mail you received one mail



**Code:**

function createAndSendDocument() {

// Create a new Google Doc named 'Hello, world!'

var doc = DocumentApp.create('Hello, world!');

// Access the body of the document, then add a paragraph.

doc.getBody().appendParagraph('This document was created by Google Apps Script.');

// Get the URL of the document.

var url = doc.getUrl();

// Get the email address of the active user - that's you.

var email = Session.getActiveUser().getEmail();

// Get the name of the document to use as an email subject line.

var subject = doc.getName();

// Append a new string to the "url" variable to use as an email body.

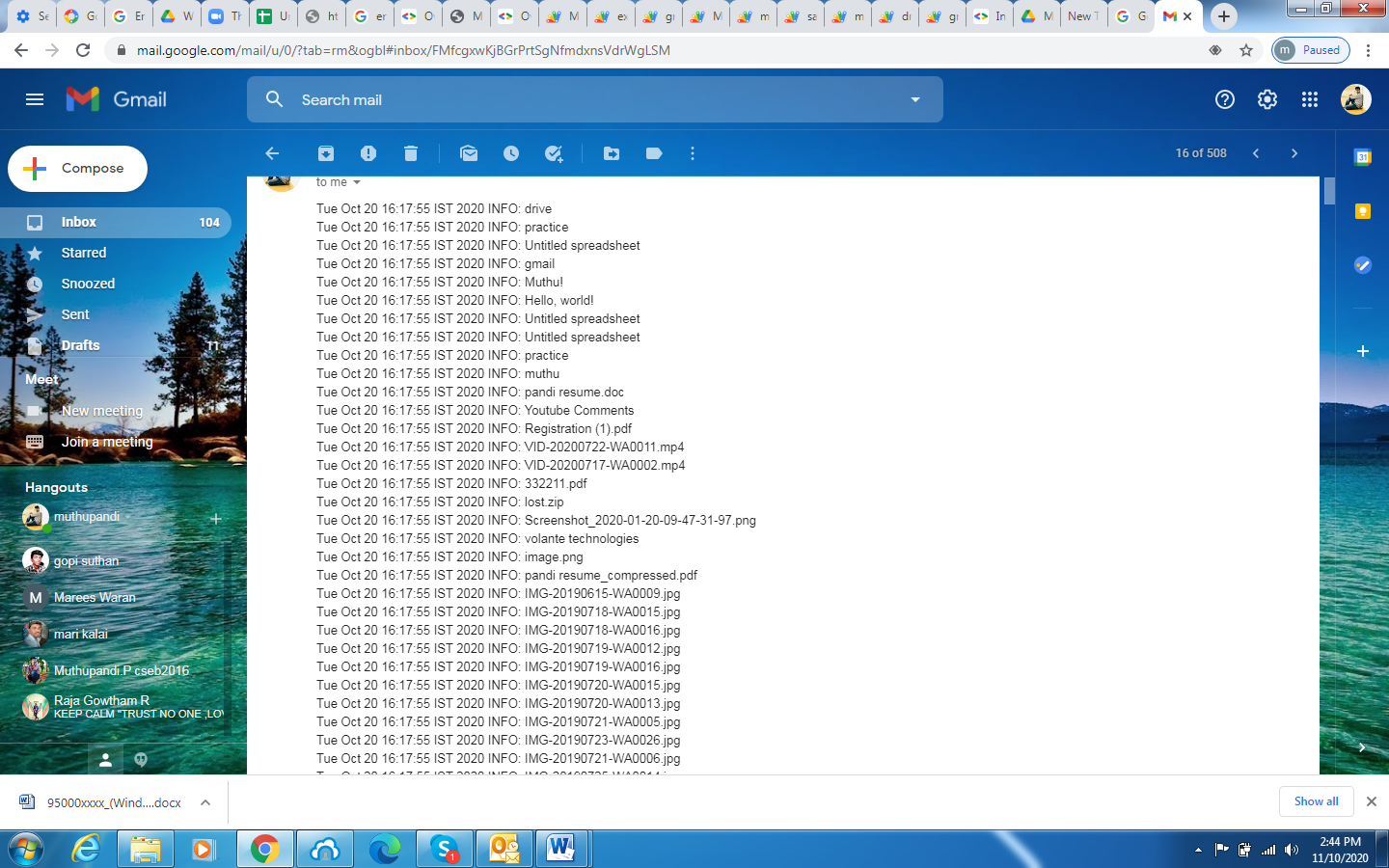
var body = 'Link to your doc: ' + url;

// Send yourself an email with a link to the document.

GmailApp.sendEmail(email, subject, body);

}

2) Get your drive list and send to your gmail same process preceed here and check your mail



Code:

function listFiles() {

var files = DriveApp.getFiles();

while (files.hasNext()) {

Logger.log(files.next().getName());

}

var recipient = 'muthupandip541@gmail.com';

var subject = 'A list of files in your Google Drive';

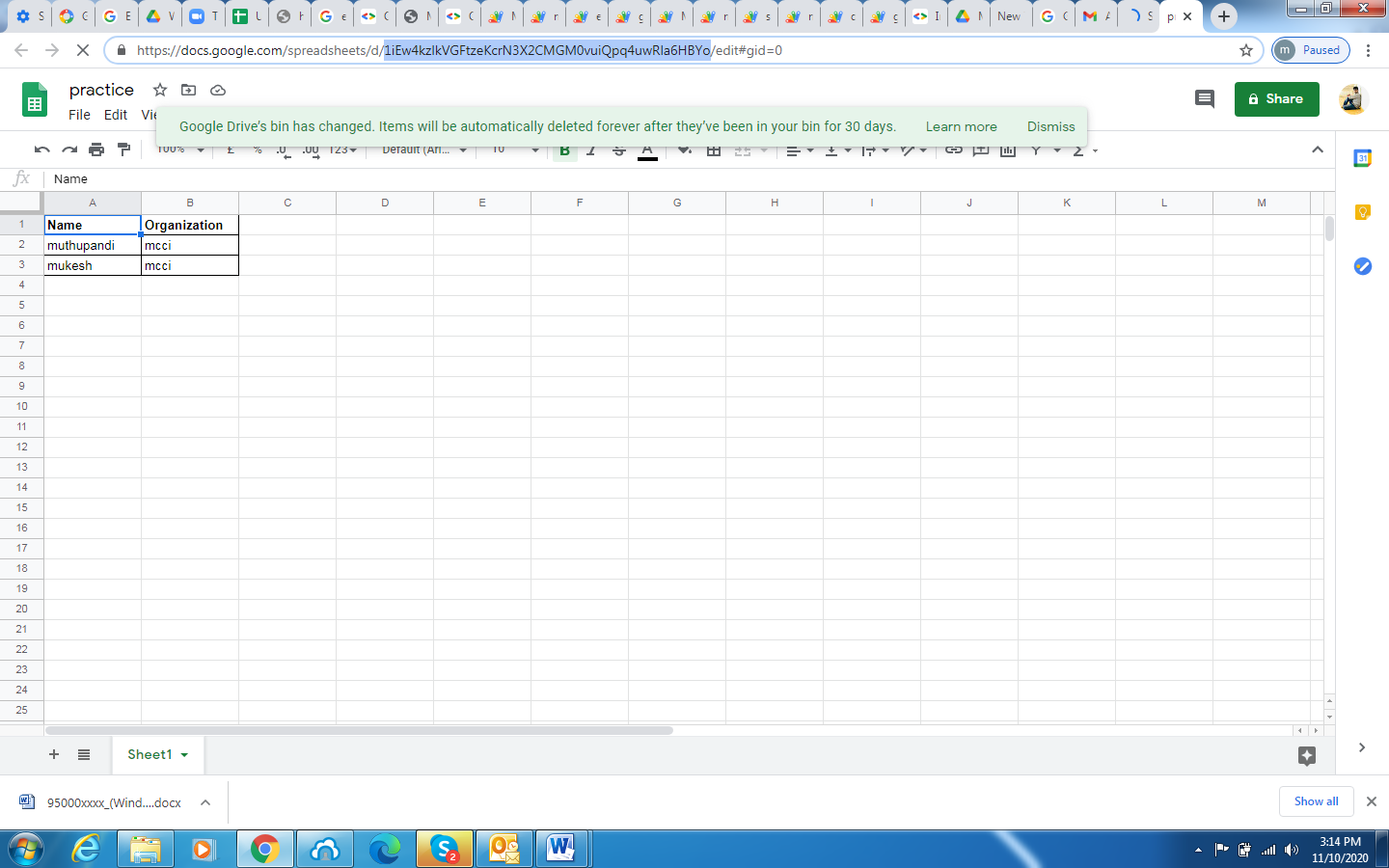
var body = Logger.getLog();

MailApp.sendEmail(recipient, subject, body);

}

3) You can access the sheet data from drive

\* copy spread sheet id from google drive



Code:

function logNamesAndMajors() {

var spreadsheetId = '1BxiMVs0XRA5nFMdKvBdBZjgmUUqptlbs74OgvE2upms';

var rangeName = 'Class Data!A2:E';

var values = Sheets.Spreadsheets.Values.get(spreadsheetId, rangeName).values;

if (!values) {

Logger.log('No data found.');

} else {

Logger.log('Name, Major:');

for (var row = 0; row < values.length; row++) {

// Print columns A and E, which correspond to indices 0 and 4.

Logger.log(' - %s, %s', values[row][0], values[row][4]);

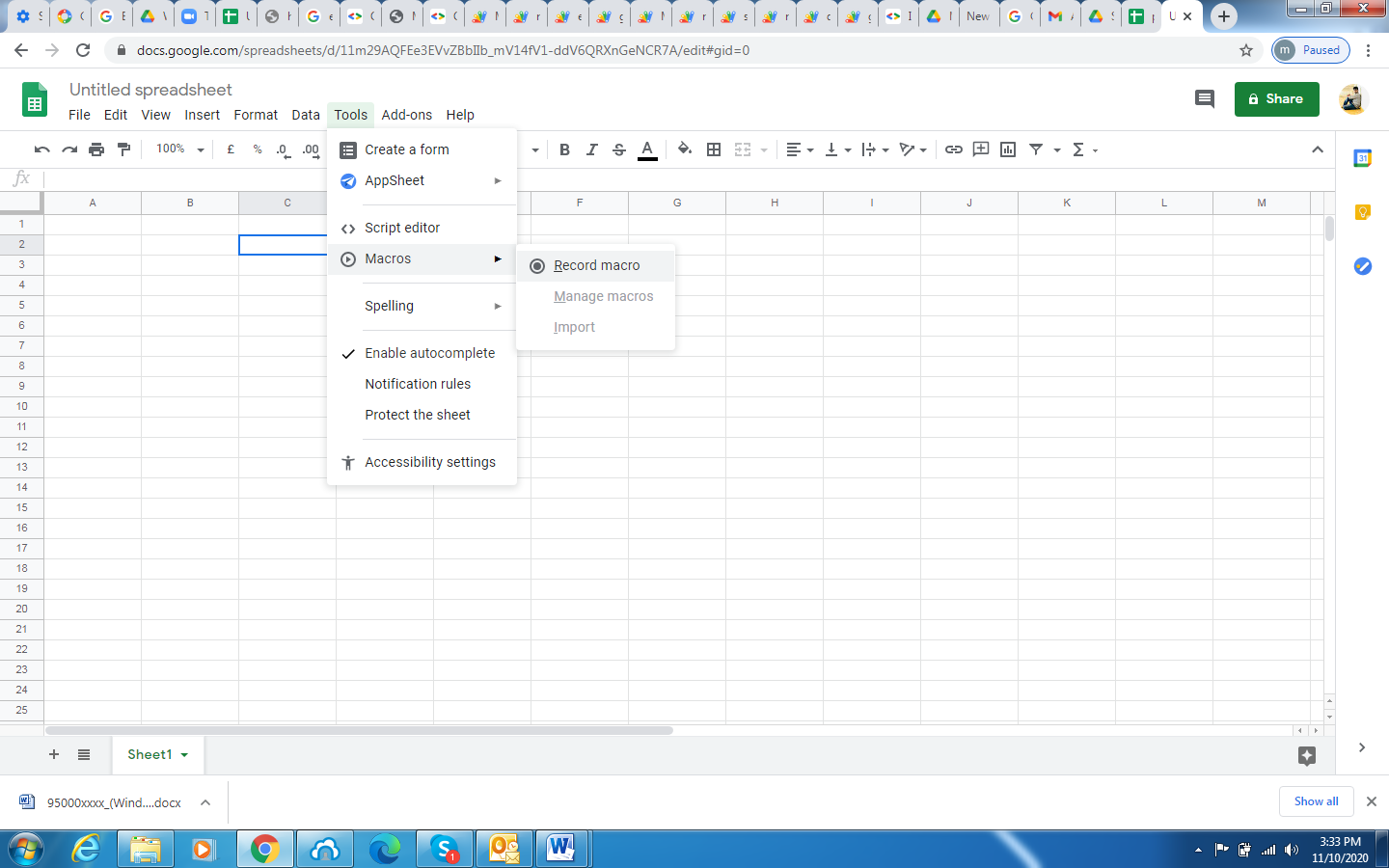
}

}

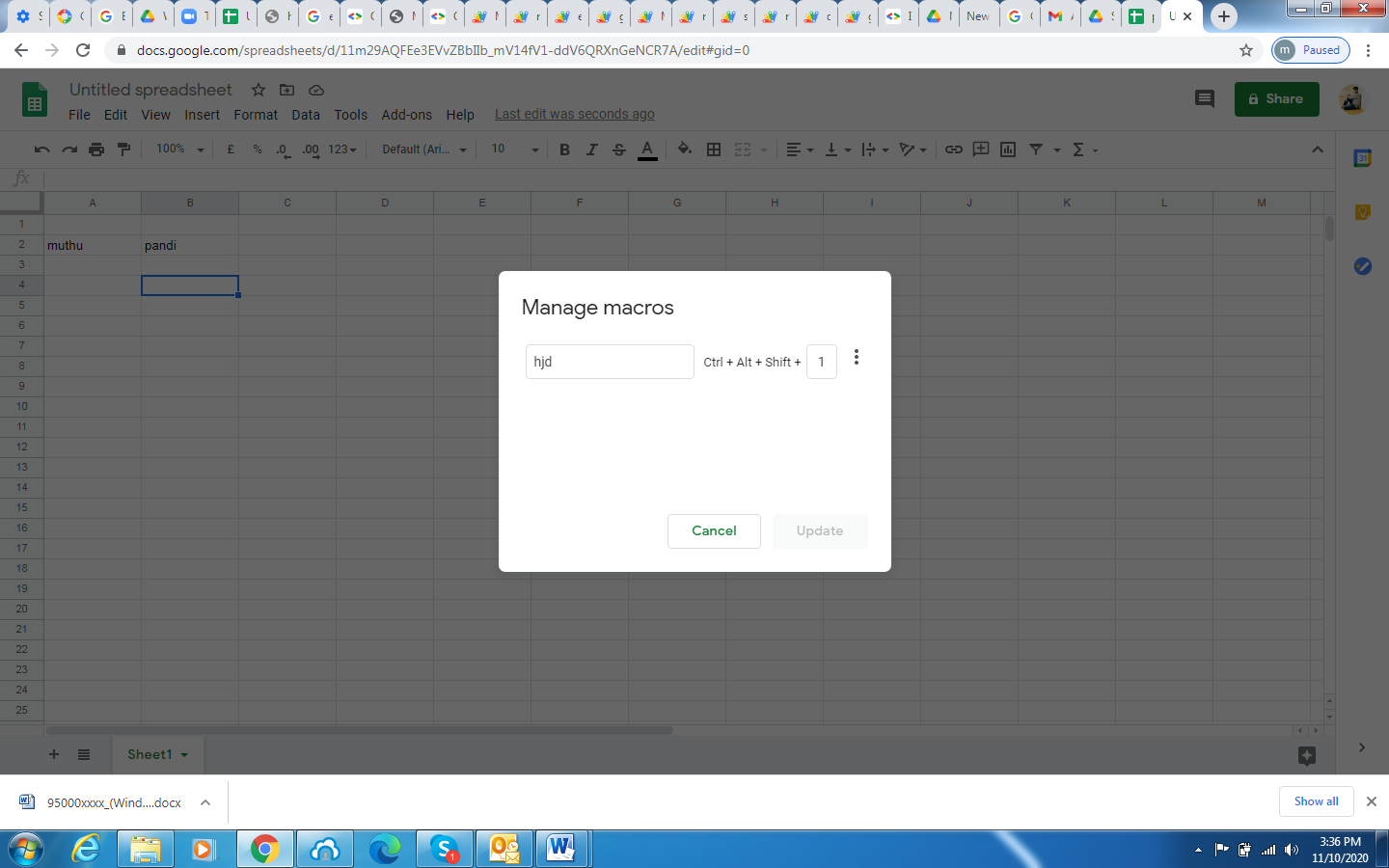
}

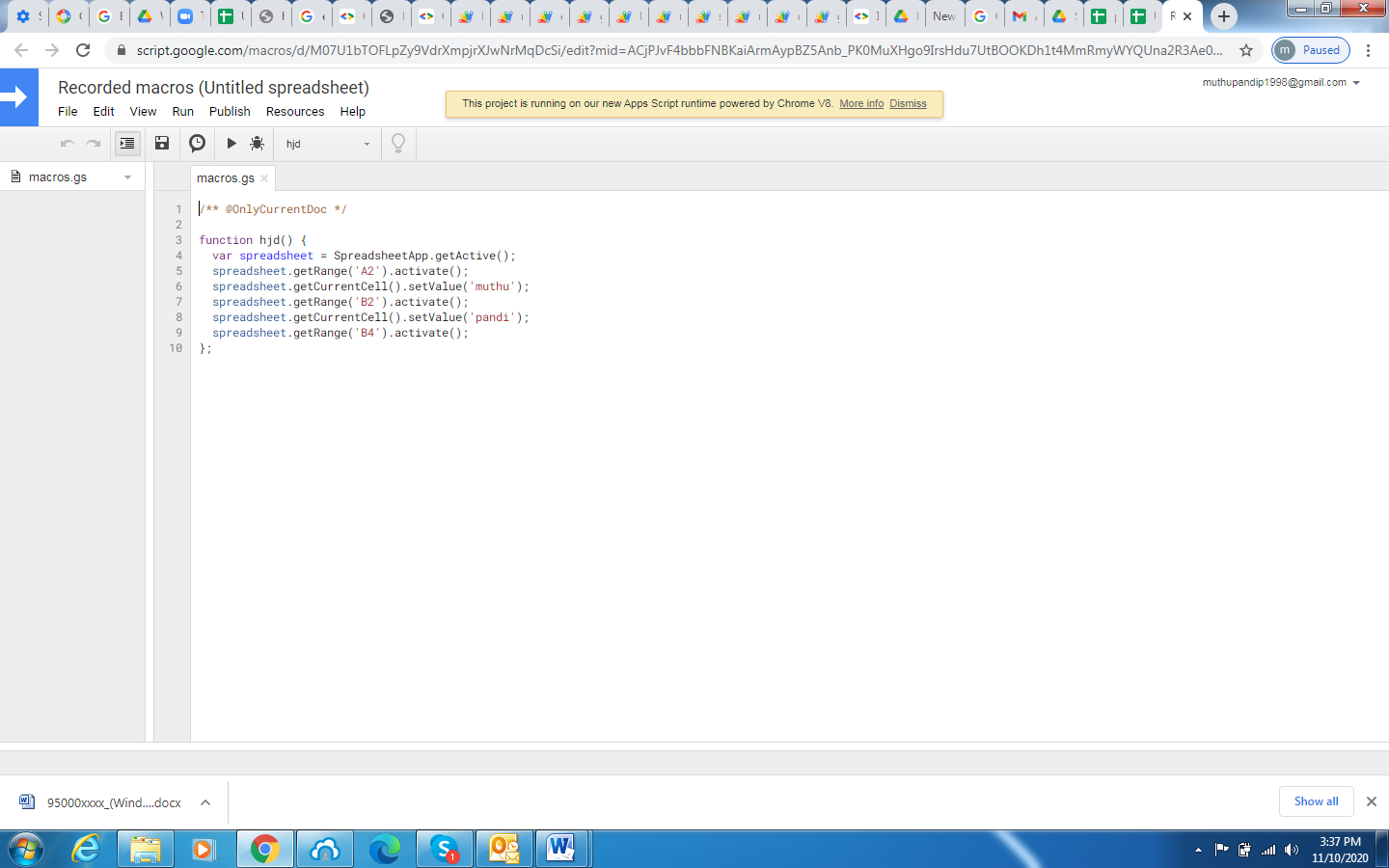
Google sheet record macro:

* 1. Open google sheet click on tools and click record macro



* 1. Then you work something in google sheet and next again click tools then click manage macro click and that 3dots then click edit





**Google sheet api:**

The Google Sheets API lets you read and modify any aspect of a spreadsheet .You can read ,write and modify in spread shhet

### Spreadsheet ID:

Every API method requires a spreadsheetId parameter which is used to identify which spreadsheet is to be accessed or altered. This ID is the value between the "/d/" and the "/edit" in the URL of your spreadsheet.

[https://docs.google.com/spreadsheets/d/***spreadsheetId***/edit#gid=0](https://docs.google.com/spreadsheets/d/spreadsheetId/edit#gid=0)

### A1 notation:

Some API methods require a range in *A1 notation*. This is a string like Sheet1!A1:B2, that refers to a group of cells in the spreadsheet, and is typically used in formulas. For example, valid ranges are:

# Create a Spreadsheet:

var sheet =Sheets.newSpreadsheet();  
sheet.properties = Sheets.newSpreadsheetProperties();  
sheet.properties.title = title;  
var spreadsheet = Sheets.Spreadsheets.create(sheet);

# Reading & Writing Cell Values:

# Read:

var result = Sheets.Spreadsheets.Values.get(spreadsheetId, range);  
var numRows = result.values ? result.values.length : 0;

# Write:

var values = [];  
var valueRange = Sheets.newValueRange();  
valueRange.values = values;  
var result = Sheets.Spreadsheets.Values.update(valueRange, spreadsheetId, range, {  
  valueInputOption: valueInputOption  
});