**Format Form**

const createTrigger = () => {

  ScriptApp.getProjectTriggers().forEach((trigger) => {

    ScriptApp.deleteTrigger(trigger);

  });

ScriptApp.newTrigger('formRowonFormSubmit').forSpreadsheet(Spreadsheet.App.getActive()).onFormSubmit().create();

};

const formRowOnFormSubmit = (e) => {

  if (!e) {

    throw new Error('Please do not run this function manually!');

  }

  const { range } = e;

  const row = range.getRowIndex();

  if (row > 2) {

    const sheet = range.getSheet();

    // Select the previous row range

    const sourceRange = sheet.getRange(`${row - 1}:${row - 1}`);

    // Get the recently added row

    const targetRange = sheet.getRange(`${row}:${row}`);

    // Copy the format only from the previous row to the current row

    sourceRange.copyTo(targetRange, SpreadsheetApp.CopyPasteType.PASTE\_FORMAT, false);

  }

};

**Combined Script (Date + Emails)**

function onEdit(e) {

  var sheet = e.source.getActiveSheet();

  var editedRange = e.range;

  // Check if the edited range is in column N or R

  if (editedRange.getColumn() == 14 && editedRange.getValue() !== "") { // Column N change it if needed

    var timestamp = new Date();

    sheet.getRange(editedRange.getRow(), editedRange.getColumn() + 1).setValue(timestamp);

  }

  if (editedRange.getColumn() == 18 && editedRange.getValue() !== "") { // Column R change it if needed

    var timestamp = new Date();

    sheet.getRange(editedRange.getRow(), editedRange.getColumn() + 1).setValue(timestamp);

    email(e);

    sendemail(e);

  }

}

function email(e) {

  var row = e.range.getRow();

  var value = e.value;

  var sheet = e.range.getSheet();

  switch (value) {

    case "TRUE":

      var to = sheet.getRange(`B${row}`).getValue();

      var subject = sheet.getRange(`C${row}`).getValue();  // order in column C, adjust if necessary

      var emailBody = `Hello, Customer Services,<br><br>${sheet.getRange(`N${row}`).getValue()}<br><br>Regards<br><br>NA Logistics Support<br>`; // if you want to change the "Customer Services" feel free to change it but please don't delete anything after the comma

      MailApp.sendEmail({ to, subject, htmlBody: emailBody });

      break;

    default:

      // Do nothing

      break;

  }

}

function sendemail(e) {

  var row = e.range.getRow();

  var col = e.range.getColumn();

  var sheet = e.range.getSheet();

  // Checks if something was entered on column R

  if (col === 18) {

    //

    var value = e.value;

    // If the value in column R is not empty, it will send the email

    if (value !== "") {

      var to = sheet.getRange(`P${row}`).getValue(); // Email recipient from column P it uses data validation and it can't be a name , it needs to be an email address

      var subject = sheet.getRange(`C${row}`).getValue(); // It uses the order number from column C as a subject

      var name = sheet.getRange(`P${row}`).getValue();

      var emailBody = `Hello, ${name},<br><br>${sheet.getRange(`R${row}`).getValue()}<br><br>Regards<br><br>NA Logistics Support<br>`;

      MailApp.sendEmail({ to, subject, htmlBody: emailBody });

    }

  }

}

**Archiving**

function moveOldRows() {

  var ss = SpreadsheetApp.getActiveSpreadsheet();

  var sourceSheet = ss.getSheetByName('Form responses 1');

  var targetSheet = ss.getSheetByName('Archive (Date resolved +1 Day)');

  var currentDate = new Date();

  var oneWeekAgo = new Date();

  oneWeekAgo.setDate(oneWeekAgo.getDate() - 1); // is 1 day old (change the number if needed)

  var sourceData = sourceSheet.getDataRange().getValues();

  var rowsToMove = [];

  var rowsToDelete = [];

  for (var i = 1; i < sourceData.length; i++) { // Start from 1 to skip header row

    var dateValue = sourceData[i][14]; //  N is the 14th column (index 13)

    var checkboxValue = sourceData[i][20]; //  T is the 20th column (index 19)

    if (dateValue instanceof Date && dateValue < oneWeekAgo && checkboxValue === true) {

      rowsToMove.push(sourceData[i]);

      rowsToDelete.push(i + 1); //  1 for header row

    }

  }

  if (rowsToMove.length > 0) {

    targetSheet.getRange(targetSheet.getLastRow() + 1, 1, rowsToMove.length, sourceData[0].length).setValues(rowsToMove);

    rowsToDelete.reverse().forEach(function (rowIndex) {

      sourceSheet.deleteRow(rowIndex);

    });

  }

}