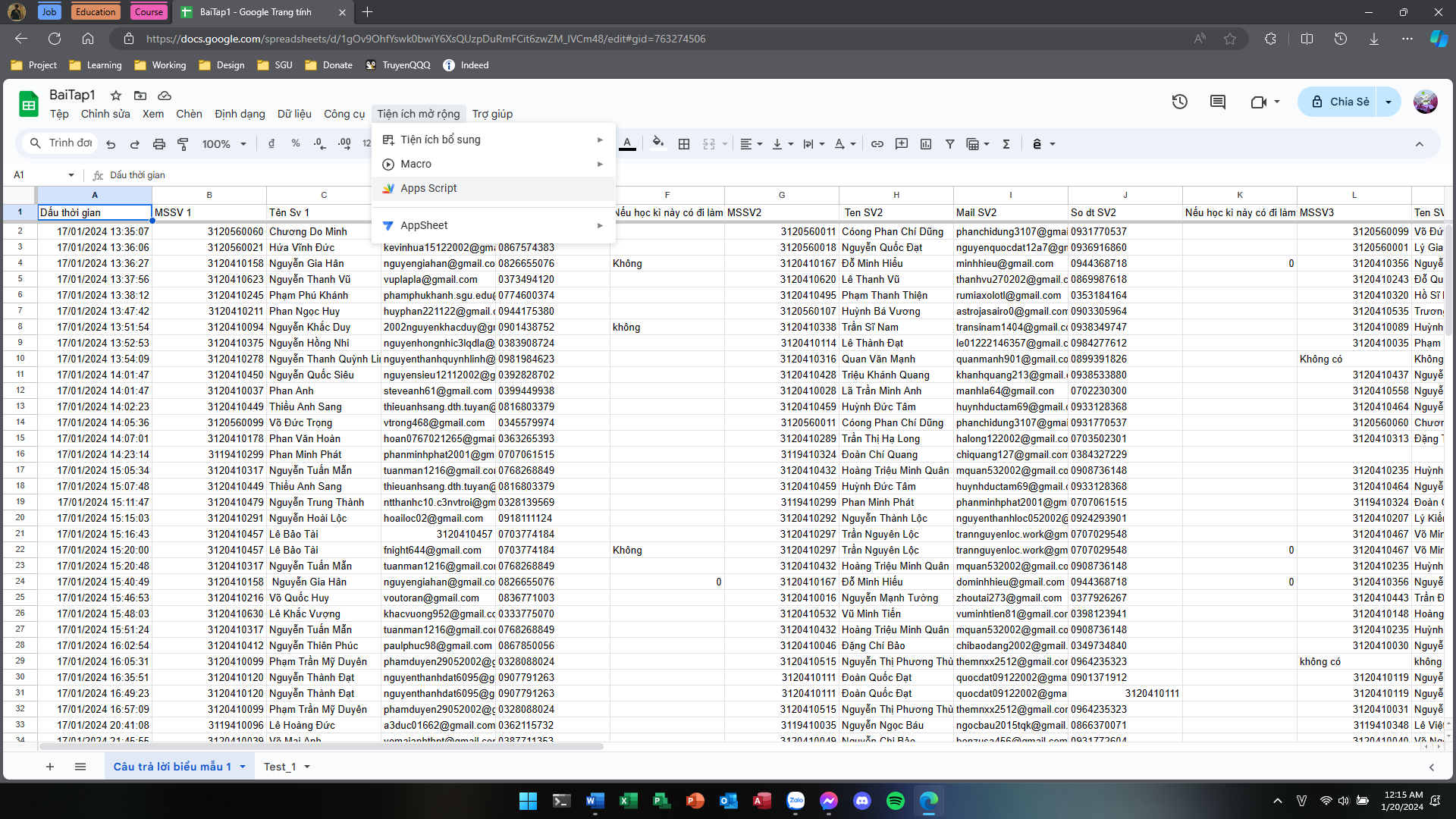
Link thư mục chứa các file demo:

<https://drive.google.com/drive/folders/1zxDcG6lOsUuTg8cWhGiIn0ZLTf1Nmlrh?usp=sharing>

1. **Chuyển file đăng ký từ 1 google sheet sang 1 google sheet khác với cách trình bày khác**

Để xử lý câu 1, em sử dụng tính năng **App Script** trên sheets:



Hình 1. Sử dụng App Script

Yêu cầu của đề bài là chuyển một file Sheets dạng như trên Hình 1 sang 1 dạng khác, cụ thể hơn là tách 1 dòng ra thành các nội dung một cách gọn gàng hơn. Vì 1 dòng tượng trưng cho 1 nhóm gồm nhiều thành viên và nhiều thông tin khác nhau. Do đó, ta sẽ viết Script để tách nhóm nội dung đó ra cho dễ đọc:



Hình 2. Giao diện mẫu

Để xử lý vấn đề đó, ta có đoạn script như sau:

function transformData() {

  // Lấy trang sheet cần chuyển đổi

  var sourceSheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();

  // Lấy dữ liệu từ sheet

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

  // Tạo sheet mới để lưu trữ dữ liệu đã chuyển đổi

  var destinationSheet = SpreadsheetApp.getActiveSpreadsheet().insertSheet("Test\_1");

  var currentGroup = 1;

  var row = 2;  // Bắt đầu từ dòng thứ 2 để tránh viết đè dòng tiêu đề

  // Tiêu đề cột

  var tieuDeCot = ['STT', 'MSSV', 'Tên Sv', 'Gmail', 'SĐT', 'Tên đề tài'];

  destinationSheet.getRange(1, 1, 1, tieuDeCot.length).setValues([tieuDeCot]).setFontWeight("bold");

  // Xử lý dữ liệu và chuyển đổi

  for (var i = 1; i < data.length; i++) { // Bắt đầu từ dòng đầu tiên của bảng tính

    var person1id    = data[i][1];

    var person1Name  = data[i][2];

    var person1Mail  = data[i][3];

    var person1Phone = data[i][4];

    var person2id    = data[i][6];

    var person2Name  = data[i][7];

    var person2Mail  = data[i][8];

    var person2Phone = data[i][9];

    var person3id    = data[i][11];

    var person3Name  = data[i][12];

    var person3Mail  = data[i][13];

    var person3Phone = data[i][14];

    var person4id    = data[i][16];

    var person4Name  = data[i][17];

    var person4Mail  = data[i][18];

    var person4Phone = data[i][19];

    var project      = data[i][21];

    // Ghi dữ liệu vào sheet mới với số nhóm tự động tăng dần

    destinationSheet.getRange(row, 1, 4, 1).mergeVertically().setFontWeight("bold").setHorizontalAlignment("center").setVerticalAlignment("middle").setFontSize(25).setValue(currentGroup);

    destinationSheet.getRange(row, 2).setValue(person1id);

    destinationSheet.getRange(row, 3).setValue(person1Name);

    destinationSheet.getRange(row, 4).setValue(person1Mail);

    destinationSheet.getRange(row, 5).setNumberFormat("@");

    destinationSheet.getRange(row, 5).setValue(person1Phone);

    destinationSheet.getRange(row, 6, 4, 1).setValue(project).mergeVertically().setFontWeight("bold").setHorizontalAlignment("center").setVerticalAlignment("middle");

    row++;

    destinationSheet.getRange(row, 2).setValue(person2id);

    destinationSheet.getRange(row, 3).setValue(person2Name);

    destinationSheet.getRange(row, 4).setValue(person2Mail);

    destinationSheet.getRange(row, 5).setNumberFormat("@");

    destinationSheet.getRange(row, 5).setValue(person2Phone);

    row++;

    destinationSheet.getRange(row, 2).setValue(person3id);

    destinationSheet.getRange(row, 3).setValue(person3Name);

    destinationSheet.getRange(row, 4).setValue(person3Mail);

    destinationSheet.getRange(row, 5).setNumberFormat("@");

    destinationSheet.getRange(row, 5).setValue(person3Phone);

    row++;

    destinationSheet.getRange(row, 2).setValue(person4id);

    destinationSheet.getRange(row, 3).setValue(person4Name);

    destinationSheet.getRange(row, 4).setValue(person4Mail);

    destinationSheet.getRange(row, 5).setNumberFormat("@");

    destinationSheet.getRange(row, 5).setValue(person4Phone);

    row++;

    // Thêm dòng trống giữa các nhóm

    row++;

    // Tăng số nhóm cho nhóm tiếp theo

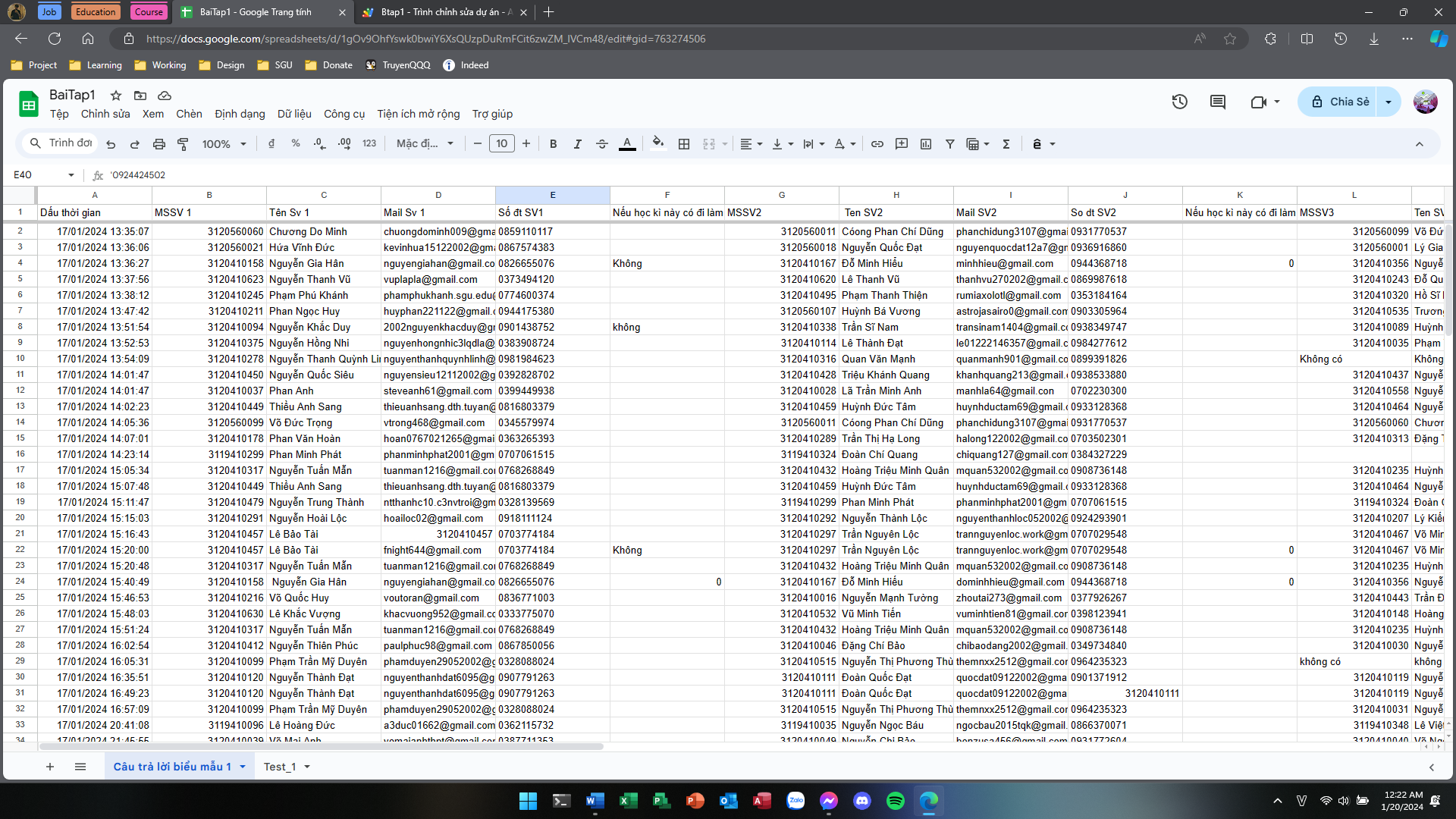
    currentGroup++;

  }

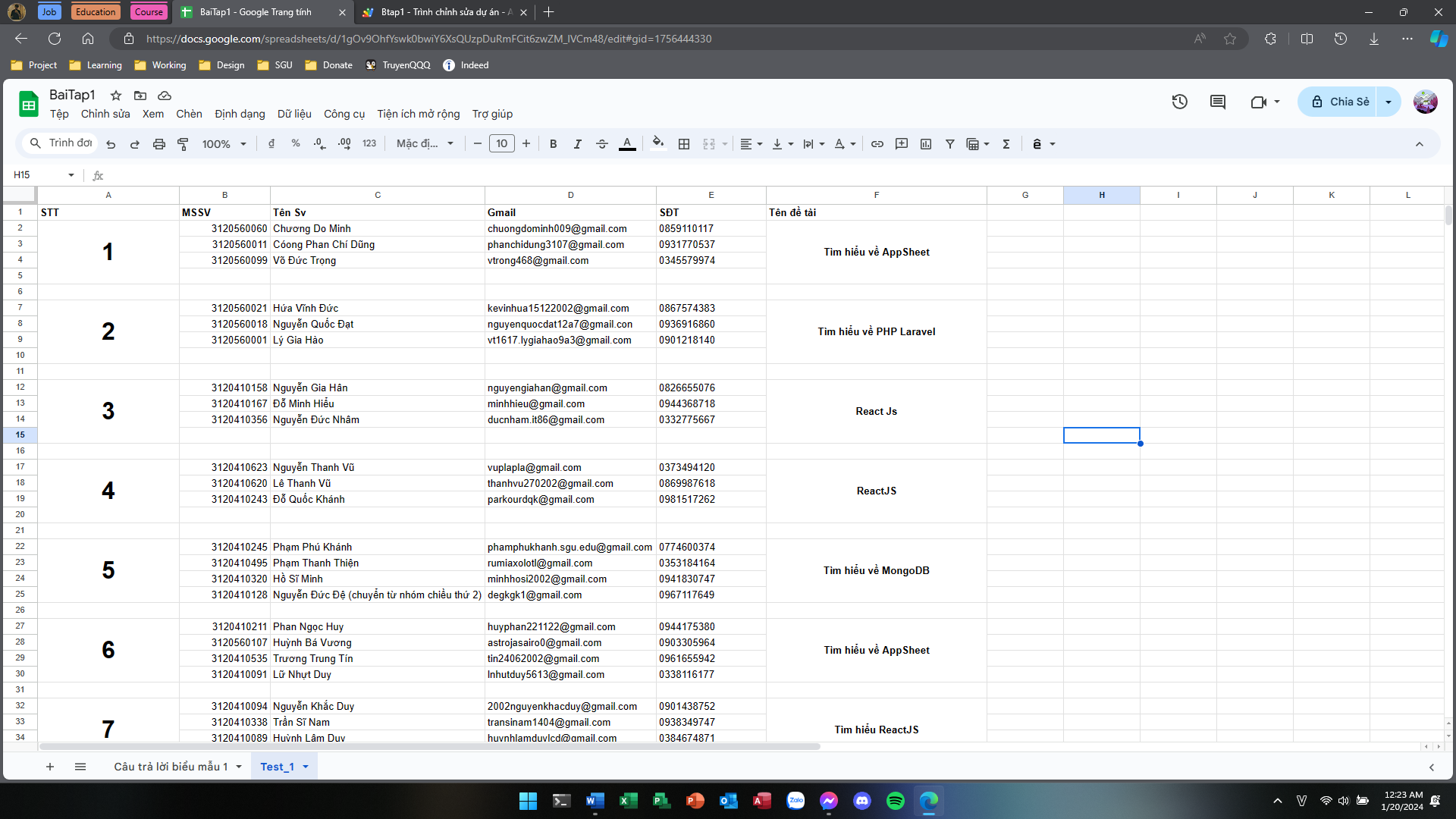
}

Giải thích nghiệp vụ đoạn code: Đầu tiên đọc và tạo một file sheets mới tương ứng cho kết quả đầu ra. Sau đó dùng vòng lặp để tiến hành quét qua toàn bộ dữ liệu, mỗi lần quét qua là cập nhật thông tin qua file sheet mới với format đã được định sẵn.

Kết quả thực nghiệm của đoạn code trên:



Hình 3. Trước khi chạy Script

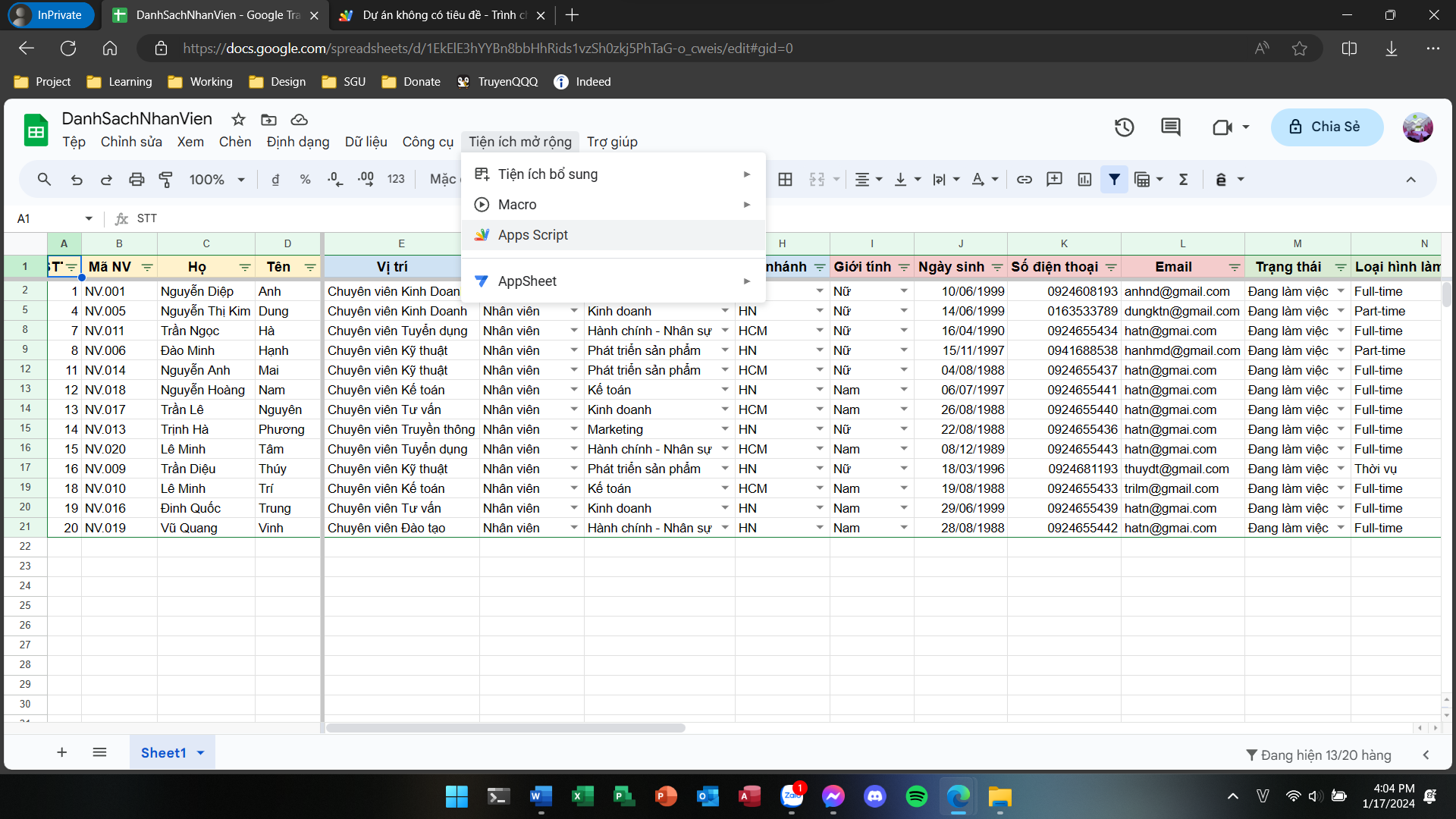


Hình 4. Sau khi chạy Script

Tuy nhiên đoạn code trên chưa thể xử lý được vấn đề trùng lắp dữ liệu liên quan tới nhóm hay các lỗi nhập liệu khác.

1. **Tách 1 file google sheet ra thành nhiều file google sheet theo điều kiện. File mẫu:**[**https://docs.google.com/spreadsheets/d/1yyI9GF0z7MpD65pNEJ7sdOpzy5nW7Wb\_ru5cSzPmIhE/edit?usp=sharing**](https://docs.google.com/spreadsheets/d/1yyI9GF0z7MpD65pNEJ7sdOpzy5nW7Wb_ru5cSzPmIhE/edit?usp=sharing)

Để xử lý câu 2, em sử dụng tính năng **App Script** trên sheets:



Hình 5. Sử dụng tính năng App Script có sẳn để giải quyết bài toán

Tính năng này cho phép chúng ta chèn mã script và thực thi trên trang script của chính mình. Sau đây là đoạn code mà em sử dụng để giải quyết bài toán:

* Giả định: Ta đang muốn lọc dữ liệu ra thành các sheet mới với điều kiện được lấy từ cột **Chức vụ**. Trong cột chức vụ bao gồm các trường dữ liệu như sau [Thực tập sinh, học việc, nhân viên, Quản lý]

const filterCondition = ["Thực tập sinh", "Học việc"]

const sheetName = "Sheet1"

function filterDataByChucVu() {

  // Mở bảng tính nguồn

  var trangNguon = SpreadsheetApp.getActiveSpreadsheet().getSheetByName(sheetName);

  // Lấy dữ liệu từ cột Chức Vụ (ví dụ: cột F chứa chức vụ)

  var duLieuNguon = trangNguon.getRange('F:F').getValues();

  // Tạo một mảng để lưu trữ dữ liệu thỏa mãn điều kiện

  var duLieuTraVe = [];

  // Duyệt qua dữ liệu nguồn và lọc theo nhiều điều kiện

  filterCondition.forEach(item => {

    for (var i = 0; i < duLieuNguon.length; i++) {

        if (duLieuNguon[i][0] === item) {

        duLieuTraVe.push(trangNguon.getRange(i + 1, 1, 1, trangNguon.getLastColumn()).getValues()[0]);

      }

    }

    // Tạo một bảng tính mới

    var bieuDoTraVe = SpreadsheetApp.create(item);

    // Lấy trang đầu tiên trong bảng tính mới

    var trangTraVe = bieuDoTraVe.getSheets()[0];

    // Chép dữ liệu vào trang đích

    trangTraVe.getRange(1, 1, duLieuTraVe.length, duLieuTraVe[0].length).setValues(duLieuTraVe);

    // reset lại data

    duLieuTraVe = [];

  })

}

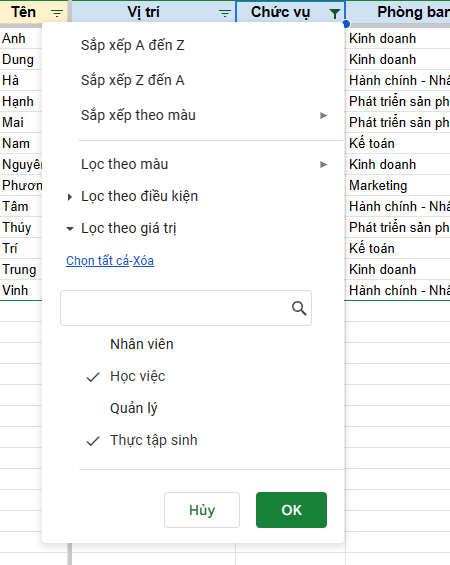
Giải thích đoạn code trên:

* **filterCondition:** là điều kiện dữ liệu để lấy các trường do ta đặt ra. Ví dụ trên đoạn code ta muốn lấy các bộ có chức vụ là **Thực tập sinh** và **nhân viên.**
* **sheetName:** là tên của trang sheet, trong sheet của em thì nó là **Sheet1**.
* Vì đây là hàm lọc dữ liệu theo **Chức vụ** do đó đoạn code var duLieuNguon = trangNguon.getRange('F:F').getValues() **F:F** có nghĩ là vị trí của cột chức vụ trên sheets.

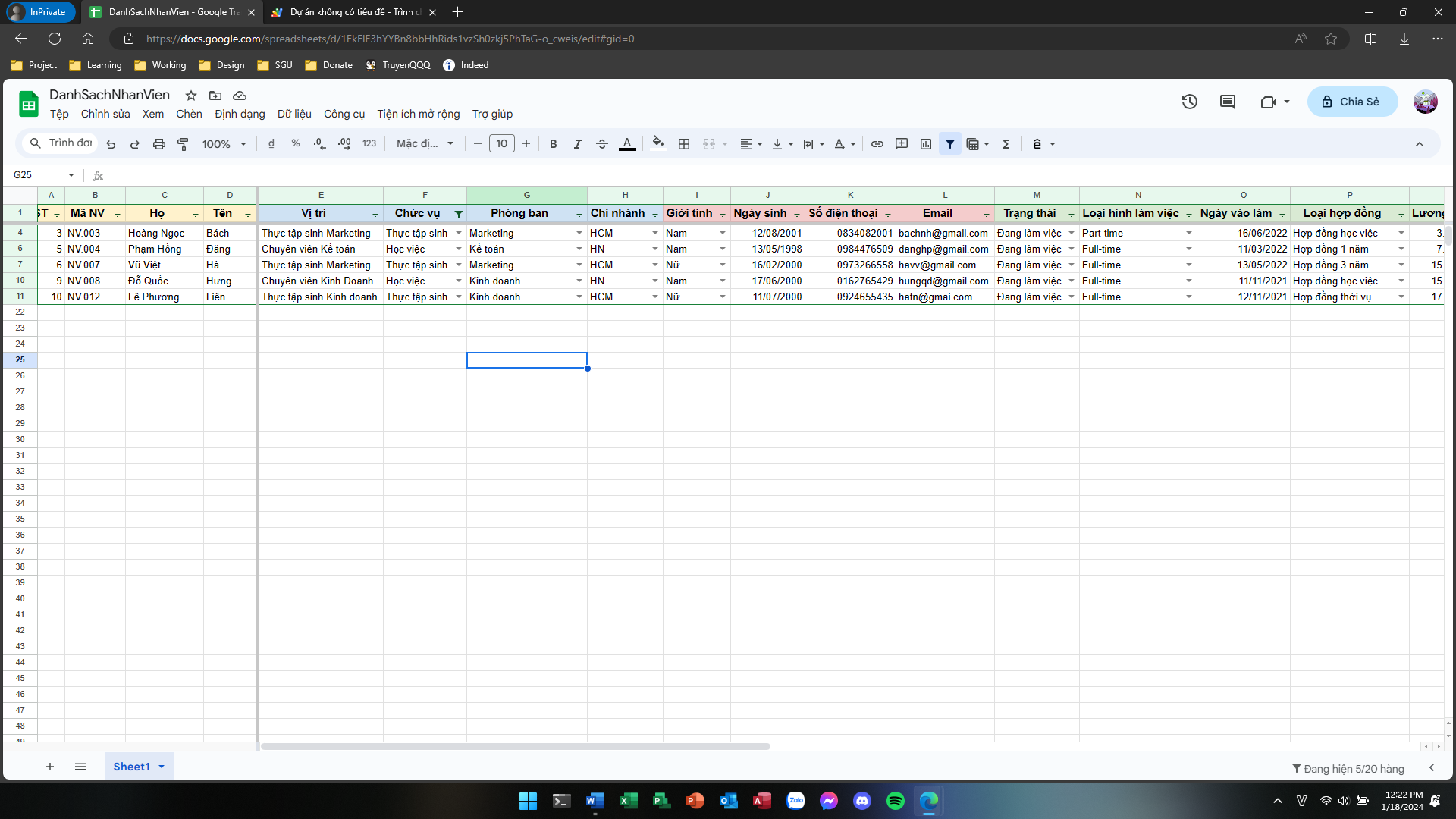
Kết quả thu được:

Để thử nghiệm tính thực thi của đoạn code trên ta có mô tả như sau:

* Ta thử nghiệm lọc dựa trên các công cụ lọc có sẵn trên sheets trước và kết quả thu được là:

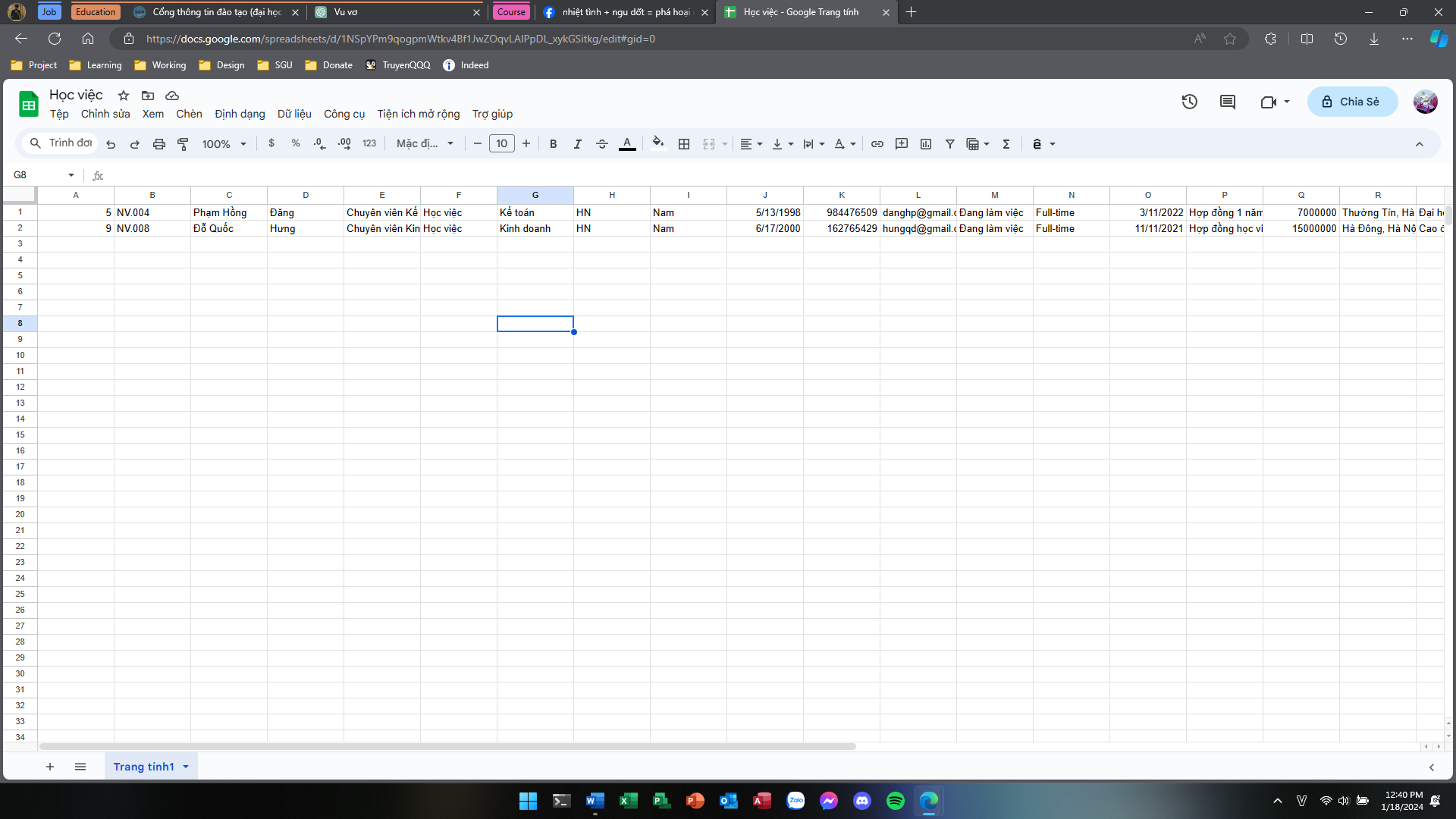


Hình 6. Sử dụng lọc theo Học viên và thực tập sinh

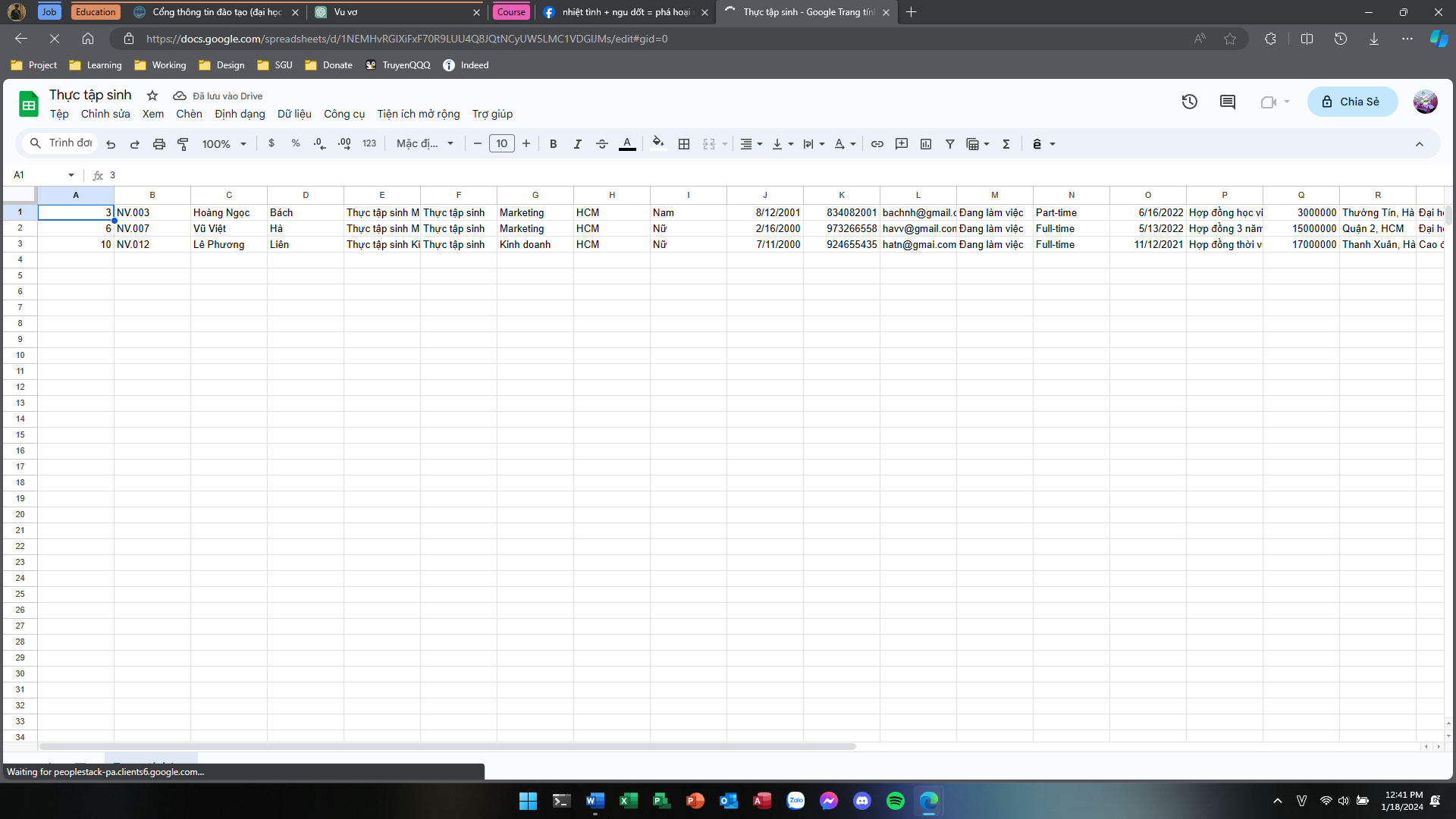


Hình 7. Kết quả trả về từ công cụ lọc trên sheets

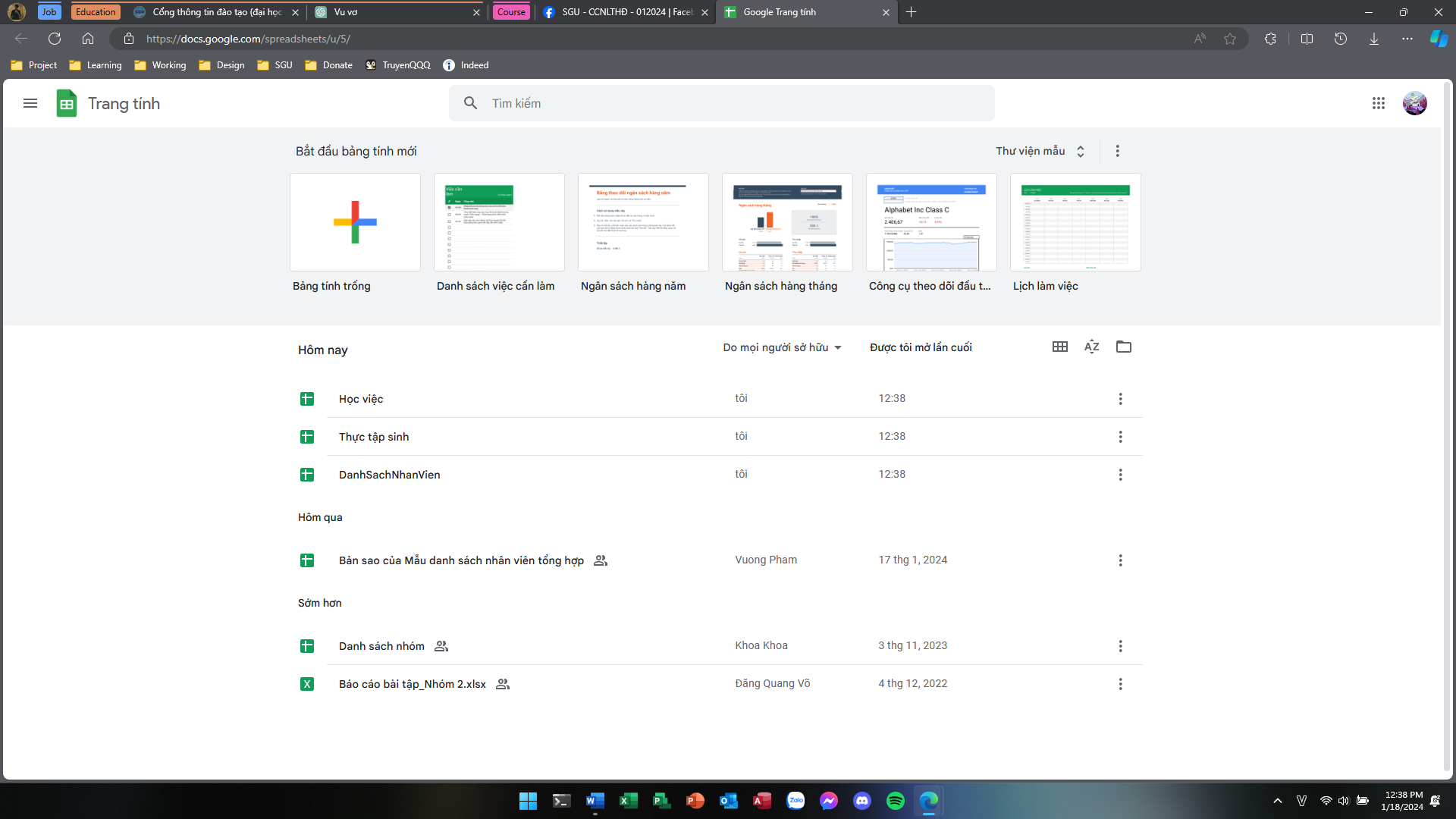
Sau đó, ta thử nghiệm khi chạy trên code:



Hình 8. Kết quả trả về từ đoạn code đã được thực thi, file Học việc



Hình 9. Kết quả trả về từ đoạn code, file thực tập sinh



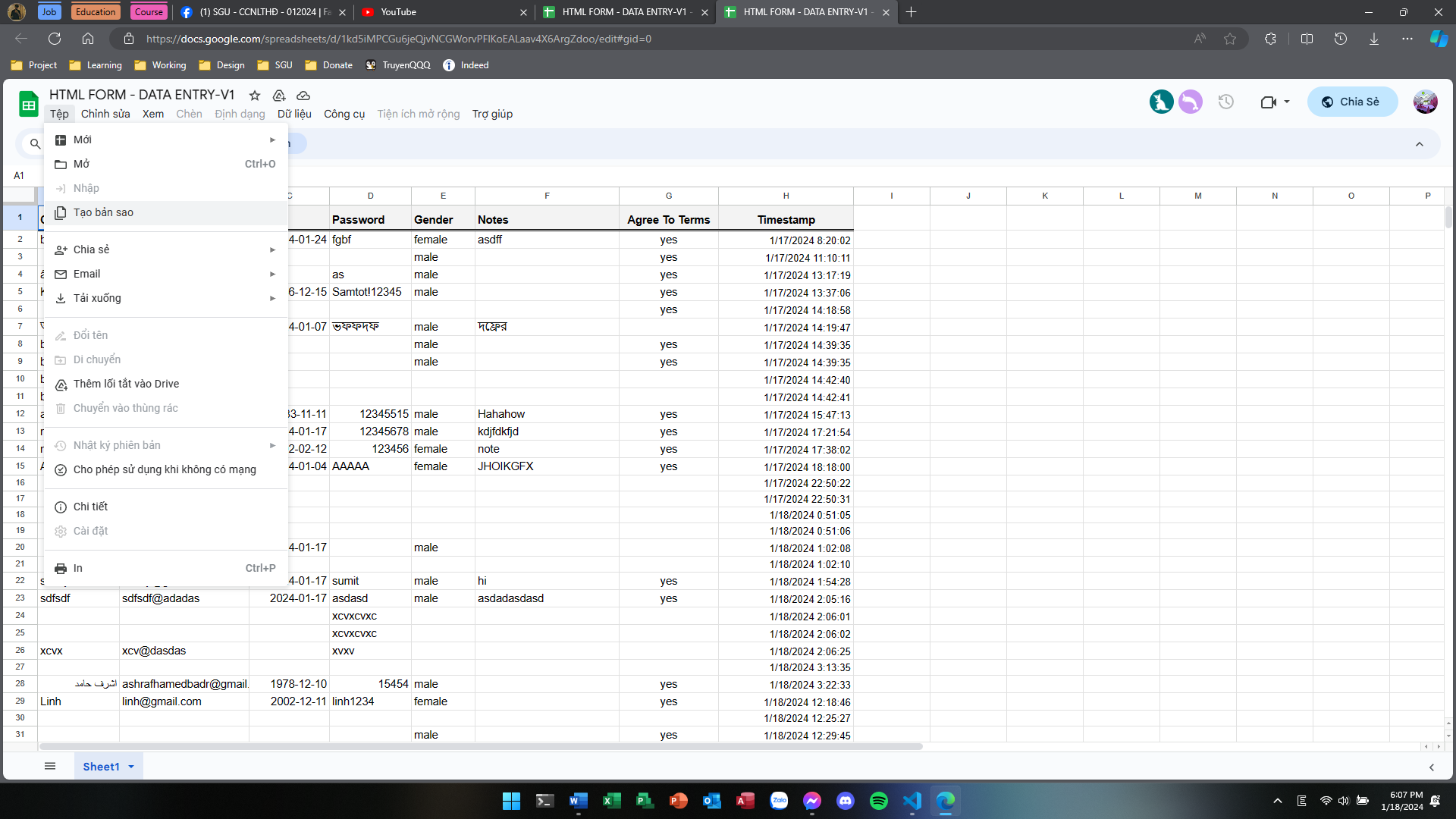
Hình 10. Thư mục được tạo như yêu cầu

1. **Xem và làm lại các video đã post (3 video)**

**Video 1:** [**https://www.youtube.com/watch?v=5Ta8TrPsdN8**](https://www.youtube.com/watch?v=5Ta8TrPsdN8)

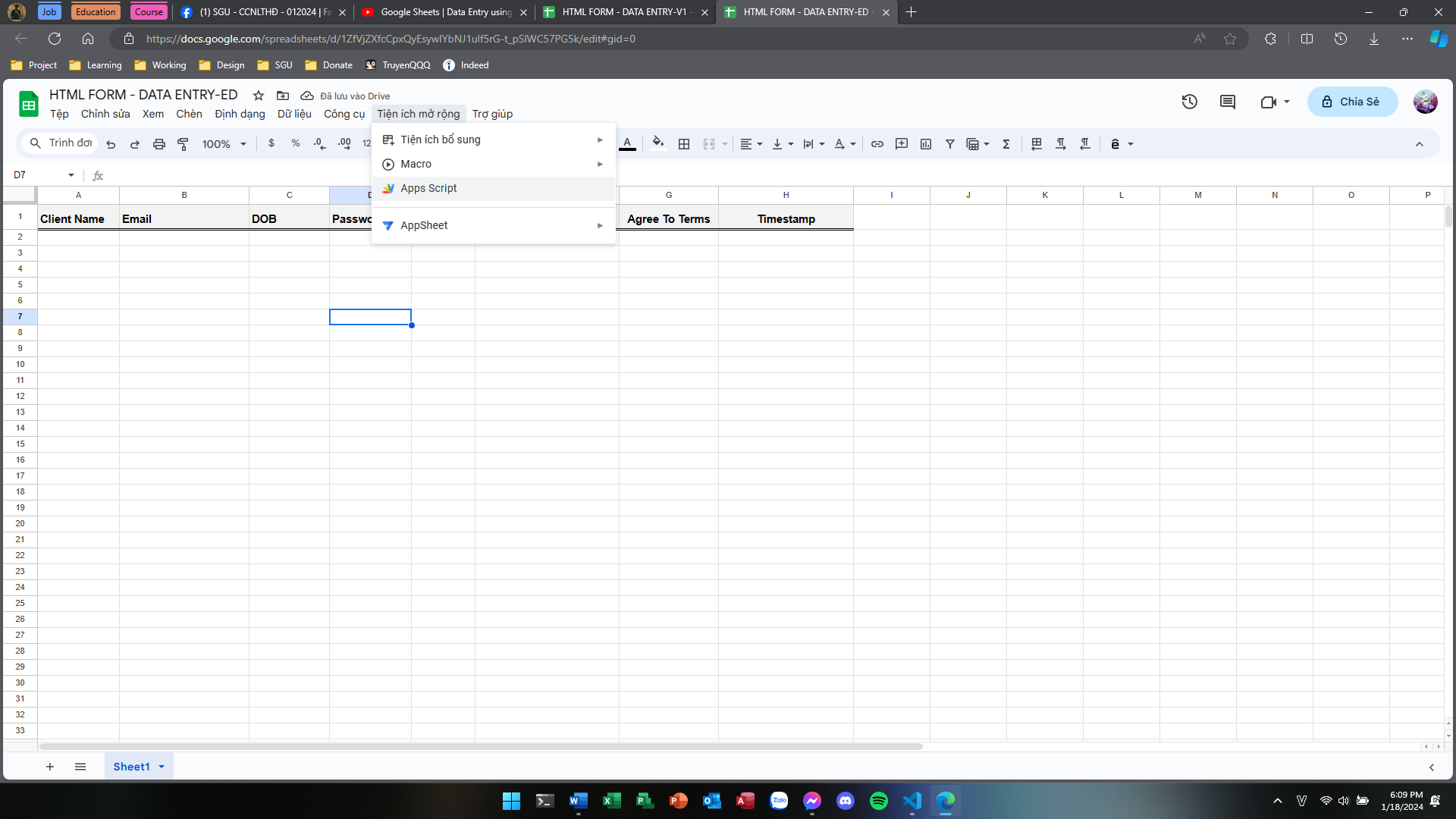
Giải thích sơ bộ: Video này minh họa cho chúng ta cách nhập dữ liệu từ một trang HTML có sẵn và dữ liệu được nhập đó sẽ được chuyển đến Google Sheet.

Đầu tiên, ta cần copy Sheets của tác giả về trang sheets cá nhân:

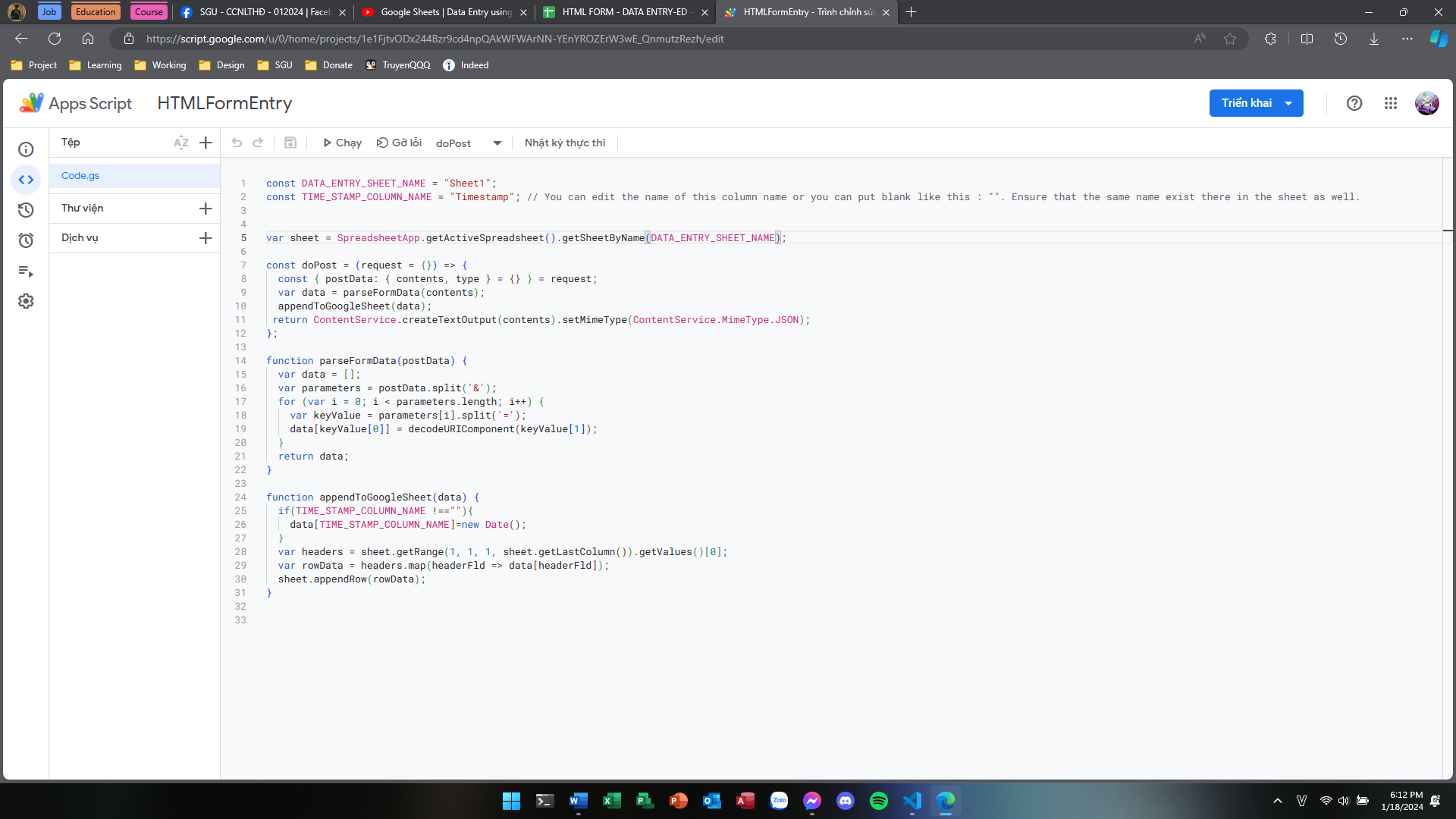


Hình 11. Sheets của tác giả

Sau đó ta clean hết bộ dữ liệu chuẩn bị để thực hiện nhập liệu vào form thông qua trang HTML. Ta sử dụng tiện ích App Script để tiến hành thực thi lấy API từ đoạn Script đã được nhúng sẵn của tác giả:



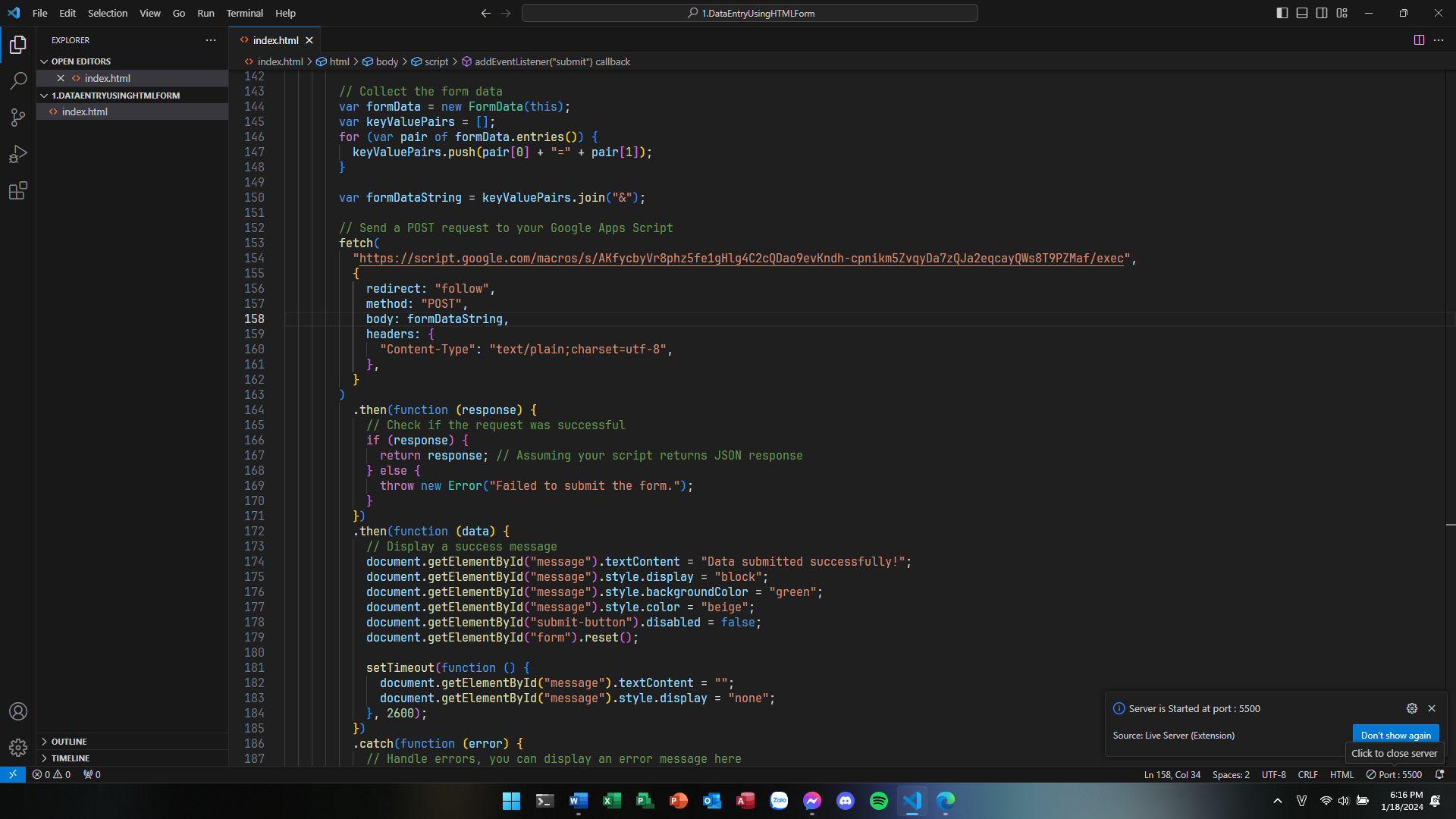
Hình 12. Vào App Script



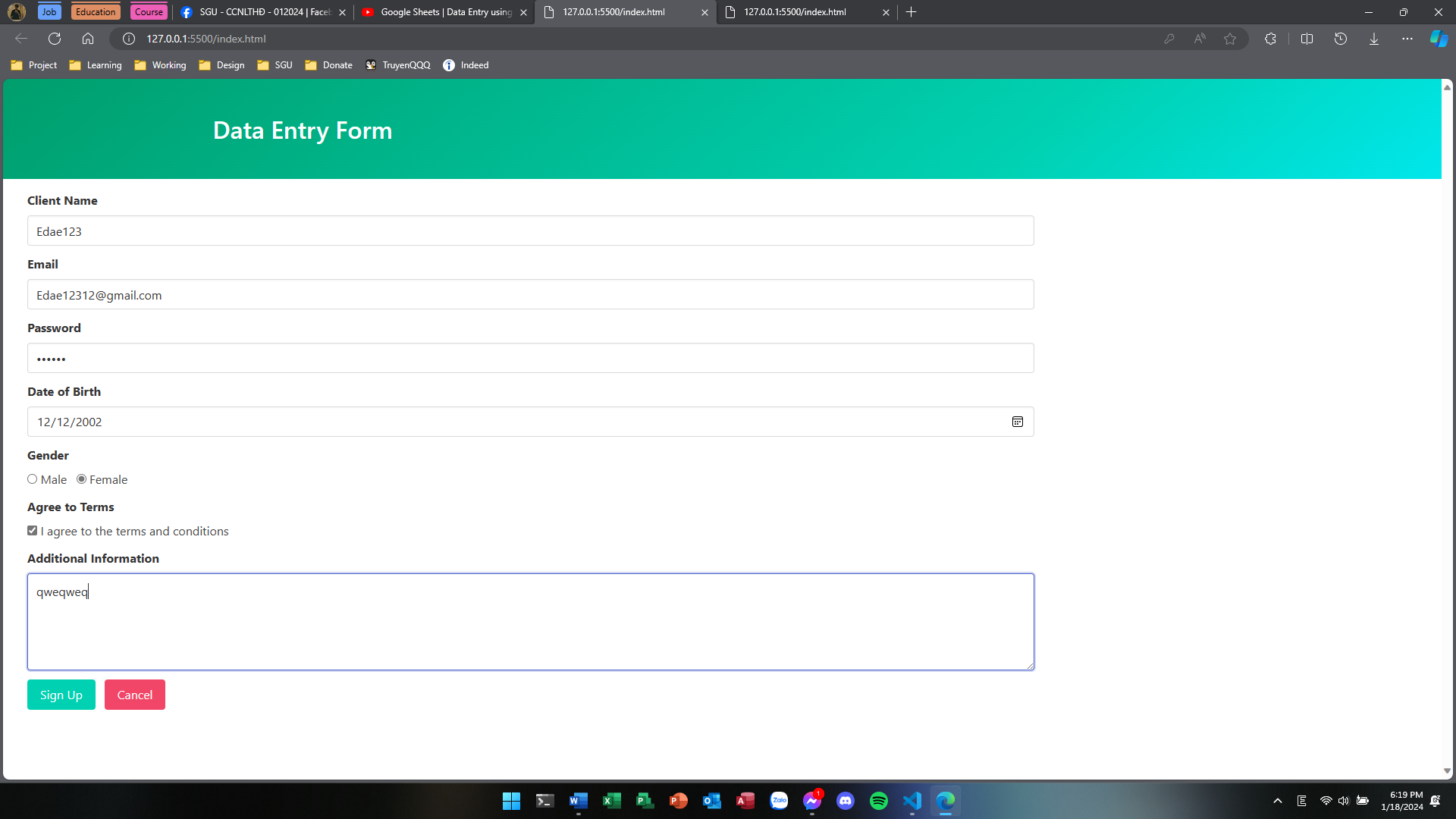
Hình 13. Script đã được gán sẵn

Sau đó ta làm theo các bước hướng dẫn của tác giả để lấy một API thực hiện giao thức HTTP Request để thực hiện hành động gửi dữ liệu từ form HTML lên App Sheets.

Bên cạnh đó, tác giả cũng dính kèm một trang HTML mẫu, ta down về và dán đoạn code API vào như hướng dẫn:

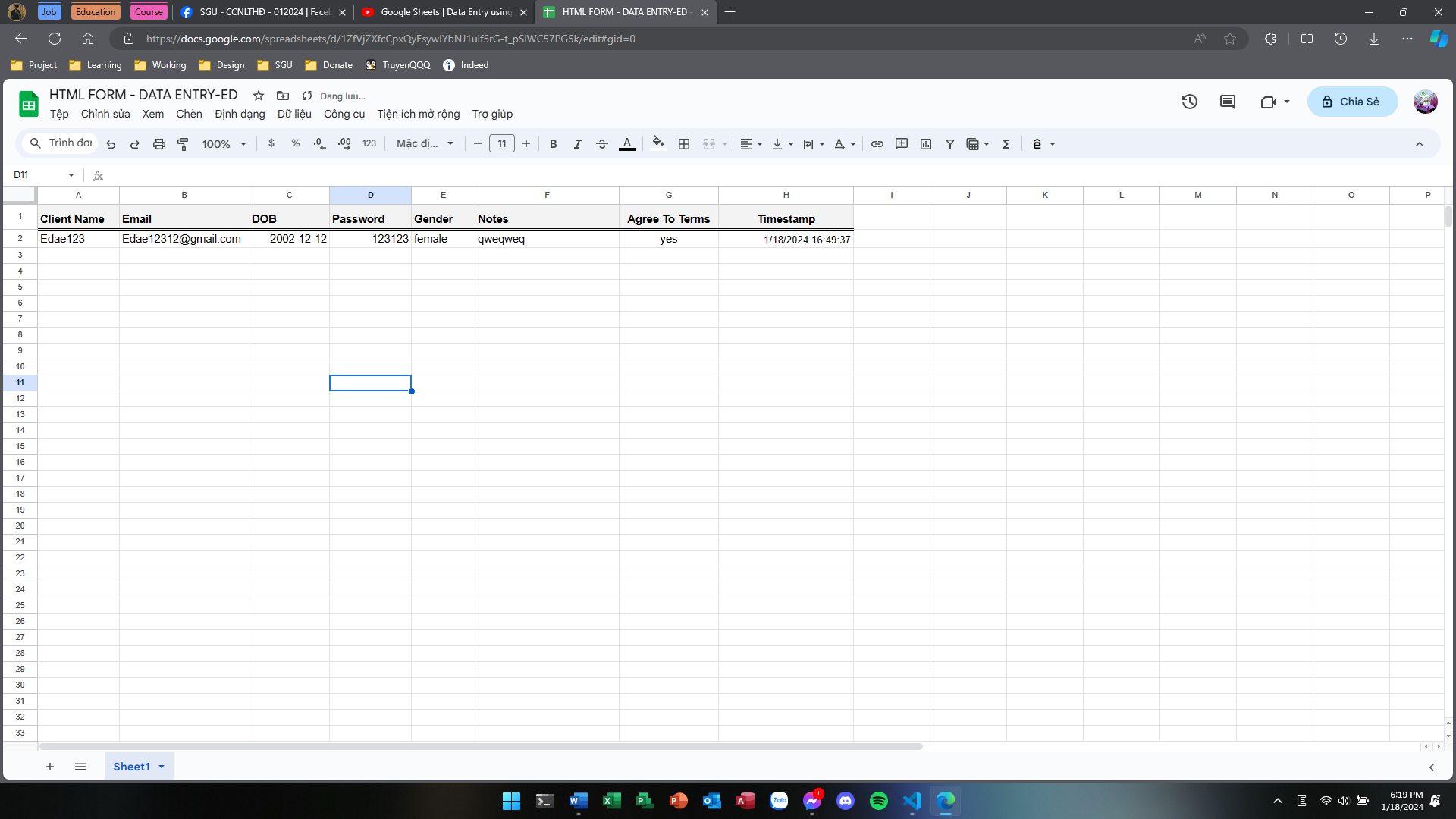


Hình 14. Dán URL vào trong code



Hình 15. Khởi động trang HTML và nhập liệu thành công

Sau khi nhập liệu trên form HTML, ta quay lại kiểm tra trên App Sheets và thành công:



Hình 16. Kiểm tra lại dữ liệu vừa nhập vào App Sheets

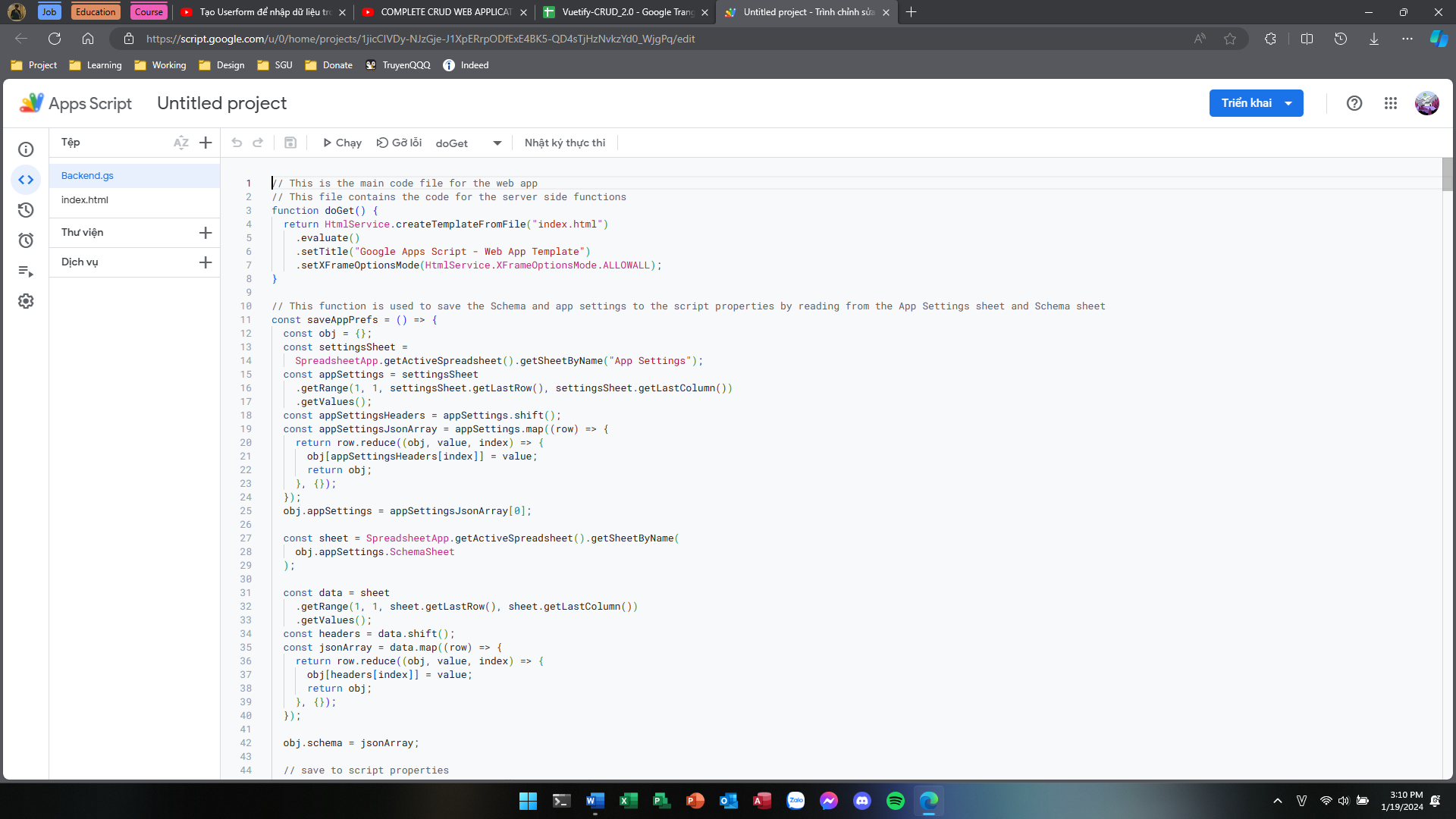
**Video 2:** [**https://www.youtube.com/watch?v=ZdDB5VVW8HM**](https://www.youtube.com/watch?v=ZdDB5VVW8HM)

**Giải thích sơ bộ:** Sử dụng Google Sheets để ứng dụng vào ứng dụng CRUD.

Đầu tiên, copy Sheets mà tác giả đã ghim sheets của bản thân và truy cập vào App Script của Sheets đó:



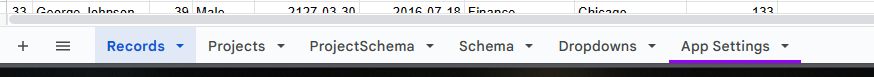
Hình 17. Copy Sheets của tác giả về



Hình 18. Truy cập vào App Script của tác giả

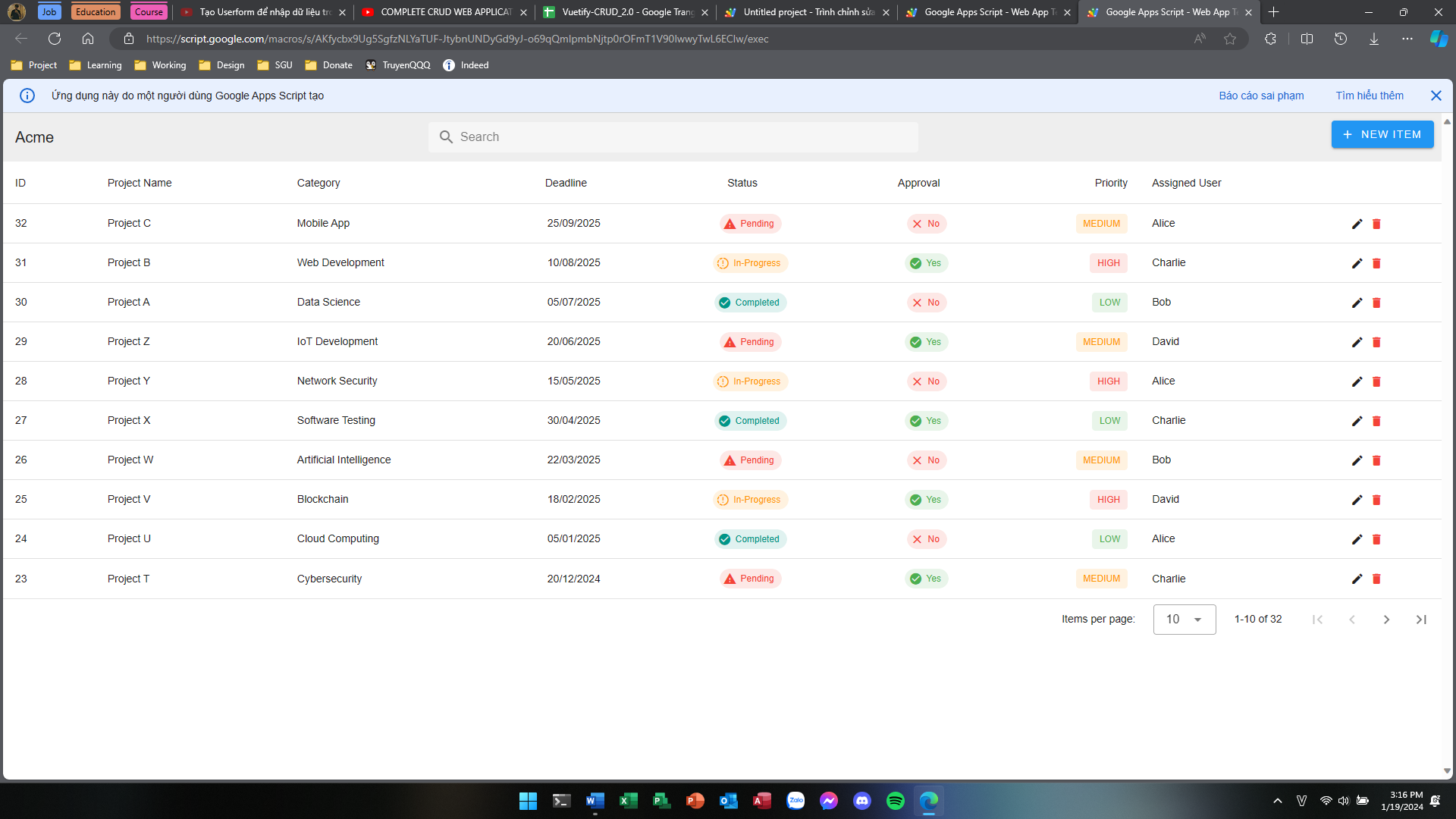
App Script trong này gồm có 2 file: **index.html** tượng trưng cho phần frontEnd, **backend.gs** tượng trưng cho phần backEnd của một hệ thống trang web

Tiến hành tiển khai dự án của App Script, copy **WebApp URL** sau khi triển khai. Sau đó quay lại trang Google Sheets để truy cập vào trang sheet có tên là **App Setting**:

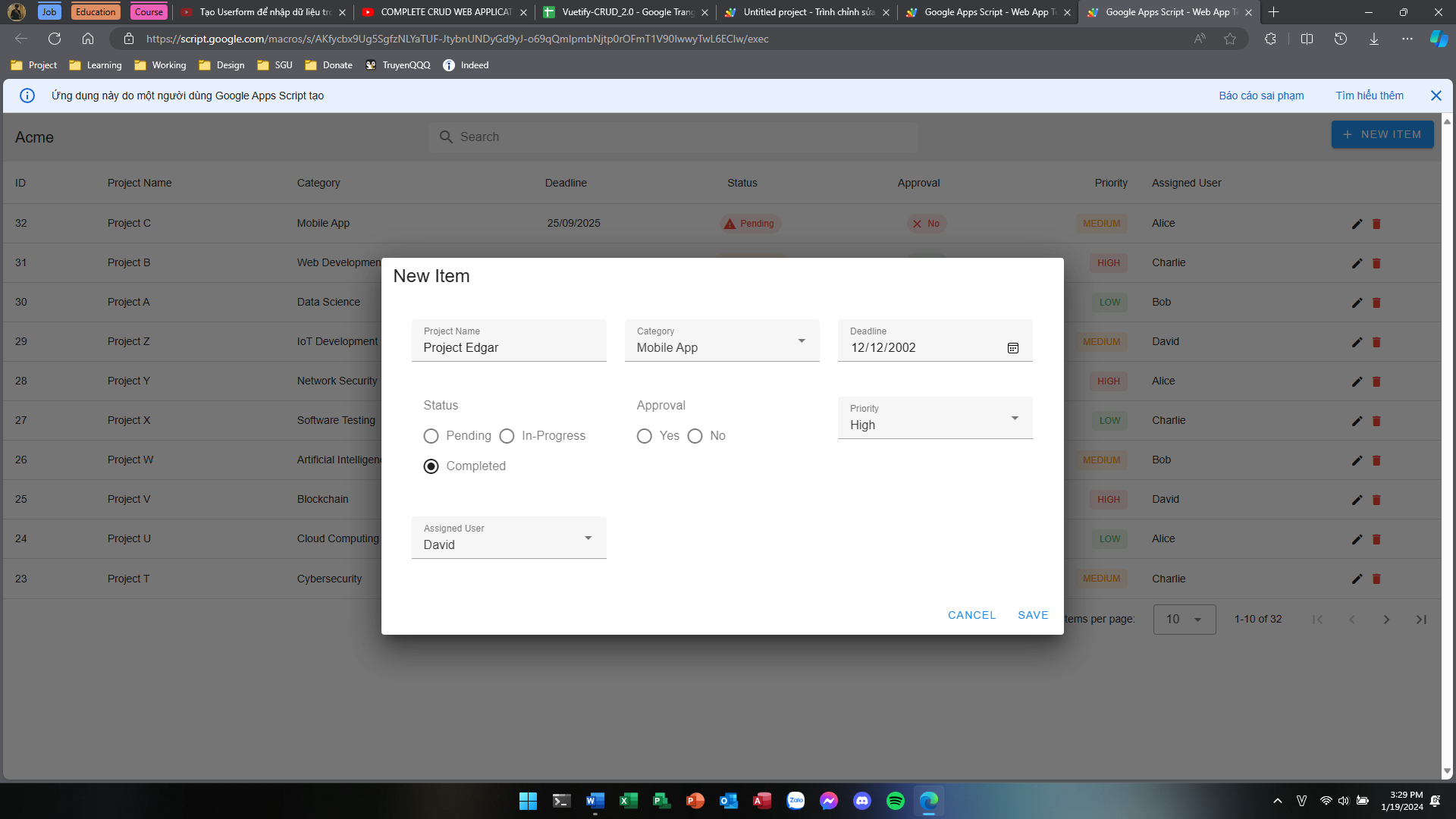


Hình 19. Danh sách các trang sheet được dính sẵn

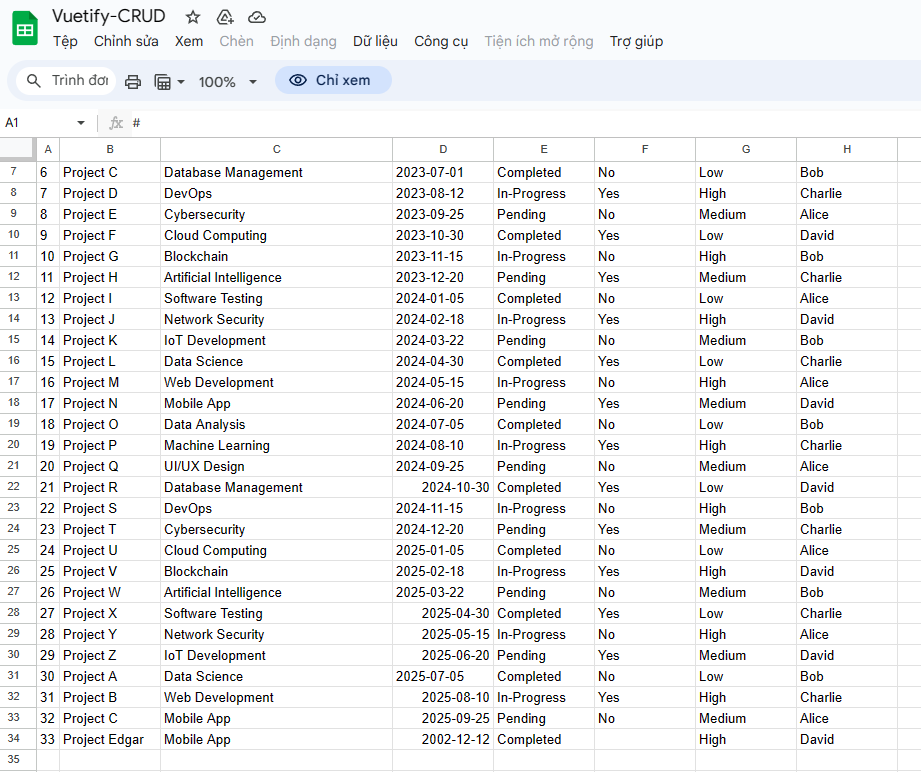
Ở bước trước, ta đã copy lại **WebApp URL** nên giờ ta paste nó vào trang web và mở nó ra, ta được giao diện web như sau:



Hình 20. Giao diện web

Ta tiến hành chạy thử demo với ứng dụng trên, các thông số nhập liệu như sau: 

Hình 21. Tiến hành chạy demo



Hình 22. Kiểm tra trên Google Sheets một dự án mới đã được thêm vào

**Video 3:** [**https://www.youtube.com/watch?v=7Ty4uk40kWk**](https://www.youtube.com/watch?v=7Ty4uk40kWk)

Giải thích sơ bộ: Hướng dẫn tạo Userform để nhập dữ liệu trong Google Sheets. Hướng dẫn cách code dể tạo ra FE và BE trong App Script.

Đầu tiên ta truy cập vào App Script và bắt đầu viết function để tạo ra một itemUI trên thanh sheet như sau, đoạn code được mô tả như sau:

function createMenu() {

  const ui = SpreadsheetApp.getUi();

  const menu = ui.createMenu("Customer Management System");

  menu.addItem("openUserForm", "loadUserForm");

  menu.addToUi();

}

function loadUserForm(){

  const width = 640, height = 480;

  const html = HtmlService.createHtmlOutputFromFile("index");

  html.setWidth(width).setHeight(height);

  SpreadsheetApp.getUi().showModalDialog(html, "Customer Management System");

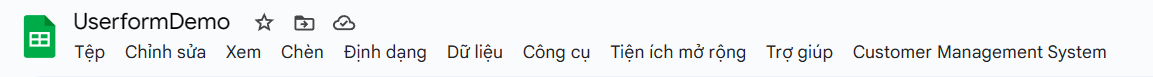
}

function onOpen(){

  createMenu();

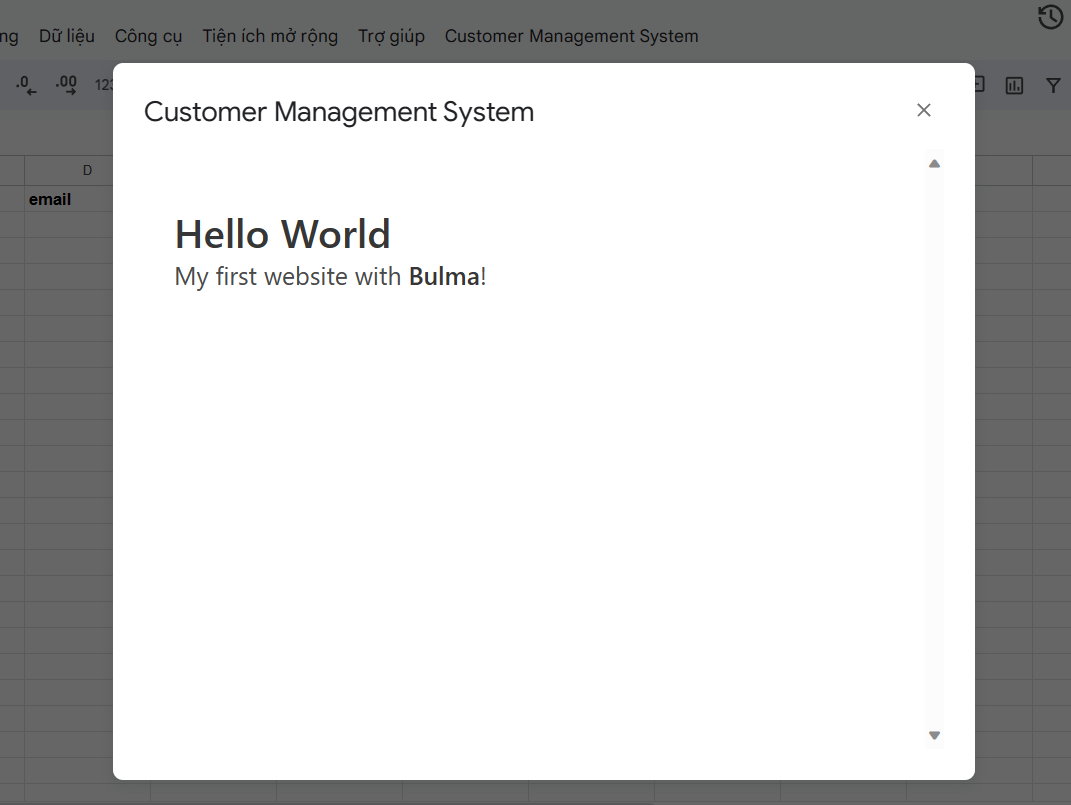
}

Thành quả mà chúng ta thu được:



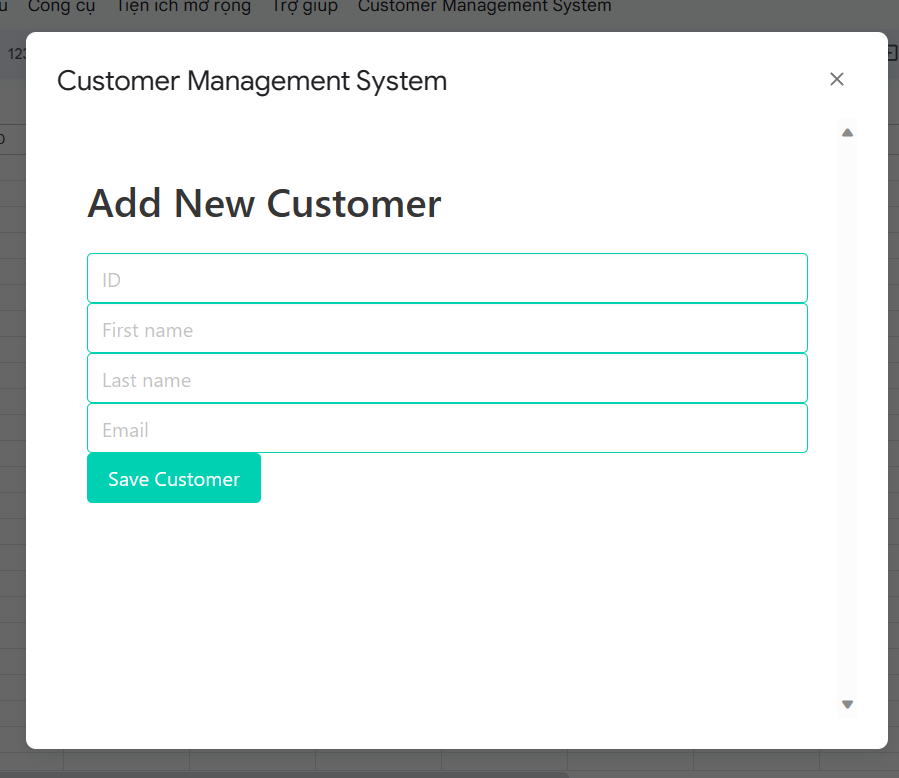
Hình 23. Customer Management System chính là thứ mà chúng ta tạo ra

Đây là giao diện lúc chúng ta khởi tạo, và từ giao diện này chúng ta sẽ viết code HTML để tạo ra một form nhập liệu cho người dùng một cách chỉnh chu hơn:



Hình 24. Form nhập liệu cơ bản ban đầu

Để cho việc lập trình form thuận tiện ta sử dụng form có sẵn từ **Bulma.io:**



Hình 25. Giao diện nhập liệu có sẵn từ Bulma.io

Đoạn code FrontEnd cho giao diện nhập trên được thể hiện như sau:

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Hello Bulma!</title>

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.4/css/bulma.min.css">

  </head>

  <body>

    <section class="section">

    <div class="container">

      <h1 class="title">

        Add New Customer

      </h1>

      <div class="field">

        <div class="control">

          <input id="iID" class="input is-primary" type="text" placeholder="ID">

      </div>

      <div class="field">

        <div class="control">

          <input id="iFirstName" class="input is-primary" type="text" placeholder="First name">

      </div>

      <div class="field">

        <div class="control">

          <input id="iLastName" class="input is-primary" type="text" placeholder="Last name">

      </div>

      <div class="field">

        <div class="control">

          <input id="iEmail" class="input is-primary" type="text" placeholder="Email">

      </div>

      <div class="buttons">

        <button id="btnSaveCustomer" class="button is-primary">Save Customer</button>

      </div>

    </div>

    </section>

    <script>

      const btnSaveCustomer = document.getElementById("btnSaveCustomer");

      btnSaveCustomer.addEventListener("click", function(){

        const inputID = document.getElementById("iID").value;

        const inputFirstName = document.getElementById("iFirstName").value;

        const inputLastName = document.getElementById("iLastName").value;

        const inputEmail = document.getElementById("iEmail").value;

        // kiểm tra

        if(inputID.length == 0 || inputFirstName.length == 0 || inputLastName.length == 0 ||inputEmail.length == 0) {

          google.script.run.errorMsg();

        } else{

          const data = {

            id: inputID,

            firstname: inputFirstName,

            lastname: inputLastName,

            email: inputEmail

          }

          google.script.run.postData(data);

          //clear data

          inputID = '';

          inputFirstName = '';

          inputLastName = '';

          inputEmail = '';

        }

      });

    </script>

  </body>

</html>

Trong đó chứa các thành phần giao diện và các đoạn script thực thi, chủ yếu là bắt sự kiện của các nút bấm và các field nhập để gửi dữ liệu đến cho phần BackEnd thực hiện

Đoạn code của phần BackEnd được thể hiện như sau đây:

function createMenu() {

  const ui = SpreadsheetApp.getUi();

  const menu = ui.createMenu("Customer Management System");

  menu.addItem("openUserForm", "loadCustomerForm");

  menu.addToUi();

}

function loadCustomerForm(){

  const width = 640, height = 480;

  const html = HtmlService.createHtmlOutputFromFile("index");

  html.setWidth(width).setHeight(height);

  const ui = SpreadsheetApp.getUi();

  ui.showSidebar(html);

}

function onOpen(){

  createMenu();

}

function postData(data){

  const ws = SpreadsheetApp.getActiveSpreadsheet().getSheetByName("data");

  ws.appendRow([data.id, data.firstname, data.lastname, data.email]);

}

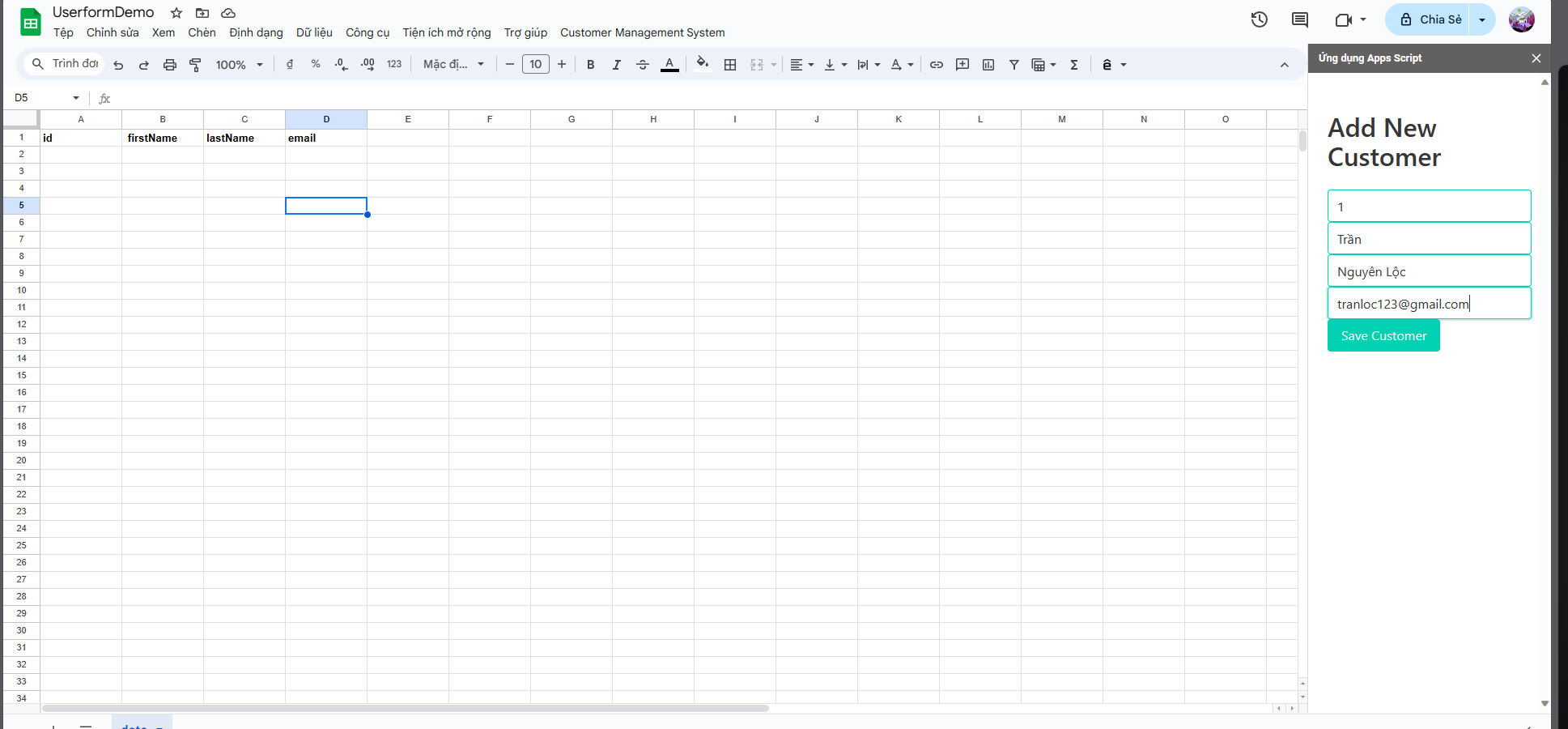
function errorMsg() {

  Browser.msgBox("Bạn cần điền đầy đủ thông tin")

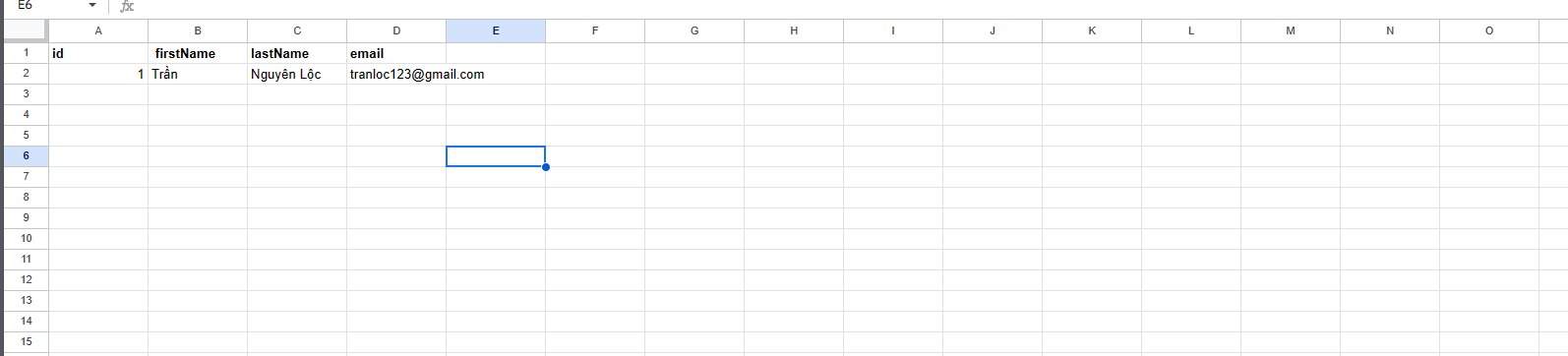
}

Đoạn code bao gồm việc tạo ra nút bấm **Customer Management System** trên trang Sheet cũng như việc khởi tạo khung cho phần FrontEnd. Bên cạnh đó, nó còn nhận đữ liệu từ phía FrontEnd và update dữ liệu vừa nhận được lên lại Google Sheets.

Bên đây là demo chương trình:



Hình 26. Nhập liệu cho chương trình và ấn nút thực thi



Hình 27. Dữ liệu được nhập lên Google Sheets thành công