var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

var streetSheetName = "JALAN SANGGUL 4";

function doGet(request) {

  var streetSheetName = "JALAN SANGGUL 4";

  PropertiesService.getScriptProperties().setProperty("streetSheetName", streetSheetName);

  var html = HtmlService.createTemplateFromFile('Index').evaluate()

  .setTitle("Admin Login")

  .addMetaTag('viewport', 'width=device-width, initial-scale=1')

  return html.setXFrameOptionsMode(HtmlService.XFrameOptionsMode.ALLOWALL);

}

//Check Login

function checkLogin(loginUsername, loginPassword) {

  var url = 'https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1151242353';

  var ss= SpreadsheetApp.openByUrl(url);

  var webAppSheet = ss.getSheetByName("AdminClerkData");

  var getLastRow =  webAppSheet.getLastRow();

  var found\_record = '';

  for(var i = 1; i <= getLastRow; i++)

  {

     if(webAppSheet.getRange(i, 3).getValue() == "1") //admin=1, clerk=0

     {

       if(webAppSheet.getRange(i, 1).getValue() == loginUsername &&

         webAppSheet.getRange(i, 2).getValue() == loginPassword)

       {

         found\_record = 'ADMIN\_TRUE';

       }

       if(found\_record == '')

       {

         found\_record = 'FALSE';

       }

     }

     else if(webAppSheet.getRange(i, 3).getValue() == "0")

     {

       if(webAppSheet.getRange(i, 1).getValue() == loginUsername &&

         webAppSheet.getRange(i, 2).getValue() == loginPassword)

       {

         found\_record = 'CLERK\_TRUE';

       }

       if(found\_record == '')

       {

         found\_record = 'FALSE';

       }

     }

  }

  return found\_record;

}

//----------------------------------------------------------------

//--------- Manage Payment ---------------------------------------

function checkPaymentUsername(pm\_uname) {

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var ss = SpreadsheetApp.openByUrl(url);

  var webAppSheet = ss.getSheetByName("USERNAMES");

  var getLastRow =  webAppSheet.getLastRow();

  var streetSheetName = "JALAN SANGGUL 4";

  var found\_record = '';

  var name = '';

  for(var i = 2; i <= getLastRow; i++) {

   if(webAppSheet.getRange(i, 1).getValue() == pm\_uname) {

     found\_record = 'TRUE';

     name = webAppSheet.getRange(i, 4).getValue().toUpperCase() + " " + webAppSheet.getRange(i, 5).getValue().toUpperCase();

     streetSheetName = webAppSheet.getRange(i, 3).getValue().toUpperCase();

   }

  }

  PropertiesService.getScriptProperties().setProperty("streetSheetName", streetSheetName);

  if(found\_record == '') {

    found\_record = 'FALSE';

  }

  return [found\_record, pm\_uname,name];

}

function GetRecords(pm\_uname,filter) {

  var filteredDataRangeValues = GetUsernameAssociatedProperties(pm\_uname);

  var resultArray = GetPaymentRecords(filteredDataRangeValues,filter);

  var resultFilter = getYears();

  var result = {

    data: resultArray,

    filter: resultFilter

  };

  return result;

}

function getYears() {

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var ss= SpreadsheetApp.openByUrl(url);

  var yearSheet = ss.getSheetByName("Configuration");

  var getLastRow = yearSheet.getLastRow();

  var return\_array = [];

  for(var i = 2; i <= getLastRow; i++)

  {

      if(return\_array.indexOf(yearSheet.getRange(i, 2).getDisplayValue()) === -1) {

        return\_array.push(yearSheet.getRange(i, 2).getDisplayValue());

      }

  }

  return return\_array;

}

function GetUsernameAssociatedProperties(pm\_uname) {

  var filteredDataRangeValues = '';

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var ss = SpreadsheetApp.openByUrl(url);

  var displaySheet = ss.getSheetByName("USERNAMES");

  var dataRangeValues = displaySheet.getDataRange().getValues();

  if (pm\_uname.toUpperCase() == 'ADMIN') {

    dataRangeValues.shift();

    filteredDataRangeValues = dataRangeValues;

  } else {

    filteredDataRangeValues = dataRangeValues.filter(row => row[0].toUpperCase() == pm\_uname.toUpperCase());

  }

  return filteredDataRangeValues;

}

function GetPaymentRecords(userProperties,filter) {

  var streetSheetName = PropertiesService.getScriptProperties().getProperty("streetSheetName"); // Added

  var transpose = m => m[0].map((\_, i) => m.map(x => x[i]));

  var resultArray = [];

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var ss = SpreadsheetApp.openByUrl(url);

  var displaySheet = ss.getSheetByName(streetSheetName);

  var addressValues = displaySheet.getRange("B:C").getValues();

  var paidMonthValues = displaySheet.getRange(1, 7, displaySheet.getLastRow(), displaySheet.getLastColumn() - 6).getValues();

  userProperties.forEach((v, i) => {

    var userHouseNumber = v[1];

    var userStreet = v[2];

    var column = addressValues.reduce(function callbackFn(accumulator, currentValue, index, array) {

      if (currentValue[0] == userHouseNumber && currentValue[1] == userStreet) {

        return index

      } else {

        return accumulator

      }

    }, '');

    Logger.log(filter)

    Logger.log(paidMonthValues);

    if(filter=="None"){

      var result = transpose(paidMonthValues).map(function callbackFn(element, index, array) {

        return [element[0], userHouseNumber, userStreet, element[column] || '']

      });

    }else{

      var result = transpose(paidMonthValues).map(function callbackFn(element, index, array) {

        if(element[0].includes(filter))return [element[0], userHouseNumber, userStreet, element[column] || '']

      });

    }

    resultArray = resultArray.concat(result);

  })

  //Remove null elements

  resultArray = resultArray.filter(element=>{

    Logger.log(element!=null)

    return element != null;

  });

  return resultArray;

}

function enterAmount(inputInfo){

  var url = 'https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1151242353';

  var ss = SpreadsheetApp.openByUrl(url);

  var ws = ss.getSheetByName("Manual Payment");

  var time = new Date();

  var formattedDate = Utilities.formatDate(time, "GMT", "MM-dd-yyyy HH:mm:ss");

  var repeatedPayment = ( inputInfo.amount / 50 );

  for(var i = 0; i < repeatedPayment; i++){

           ws.appendRow([formattedDate,inputInfo.uname]);

  }

}

//----------------------------------------------------------------

//--------- CHANGE PASSWORD (ADMIN) ---------------------------------------

function changeAdminPassword(loginUsername, newAdminPw) {

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var sheet = SpreadsheetApp.openByUrl(url).getSheetByName("Admin Data");

  var range = sheet.getRange("A2:A").createTextFinder(loginUsername).matchEntireCell(true).findNext();

  if (range) {

    range.offset(0, 1).setValue(newAdminPw);

  }

}

//----------------------------------------------------------------

//--------- CHANGE PASSWORD (CLERK) ---------------------------------------

function changeClerkPassword(loginUsername, newClerkPw) {

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var sheet = SpreadsheetApp.openByUrl(url).getSheetByName("Clerk Data");

  var range = sheet.getRange("A2:A").createTextFinder(loginUsername).matchEntireCell(true).findNext();

  if (range) {

    range.offset(0, 1).setValue(newClerkPw);

  }

}

//--------------------------------------------------------------------------------------------------------

//---------------GLOBAL VARIABLES---------------------------------------------------------------------------

function globalVariables(){

  var array = {

    spreadsheetId        : '1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8',

    clerkdataRange       : 'Clerk Data!A2:B',

    clerkidRange         : 'Clerk Data!A2:A',

    clerklastCol         : 'B',

    clerkinsertRange     : 'Clerk Data!A1:B1',

    clerksheetID         : '1019728383',

    dataRange            : 'USERNAMES!A2:H',

    idRange              : 'USERNAMES!A2:A',

    lastColumn           : 'H',

    insertRange          : 'USERNAMES!A1:H1',

    sheetID              : '0'

  };

  return array;

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------------

//------------------FUCNTION FOR CLERK CRUD PART-----------------------------------------------------------------------------------------------------------------

function processClerkForm(formObject){

  if(formObject.clerk\_uname && checkClerkID(formObject.clerk\_uname)){

    updateClerkData(getClerkFormValues(formObject),globalVariables().spreadsheetId,getClerkRangeByID(formObject.clerk\_uname)); //Update data if form pass ID

  }else{

    appendClerkData(getClerkFormValues(formObject),globalVariables().spreadsheetId,globalVariables().clerkinsertRange); //Append Data if form does not pass ID

  }

  return getClerkLastTenRows();

}

function getClerkFormValues(formObject){

  if(formObject.clerk\_uname && checkClerkID(formObject.clerk\_uname)){

    var values = [[

                  formObject.clerk\_uname,

                  formObject.clerk\_pw]];

  }else{

    var values = [[

                  formObject.clerk\_uname,

                  formObject.clerk\_pw]];

  }

  return values;

}

function appendClerkData(values, spreadsheetId,range){

  var valueRange = Sheets.newRowData();

  valueRange.values = values;

  var appendRequest = Sheets.newAppendCellsRequest();

  appendRequest.adminsheetID = spreadsheetId;

  appendRequest.rows = valueRange;

  var results = Sheets.Spreadsheets.Values.append(valueRange, spreadsheetId, range,{valueInputOption: "RAW"});

}

function readClerkData(spreadsheetId,range){

  var result = Sheets.Spreadsheets.Values.get(spreadsheetId, range);

  return result.values;

}

function updateClerkData(values,spreadsheetId,range){

  var valueRange = Sheets.newValueRange();

  valueRange.values = values;

  var result = Sheets.Spreadsheets.Values.update(valueRange, spreadsheetId, range, {

  valueInputOption: "RAW"});

}

function deleteClerkData(ID){

  var startIndex = getClerkRowIndexByID(ID);

  var deleteRange = {

                      "sheetId"     : globalVariables().clerksheetID,

                      "dimension"   : "ROWS",

                      "startIndex"  : startIndex,

                      "endIndex"    : startIndex+1

                    }

  var deleteRequest= [{"deleteDimension":{"range":deleteRange}}];

  Sheets.Spreadsheets.batchUpdate({"requests": deleteRequest}, globalVariables().spreadsheetId);

  return getClerkLastTenRows();

}

function checkClerkID(ID){

  var idList = readClerkData(globalVariables().spreadsheetId,globalVariables().clerkidRange,).reduce(function(a,b){return a.concat(b);});

  return idList.includes(ID);

}

function getClerkRangeByID(id){

  if(id){

    var idList = readClerkData(globalVariables().spreadsheetId,globalVariables().clerkidRange);

    for(var i=0;i<idList.length;i++){

      if(id==idList[i][0]){

        return 'Clerk Data!A'+(i+2)+':'+globalVariables().clerklastCol+(i+2);

      }

    }

  }

}

function getClerkRecordById(id){

  if(id && checkClerkID(id)){

    var result = readClerkData(globalVariables().spreadsheetId,getClerkRangeByID(id));

    return result;

  }

}

function getClerkRowIndexByID(id){

  if(id){

    var idList = readClerkData(globalVariables().spreadsheetId,globalVariables().clerkidRange);

    for(var i=0;i<idList.length;i++){

      if(id==idList[i][0]){

        var rowIndex = parseInt(i+1);

        return rowIndex;

      }

    }

  }

}

function getClerkLastTenRows(){

  var lastRow = readClerkData(globalVariables().spreadsheetId,globalVariables().clerkdataRange).length+1;

  if(lastRow<=11){

    var range = globalVariables().clerkdataRange;

  }else{

    var range = 'Clerk Data!A'+(lastRow-9)+':'+globalVariables().clerklastCol;

  }

  var lastTenRows = readClerkData(globalVariables().spreadsheetId,range);

  return lastTenRows;

}

function getClerkAllData(){

  var data = readClerkData(globalVariables().spreadsheetId,globalVariables().clerkdataRange);

  return data;

}

//-------------------------------------------------------------------------------------------------------------------------------------------------------------

//------------------FUCNTION FOR RESIDENT CRUD PART-----------------------------------------------------------------------------------------------------------------

function processResForm(formObject){

  if(formObject.username && checkResID(formObject.username)){

    updateResData(getResFormValues(formObject),globalVariables().spreadsheetId,getResRangeByID(formObject.username));

  }else{

    appendResData(getResFormValues(formObject),globalVariables().spreadsheetId,globalVariables().insertRange);

  }

  return getResLastTenRows();

}

function getResFormValues(formObject){

  if(formObject.username && checkResID(formObject.username)){

    var values = [[

                  formObject.username,

                  formObject.housenumber,

                  formObject.street,

                  formObject.firstname,

                  formObject.lastname,

                  formObject.noOfPaid,

                  formObject.password,

                  formObject.phone]];

  }else{

    var values = [[

                  formObject.username,

                  formObject.housenumber,

                  formObject.street,

                  formObject.firstname,

                  formObject.lastname,

                  formObject.noOfPaid,

                  formObject.password,

                  formObject.phone]];

  }

  return values;

}

function appendResData(values, spreadsheetId,range){

  var valueRange = Sheets.newRowData();

  valueRange.values = values;

  var appendRequest = Sheets.newAppendCellsRequest();

  appendRequest.sheetID = spreadsheetId;

  appendRequest.rows = valueRange;

  var results = Sheets.Spreadsheets.Values.append(valueRange, spreadsheetId, range,{valueInputOption: "RAW"});

}

function readResData(spreadsheetId,range){

  var result = Sheets.Spreadsheets.Values.get(spreadsheetId, range);

  return result.values;

}

function updateResData(values,spreadsheetId,range){

  var valueRange = Sheets.newValueRange();

  valueRange.values = values;

  var result = Sheets.Spreadsheets.Values.update(valueRange, spreadsheetId, range, {

  valueInputOption: "RAW"});

}

function deleteResData(ID){

  var startIndex = getResRowIndexByID(ID);

  var deleteRange = {

                      "sheetId"     : globalVariables().sheetID,

                      "dimension"   : "ROWS",

                      "startIndex"  : startIndex,

                      "endIndex"    : startIndex+1

                    }

  var deleteRequest= [{"deleteDimension":{"range":deleteRange}}];

  Sheets.Spreadsheets.batchUpdate({"requests": deleteRequest}, globalVariables().spreadsheetId);

  return getResLastTenRows();

}

function checkResID(ID){

  var idList = readResData(globalVariables().spreadsheetId,globalVariables().idRange,).reduce(function(a,b){return a.concat(b);});

  return idList.includes(ID);

}

function getResRangeByID(id){

  if(id){

    var idList = readResData(globalVariables().spreadsheetId,globalVariables().idRange);

    for(var i=0;i<idList.length;i++){

      if(id==idList[i][0]){

        return 'USERNAMES!A'+(i+2)+':'+globalVariables().lastColumn+(i+2);

      }

    }

  }

}

function getResRecordById(id){

  if(id && checkResID(id)){

    var result = readResData(globalVariables().spreadsheetId,getResRangeByID(id));

    return result;

  }

}

function getResRowIndexByID(id){

  if(id){

    var idList = readResData(globalVariables().spreadsheetId,globalVariables().idRange);

    for(var i=0;i<idList.length;i++){

      if(id==idList[i][0]){

        var rowIndex = parseInt(i+1);

        return rowIndex;

      }

    }

  }

}

function getResLastTenRows(){

  var lastRow = readResData(globalVariables().spreadsheetId,globalVariables().dataRange).length+1;

  if(lastRow<=11){

    var range = globalVariables().dataRange;

  }else{

    var range = 'USERNAMES!A'+(lastRow-9)+':'+globalVariables().lastColumn;

  }

  var lastTenRows = readResData(globalVariables().spreadsheetId,range);

  return lastTenRows;

}

function getResAllData(){

  var data = readResData(globalVariables().spreadsheetId,globalVariables().dataRange);

  return data;

}

//GET DROPDOWN LIST FROM GOOGLE SHEETS

function getDropdownList(range){

  var list = readResData(globalVariables().spreadsheetId,range);

  return list;

}

//SEARCH DATA (SEARCH FORM)

function searchResData(formObject){

  var result = [];

  if(formObject.searchtext){

      var data = Sheets.Spreadsheets.Values.get(globalVariables().spreadsheetId, globalVariables().dataRange).values;

      for(var i=0;i<data.length;i++){

        for(var j=0;j<data[i].length;j++){

          if(data[i][j].toLowerCase().search(formObject.searchtext.toLowerCase())!=-1){

            result.push(data[i]);

            break;   //break from loop

          }

        }

      }

  }

  return result;

}

//RESET RESIDENT PASSWORD

function admin\_resetResPw(username, password){

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var sheet = SpreadsheetApp.openByUrl(url).getSheetByName("USERNAMES");

  var range = sheet.getRange("A2:A").createTextFinder(username).matchEntireCell(true).findNext();

  if (range) {

    range.offset(0,6).setValue(password);

  }

}

//RESET CLERK PASSWORD

function admin\_resetClerkPw(clerk\_uname, clerk\_pw){

  var url = "https://docs.google.com/spreadsheets/d/1bM8l6JefFsPrlJnTWf56wOhnuSjdIwg3hMbY1tN1Zp8/edit#gid=1775459006";

  var sheet = SpreadsheetApp.openByUrl(url).getSheetByName("Clerk Data");

  var range = sheet.getRange("A2:A").createTextFinder(clerk\_uname).matchEntireCell(true).findNext();

  if (range) {

    range.offset(0,1).setValue(clerk\_pw);

  }

}

//INCLUDE HTML FILES

function include(filename) {

  return HtmlService.createHtmlOutputFromFile(filename)

      .getContent();

}