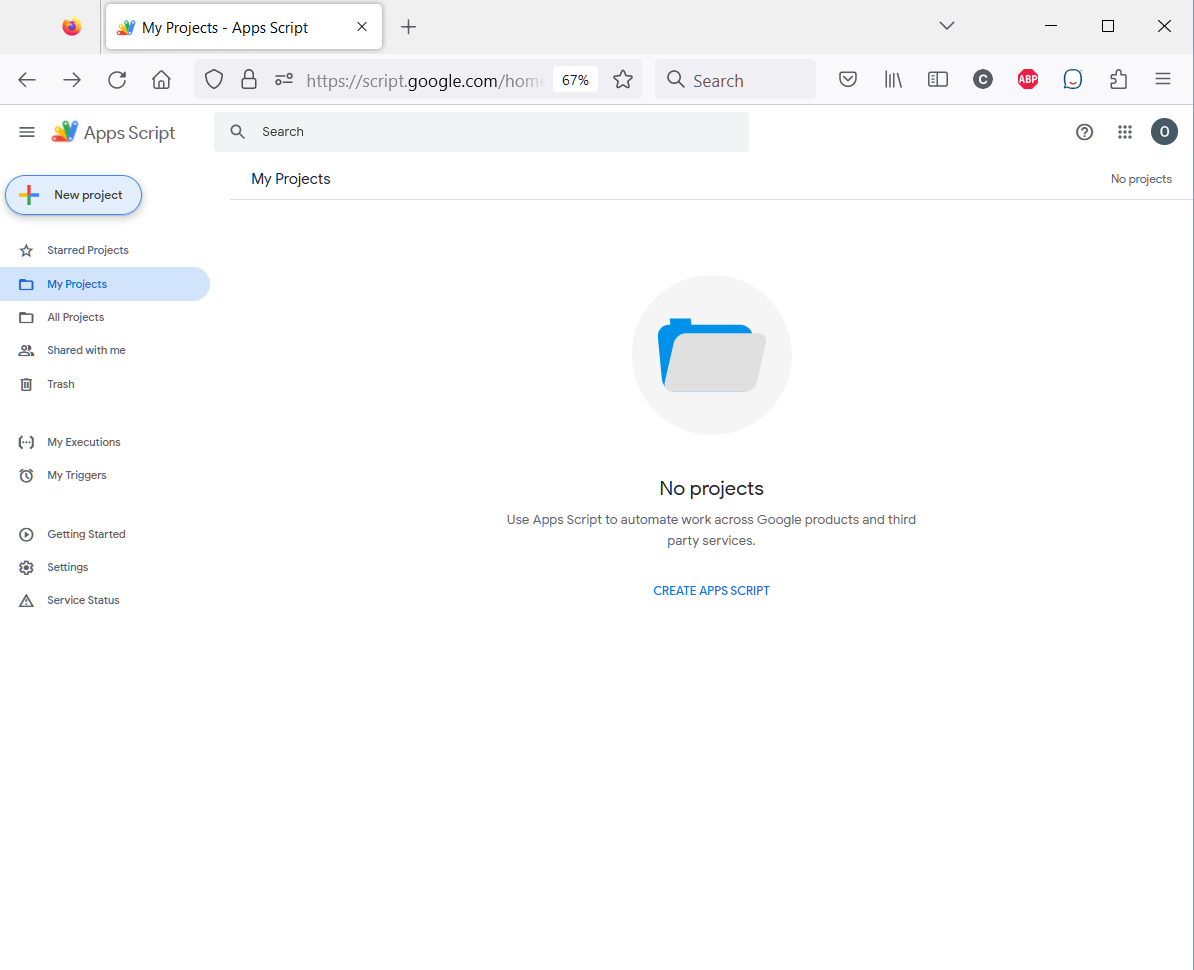
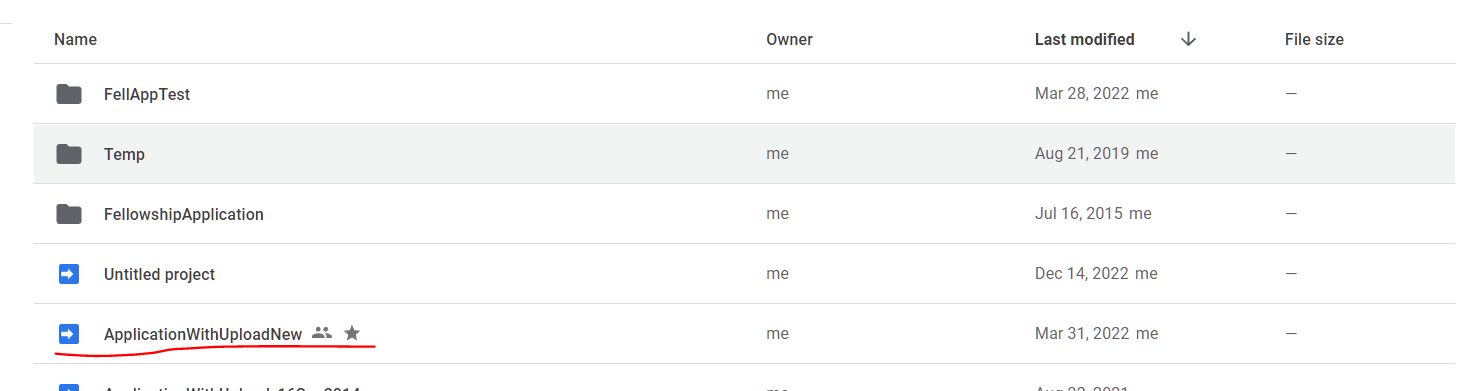
# Create Weill Cornell Pathology Fellowship Application

# Create Apps Script

* + 1. <https://script.google.com/home>
    2. Click on “CREATE APPS SCRIPT”



* + 1. Copy all files from existing script “ApplicationWithUploadNew”. These files can be copied from orderflex\src\App\FellAppBundle\Util\GoogleForm\FellowshipApplication
    2. This script is stored in the Google Drive
    3. All script can be founded here <https://script.google.com/home>
    4. Alternatively, all Google Api Scripts can be pull from github repository <https://github.com/cinava/fellowship-application> using a Chrome browser extension “Google Apps Script GitHub Assistant” ” (see 4 below).



# Create folders in Google Drive to store application’s spreadsheets and uploaded files

All files are located here orderflex\src\App\FellAppBundle\Util\GoogleForm\FellowshipApplication

* + 1. Create folder “FellowshipApplication”
    2. In “FellowshipApplication” folder, create empty file “config.json”
    3. In “FellowshipApplication” folder, create new folder “Responses”
    4. In “Responses” folder, create new folders:

FellowshipApplicantUploads

RecommendationLetters

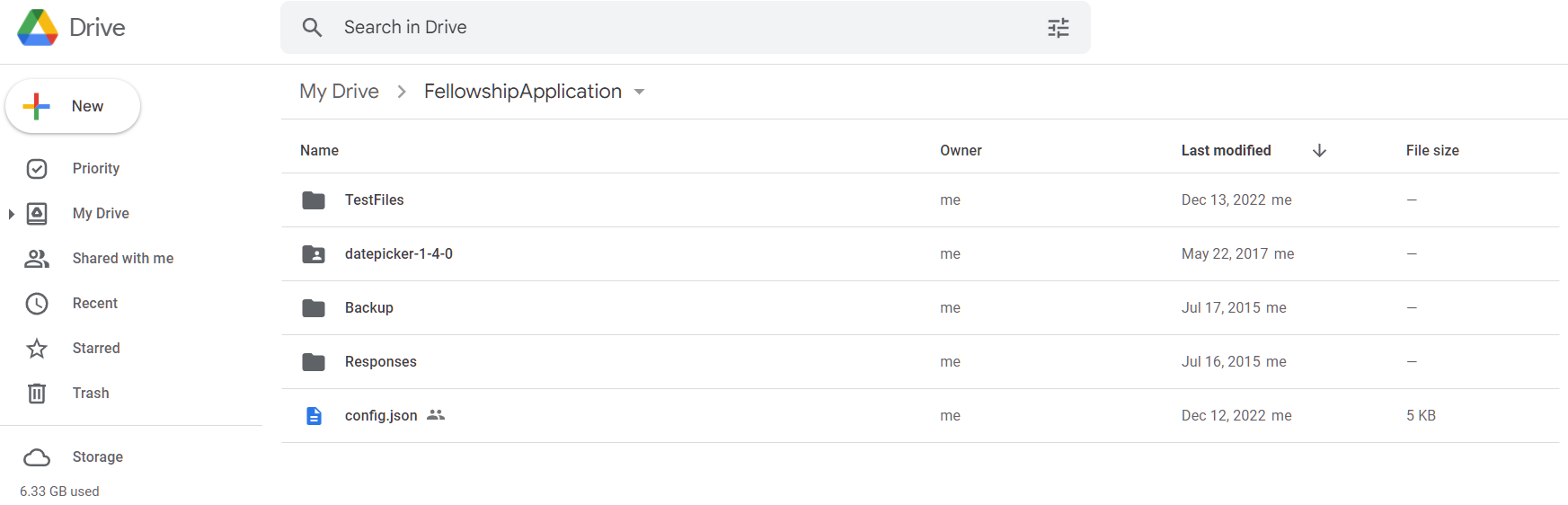
Spreadsheets

Template

* + 1. In “Template” folder, create new files:

BackupSpreadsheet – copy/paste from orderflex\src\App\FellAppBundle\Util\GoogleForm\FellowshipApplication\ BackupSpreadsheet.xlsx

TemplateSpreadsheet - copy/paste from orderflex\src\App\FellAppBundle\Util\GoogleForm\FellowshipApplication\ TemplateSpreadsheet.xlsx



# Adding IDs to the script ApplicationWithUploadNew

* Assign file id of the BackupSpreadsheet to \_backupSSKey
* Assign file id of the TemplateSpreadsheet to \_templateSSKey
* Assign folder id of the Spreadsheets to \_destinationFolder
* Assign folder name “FellowshipApplicantUploads” to \_dropbox
* Assign file id of the “config.json” to \_configFolderId

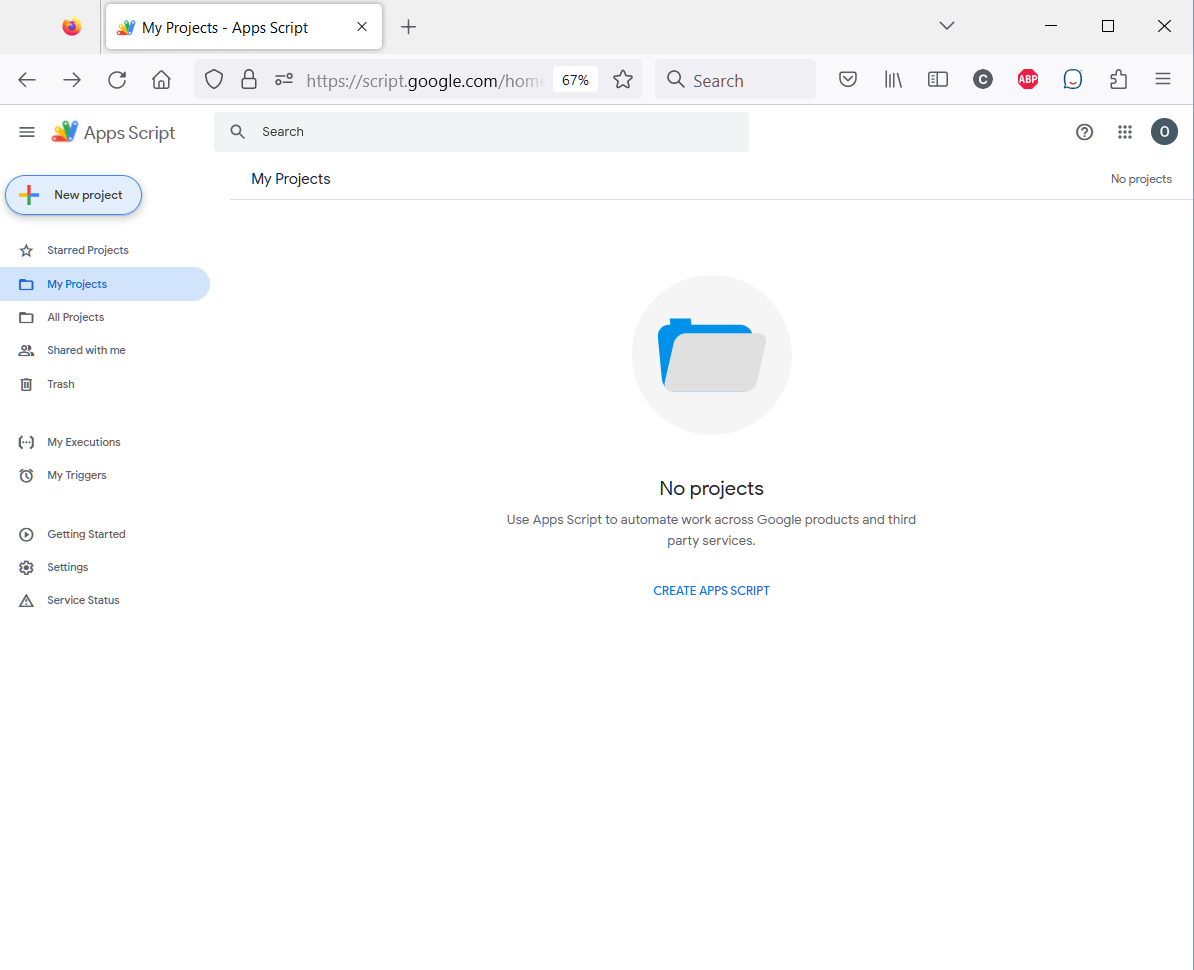
# 1.4 Assign folder IDs to the Fellowship’s Site Settings

* Google Drive Folder ID where config file is located (field configFileFolderIdFellApp)
* Backup Google Spreadsheet ID (field backupFileIdFellApp)

# Create Fellowship Recommendation Letter Submission

# Create Apps Script

* + 1. <https://script.google.com/home>
    2. Click on “CREATE APPS SCRIPT”



* + 1. Copy all files from existing script “fellowship-rec-letter-submission”. These files can be copied from orderflex\src\App\FellAppBundle\Util\GoogleForm\Fellowship-RecommendationLetter
    2. All script can be founded here <https://script.google.com/home>
    3. Alternatively, all Google Api Scripts can be pull from github repository <https://github.com/cinava/fellowship-rec-letter-submission> using a Chrome browser extension “Google Apps Script GitHub Assistant” (see 4 below).

# Create folders in Google Drive to store application’s spreadsheets and uploaded files for the Fellowship Recommendation Letter Submission

All files are located here orderflex\src\App\FellAppBundle\Util\GoogleForm\FellowshipRecommendationLetters

* + 1. Create folder “FellowshipApplication” (the same as in 1.1)
    2. In “FellowshipApplication” folder, create empty file “config.json” (the same as in 1.2)
    3. In “FellowshipApplication” folder, create new folder “Responses” (the same as in 1.3)
    4. In “Responses” folder, create new folders:

RecommendationLetters (the same as in 1.4)

* + 1. In “RecommendationLetters” folder, create new folders:

“RecommendationLetterUploads”

“Template”

“Spreadsheets”

* + 1. In “Template” folder, create new files:

BackupSpreadsheet – copy/paste from orderflex\src\App\FellAppBundle\Util\GoogleForm\ \FellowshipRecommendationLetters\BackupSpreadsheet.xlsx

TemplateSpreadsheet - copy/paste from orderflex\src\App\FellAppBundle\Util\GoogleForm\ \FellowshipRecommendationLetters\TemplateSpreadsheet.xlsx

# Adding IDs to the script ApplicationWithUploadNew

* Assign folder id of the Spreadsheets to \_destinationFolderSSKey
* Assign file id of the TemplateSpreadsheet to \_templateSSKey
* Assign file id of the BackupSpreadsheet to \_backupSSKey
* Assign folder name “RecommendationLetterUploads” to \_dropbox
* Assign file id of the “config.json” to \_configFolderId

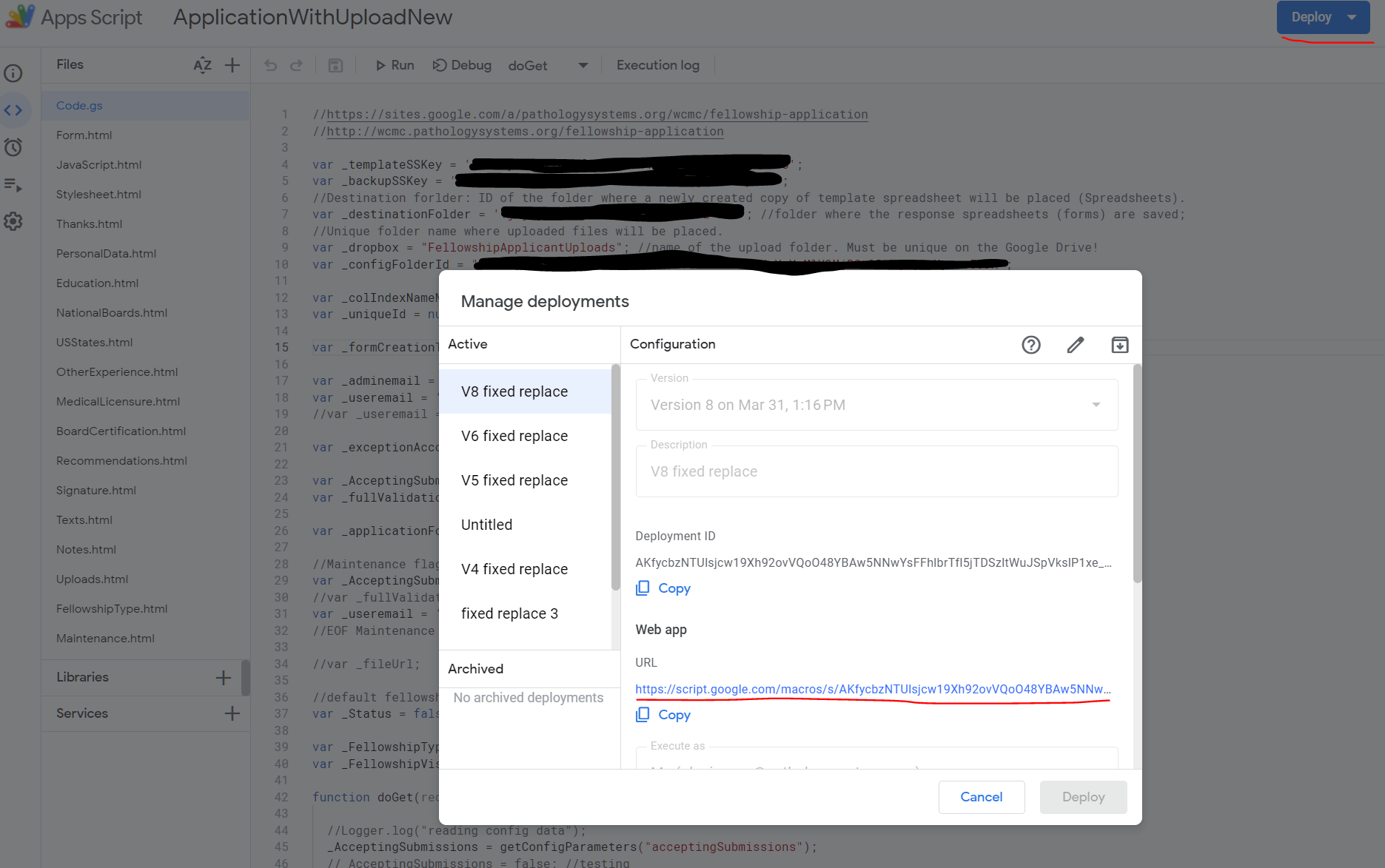
# Assign folder IDs to the Fellowship’s Site Settings

# Create web pages

Google replaced the Classic Sites with New Sites. Creating and editing in classic Sites are no longer supported. As of January 30, 2023: Editing of classic sites is turned off. You can migrate your Classic Sites for a smooth transition. If you don't migrate your Classic Sites, your classic sites will be automigrated to New Sites

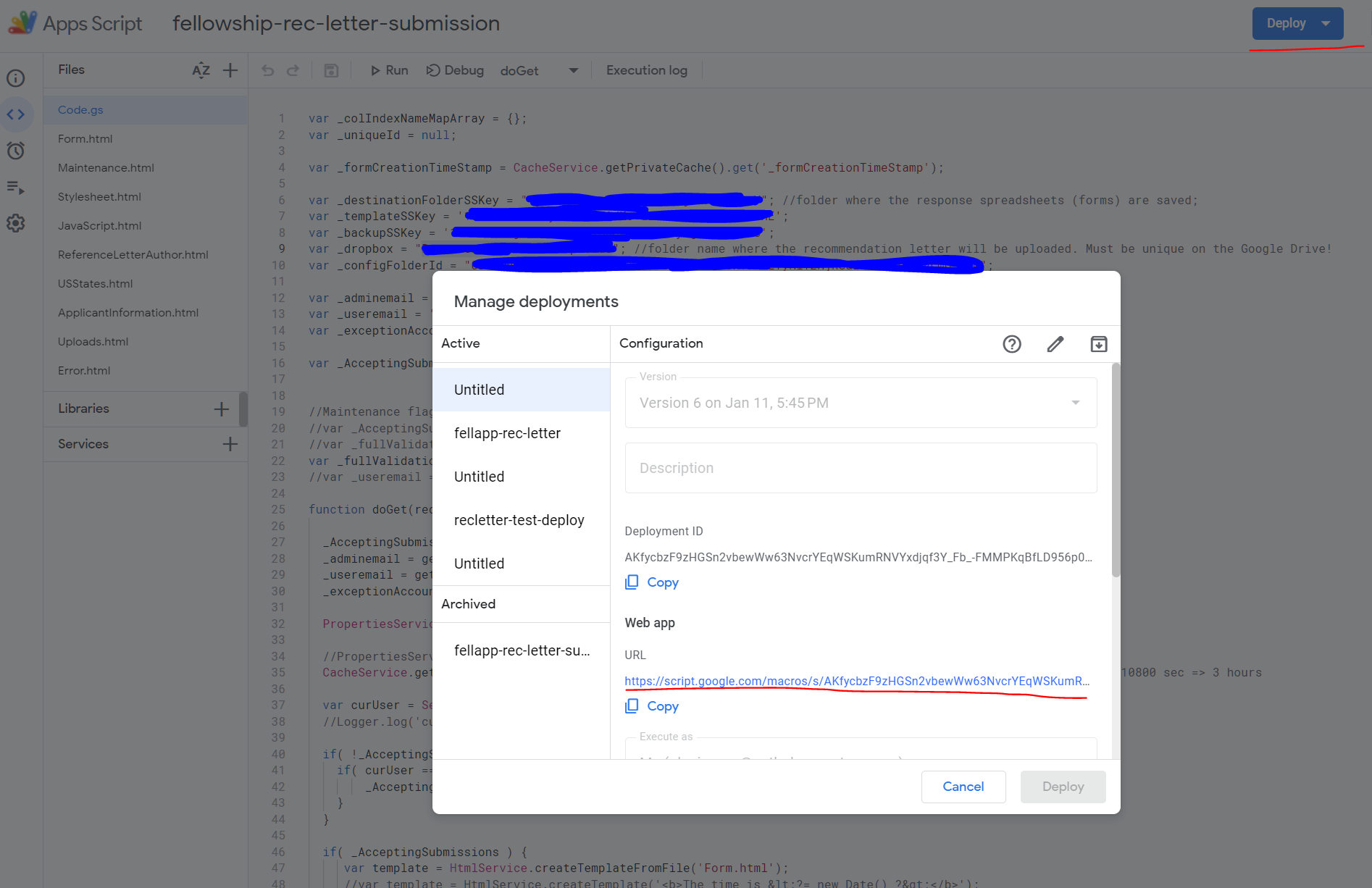
# 3.1 Create web page <http://wcmc.pathologysystems.org/fellowship-application>

* Create a new site: https://sites.google.com/new
* On a new site, in editing mode, click “Embed” and paste the url from “Manage Deployment” on ApplicationWithUploadNew script.
* The script is located here: https://script.google.com/home/projects/1yqHXTkhrWUMBGbC4oK4DSc8XZTikW5vH1DH2tCvexORCHSMfvObmZt0W/edit
* Adjust the width and height of the scrolling window
* The url of the new site will be something like <https://sites.google.com/pathologysystems.org/testsiteforfellapp/home>



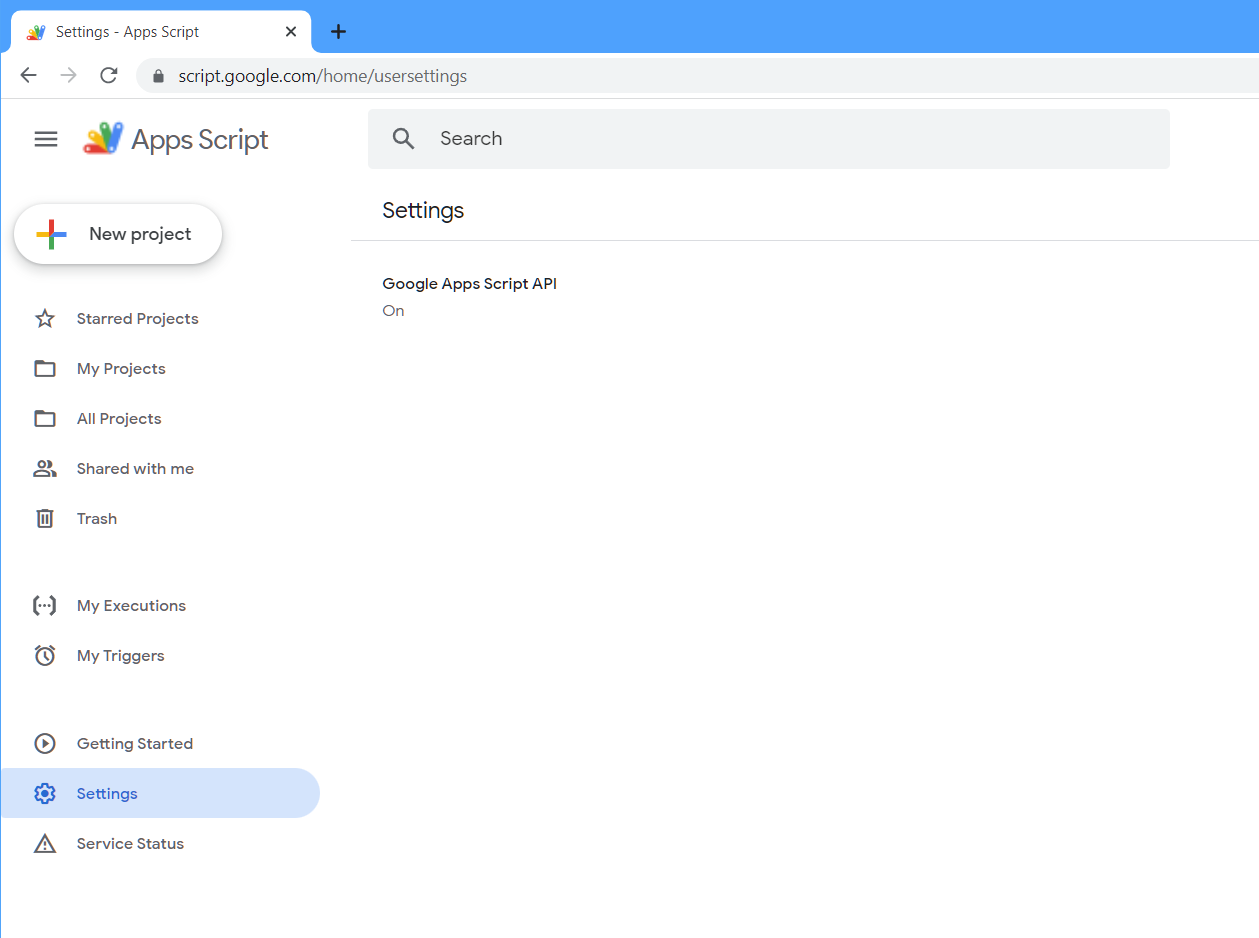
# 3.2 Create web page <http://wcmc.pathologysystems.org/fellowship-application-reference-letter-upload>

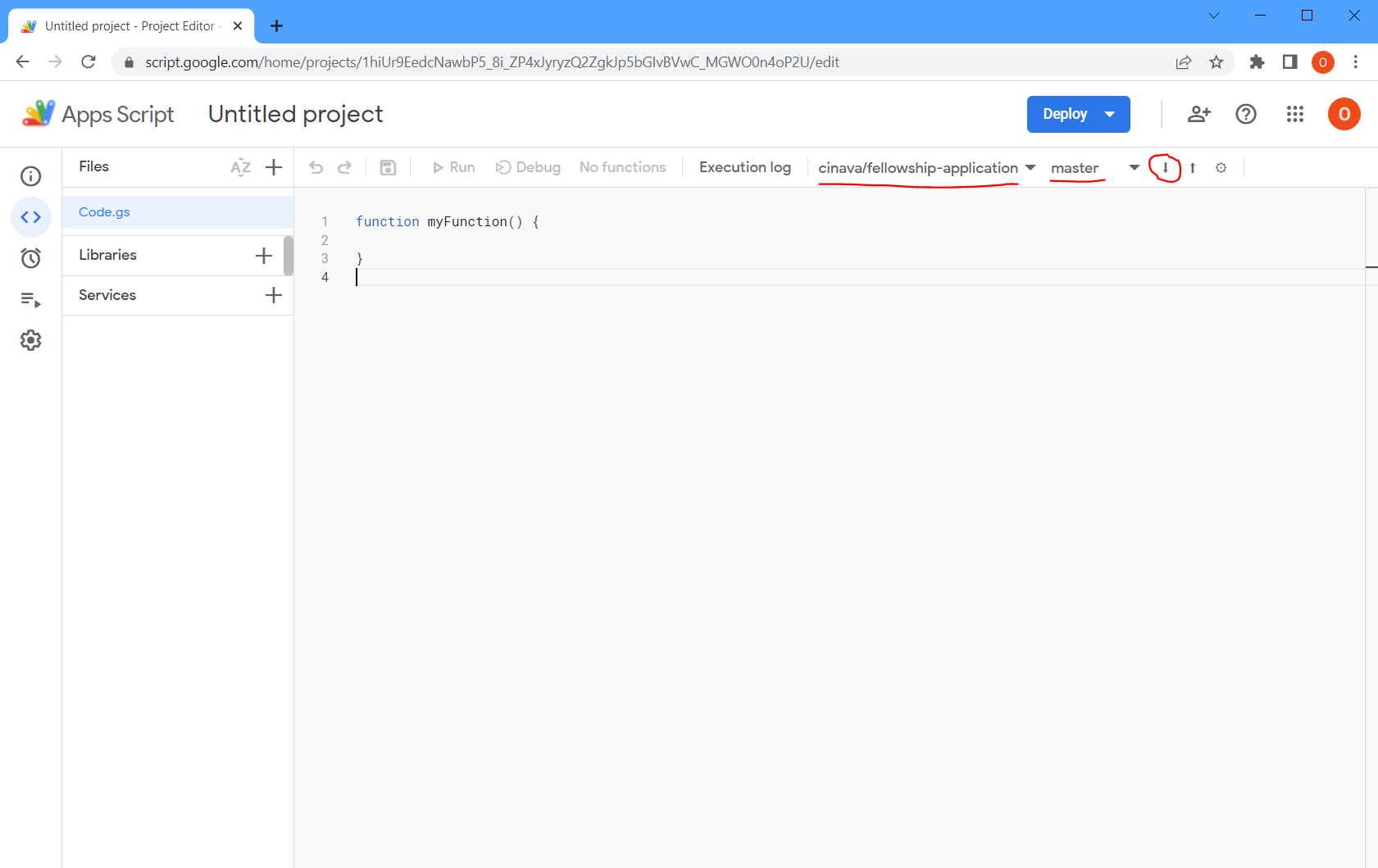
* Create a new site: https://sites.google.com/new
* On a new site, in editing mode, click “Embed” and paste the url from “Manage Deployment” on fellowship-rec-letter-submission script.
* The script is located here: https://script.google.com/home/projects/1bXDXOGx5-hxm3B6OC7lLBpGaseDmi0x8ecddK5r3H2sZKGGrx3bbYhK-/edit
* Adjust the width and height of the scrolling window
* The url of the new site will be something like <https://sites.google.com/pathologysystems.org/testsiteforfellapp/home>



# Chrome browser extension “Google Apps Script GitHub Assistant” to pull/push all Google Api Scripts

Alternatively, all Google Api Scripts can be pull from github repository <https://github.com/cinava/fellowship-application> using a Chrome browser extension “Google Apps Script GitHub Assistant”. In order to pull (arrow down) remote repository, enable the Google Apps Script API: <https://script.google.com/home/usersettings>.





Another possibility is using an open-source tool called clasp (<https://developers.google.com/apps-script/guides/clasp> and https://github.com/google/clasp).

# Authentication by PHP to use Google API to download/upload files on the Google Drive

5.1 Go to <https://console.cloud.google.com/apis/credentials?project=quickstart-1563304352237>

5.2 Choose or create new project

5.3 Click “Create Credentials” and choose “Service Account”

5.4 Follow the steps and create JSON credentials file

5.5 Share Google Drive to user from the generated JSON credentials file

5.6 Update Site Settings’ field p12KeyPathFellApp with the full path to the credentials JSON file