**pg.py – wiki page generator**

This program generates one or more wiki pages from a google sheet.

Usage: python3 pg.py [-i] <GoogleSheetId> [-b <BookName>]

Example: GoogleSheetId is the string “**1iVVOBV0Zi7fgady7Ebhp6XJIBMqMxAFCX-cpOlLrHAY**” in <https://docs.google.com/spreadsheets/d/1iVVOBV0Zi7fgady7Ebhp6XJIBMqMxAFCX-cpOlLrHAY/>

BookName is the name of the scripture. It is used in along with Column A and B to generate a wiki page name.

Organize the google sheet in the following manner to use the page generator as it is. One can modify the code to meet special cases if the data cannot be organized in the structure described below.

* Column **A** contains the **chapter name**
* Column **B** contains the **section name**
* Column **C** contains **verseNumber** and
* Column **D** contains the slokas or **verses**.

The page name is typically BookName+Chapter+’.’+Section. Every row in A, B, C and D should have correct data. If A and B are empty, the wiki page name is generated using the BookName+x+y from the verseNumber to make it unique. It is better to have meaningful names in column A and B.

verseNumber is of the form **of [Chapter.][Section.]verseNo** (x.y.z). verseNumber can be any of the following: z or y.z or x.y.z but must be the same for all rows in the sheet.

It is extremely important that column C is reviewed for **correctness of the numbering convention**. Several edge cases have been taken care of to maintain correctness of the generated pages in the event of bad data. However, any error may affect the page generation and program may crash!

There can be **one or more lines of sloka** (separated by \n) in cell in column C. A sloka can span multiple rows as well. In such cases, Columns A, B and C must contain identical data in the respective rows.

Use only '|' or '||' at the end of the line and NOT others chars like '/' etc. Avoid having empty cells in the sloka column. The code will ignore empty lines in a cell.

New page is created for each section consisting of all verses in rows having the same x.y in verseNumber (x.y.z). **New page is generated when x.y in column C changes.**

An index table of Chapter Pages is added to each page if more than one page (y) is generated from the sheet for any chapter.

A Chapter index table is added to each page if more than one chapter (x) is found in the sheet.

The page generator does not interpret the contents of the sheet. It just looks for the patterns described above.

**Preparations**

* Create a folder named **'verses'** in the folder where the code generator is located. The generated pages are stored in the verses folder.
* Get a google authentication token from Google to access a sheet from google drive.
* Create a bot in the wiki and save the bot password to use in the uploader program. Create bot password from wiki's special pages. If you get ssl error, use the browser in incognito mode or see https://support.mozilla.org/en-US/questions/1222739 . Or install and use chrome browser.
* If you are planning to use pywikibot, you can use the bot password in the following manner:
* Install pywikibot to use the bots on mediawiki
* **pip3 install "requests>=2.20.1"**
* Download the current Pywikibot stable branch https://pywikibot.toolforge.org/core\_stable.tar.gz: .tar.gz, .zip. Unpack the downloaded file, for example in your home or Documents folder and rename it to pywikibot: ~/pywikibot or C:\Users\username\pywikibot.

**cd ~/pywikibot**

**python3 pwb.py generate\_family\_file.py**

Enter local url http://127.0.1.1:8080/wiki/Main\_Page and familyname, say, Kailaasa

**python3 pwb.py generate\_user\_files.py**

enter the previously generated familyname (Kailaasa), botname and botpassword (update user-password.py with botname and botpassword)

Verify that bot can login

**python3 pwb.py login**