Summer Internship Report

On

Development of Windows offline Installer, Ubuntu offline Installer, Analytics Feature, Github Update Feature and Shell based Osdag

Submitted by

Sagar Rathore

Under the guidance of

Prof. Sidhartha Ghosh
Civil Engineering Department
IIT Bombay

Under the Mentorship of

Mr. Danish Ansari
Civil Engineering Department IIT Bombay

August 23, 2022

Acknowledgment

I would like to thank FOSSEE for providing me a platform to work on somDevelopment Work on Github Update Feature ething I am very interested in.
I am thankful to everyone who thought of having and involved in selection process based on screening tasks. I am grateful to be a part of team which promotes open source software. I thank all the Osdag members, who are wonderful mentors and great team. I thank Danish Ansari (Project Research Assistant) who guided me at every point.
In addition, I would like to thank all the people at FOSSEE Team, IIT BOMBAY, who have helped me develop as a professional. It would not have been possible without the kind support and help of many individuals and organizations. I want to extend my sincere thanks to all of them.

Contents

Chapter 1

Introduction

1.1 Osdag Internship

Osdag internship is provided under the FOSSEE project. FOSSEE project promotes the use of FOSS (Free/Libreand Open Source Software) tools to improve quality of education in our country. FOSSEE encourages the use of FOSS tools through various activities to ensure availability of competent free software equivalent to commercial (paid) softwares.

The FOSSEE project is a part of the National Mission on Education through Infrastructure and Communication Technology(ICT), Ministry of Human Resources and Development, Government of India.

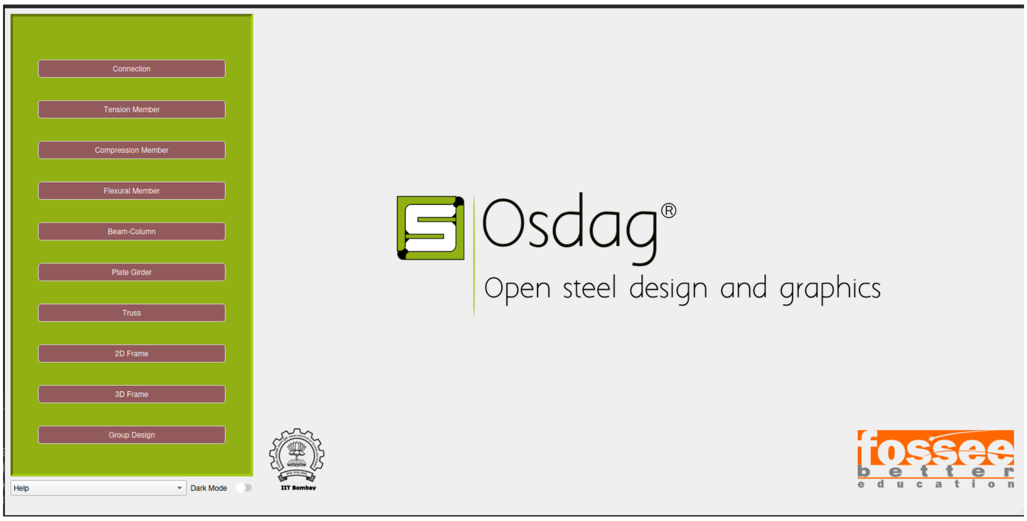
Osdag is one such open source software which comes under the FOSSEE project. Osdag internship is provided through FOSSEE project. Any UG/PG/PhD holder can apply forthis internship, and the selection will be based on a screening task.

Osdag®
Open steel design and graphics

1.2 What is Osdag?

Osdag is a cross-platform free/libre and open-source software for the design (and detailing) of steel structures, following the Indian Standard IS 800:2007. It allows the user to design steel connections, members and systems using a graphical user interface. The interactive GUI provides a 3D visualisation of the designed component and an option to export the CAD model to any drafting software for the creation of construction/fabrication drawings. The design is typically optimised following industry best practices.

Osdag is primarily built upon Python and other Python-based FLOSS tools, such as, PyQt, OpenCascade, PythonOCC, and svgwrite. It uses SQLite for managing steel section databases.



1.3 Who can use?

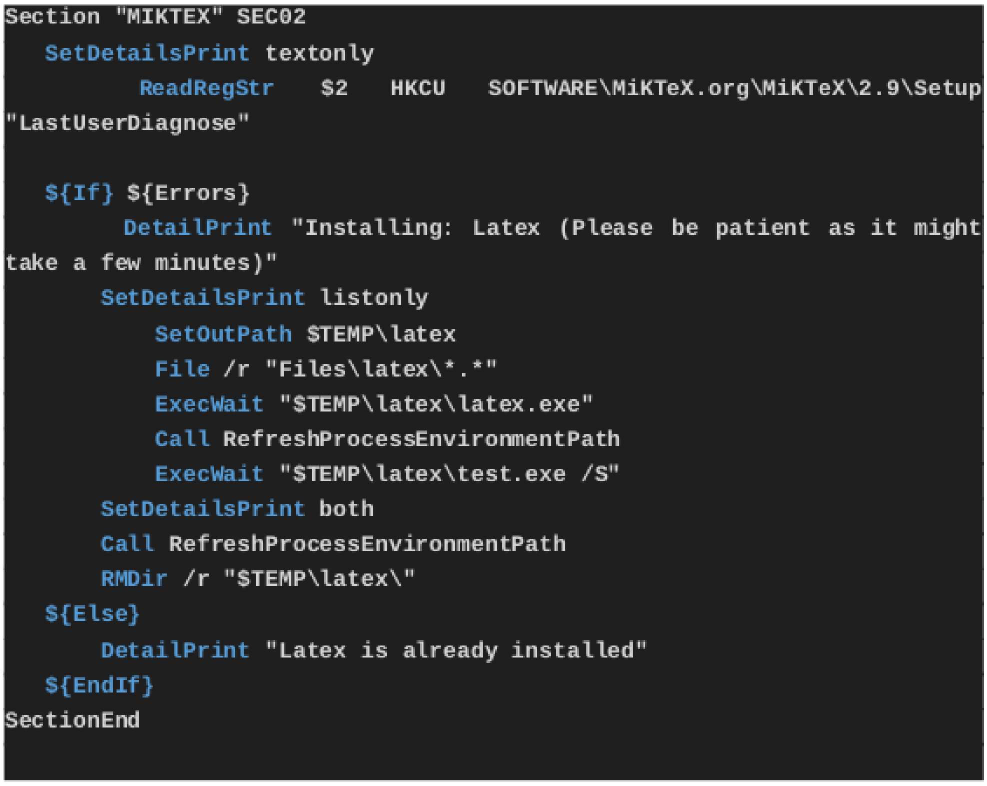
Osdag is created for both educational purpose and industry professionals. As Osdag is currently funded by MHRD, 6 Osdag team is developing software in such a way that it can be used by the students during their academics and to give them a better insight look in the subject. Osdag can be used by anyone starting from novice to professionals. It's simple user interface makes it flexible and attractive than other software. Video tutorials are available to help get started. The video tutorials of Osdag can be accessed here.

Chapter 2

Development work on Windows Installer

2.1 Added Check for MikTex

Added a check for **MikTex**, if it's already installed then installer will not ask for installing it again if ran another time.



```
Section "MIKTEX" SEC02
   SetDetailsPrint textonly
         ReadRegStr   $2   HKCU   SOFTWARE\MiKTeX.org\MiKTeX\2.9\Setup
"LastUserDiagnose"

   ${If} ${Errors}
         DetailPrint "Installing: Latex (Please be patient as it might
take a few minutes)"
       SetDetailsPrint listonly
           SetOutPath $TEMP\latex
           File /r "Files\latex\*.*"
           ExecWait "$TEMP\latex\latex.exe"
           Call RefreshProcessEnvironmentPath
           ExecWait "$TEMP\latex\test.exe /S"
       SetDetailsPrint both
       Call RefreshProcessEnvironmentPath
       RMDir /r "$TEMP\latex\"
   ${Else}
       DetailPrint "Latex is already installed"
   ${EndIf}
SectionEnd
```

2.2 Bug Fixes

Following are some changes and bug fixes are done in windows

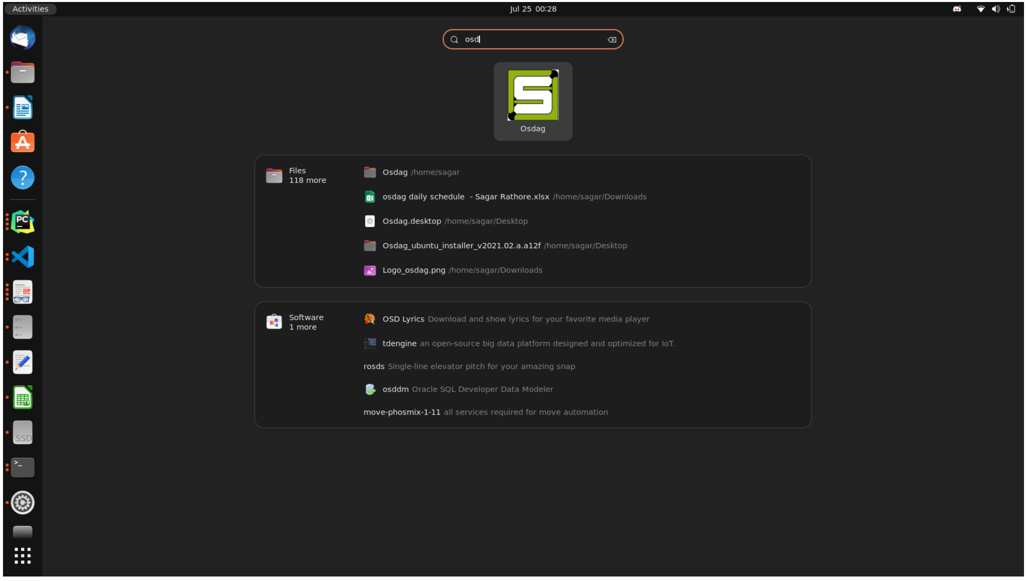
- Changing installation directory from C://Osdag to Program Files/Osdag.
- Removed Unneccasary paths addition for miniconda installation and smoothening the installation procedure.
- Adding a analytical feature which tells number of times Osdag is installed.
- Adding new Conda libraries and changes in setup

Chapter 3

Development on Linux Installer

3.1 Added Desktop Icon in favourites and desktop

Added **Desktop Icon** in favourites and created a shell script to run Osdag from terminal globally by entering **osdag.sh** to enhance **User Experience**.



Files
118 more

Osdag /home/sagar

osdag daily schedule - Sagar Rathore.xlsx /home/sagar/Downloads

Osdag.desktop /home/sagar/Desktop

Osdag_ubuntu_installer_v2021.02.a.a12f /home/sagar/Desktop

Logo_osdag.png /home/sagar/Downloads

Software
1 more

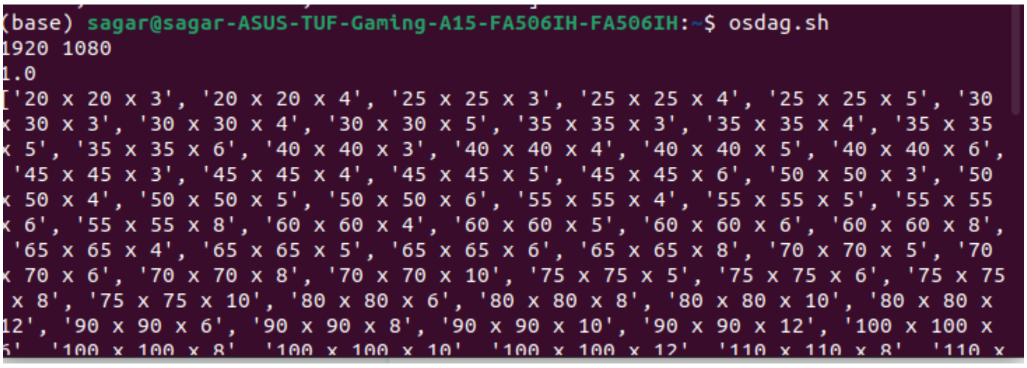
OSD Lyrics Download and show lyrics for your favorite media player

tdengine an open-source big data platform designed and optimized for IoT.

rosds Single-line elevator pitch for your amazing snap

osddm Oracle SQL Developer Data Modeler

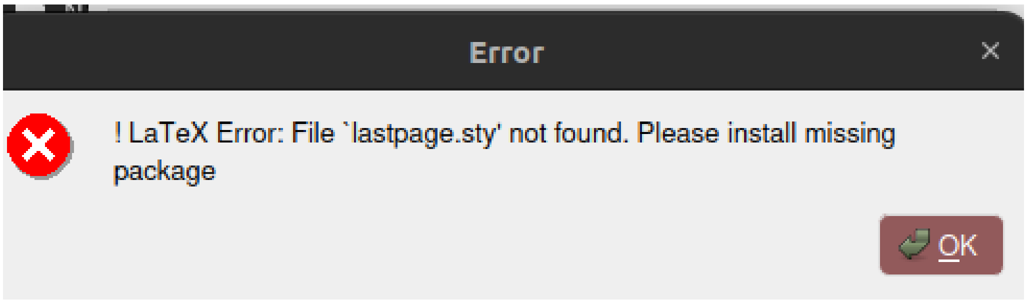
move-phosmix-1-11 all services required for move automation



```
(base) sagar@sagar-ASUS-TUF-Gaming-A15-FA506IH-FA506IH:~$ osdag.sh
1920 1080
1.0
['20 x 20 x 3', '20 x 20 x 4', '25 x 25 x 3', '25 x 25 x 4', '25 x 25 x 5', '30
x 30 x 3', '30 x 30 x 4', '30 x 30 x 5', '35 x 35 x 3', '35 x 35 x 4', '35 x 35
x 5', '35 x 35 x 6', '40 x 40 x 3', '40 x 40 x 4', '40 x 40 x 5', '40 x 40 x 6',
 '45 x 45 x 3', '45 x 45 x 4', '45 x 45 x 5', '45 x 45 x 6', '50 x 50 x 3', '50
x 50 x 4', '50 x 50 x 5', '50 x 50 x 6', '55 x 55 x 4', '55 x 55 x 5', '55 x 55
x 6', '55 x 55 x 8', '60 x 60 x 4', '60 x 60 x 5', '60 x 60 x 6', '60 x 60 x 8',
 '65 x 65 x 4', '65 x 65 x 5', '65 x 65 x 6', '65 x 65 x 8', '70 x 70 x 5', '70
x 70 x 6', '70 x 70 x 8', '70 x 70 x 10', '75 x 75 x 5', '75 x 75 x 6', '75 x 75
 x 8', '75 x 75 x 10', '80 x 80 x 6', '80 x 80 x 8', '80 x 80 x 10', '80 x 80 x
12', '90 x 90 x 6', '90 x 90 x 8', '90 x 90 x 10', '90 x 90 x 12', '100 x 100 x
5', '100 x 100 x 8', '100 x 100 x 10', '100 x 100 x 12', '110 x 110 x 8', '110 x
```

3.2 Fixed Latex Installation Error

Fixed error by adding following command in **3-install-texlive.sh**.



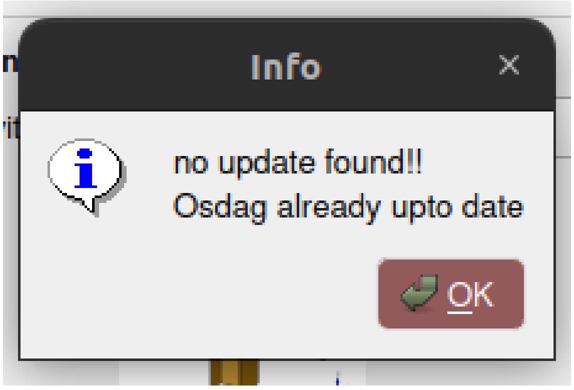
! LaTeX Error: File `lastpage.sty' not found. Please install missing package

sudo apt install texlive-latex-extra

Chapter 4

Development Work on Github Update Feature

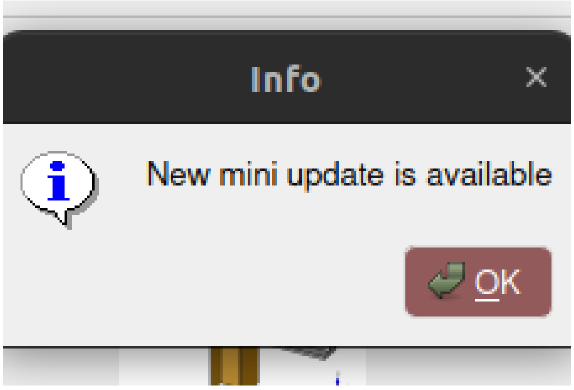
This features checks if there is a major or minor update of Osdag. If it's minor means there is commit on github. When we click on it, if there is no newest commit then it will be showing this message. If a new commit takes place then this message will be shown



Info

no update found!!
Osdag already upto date

If a new commit takes place then this message will be shown



Info

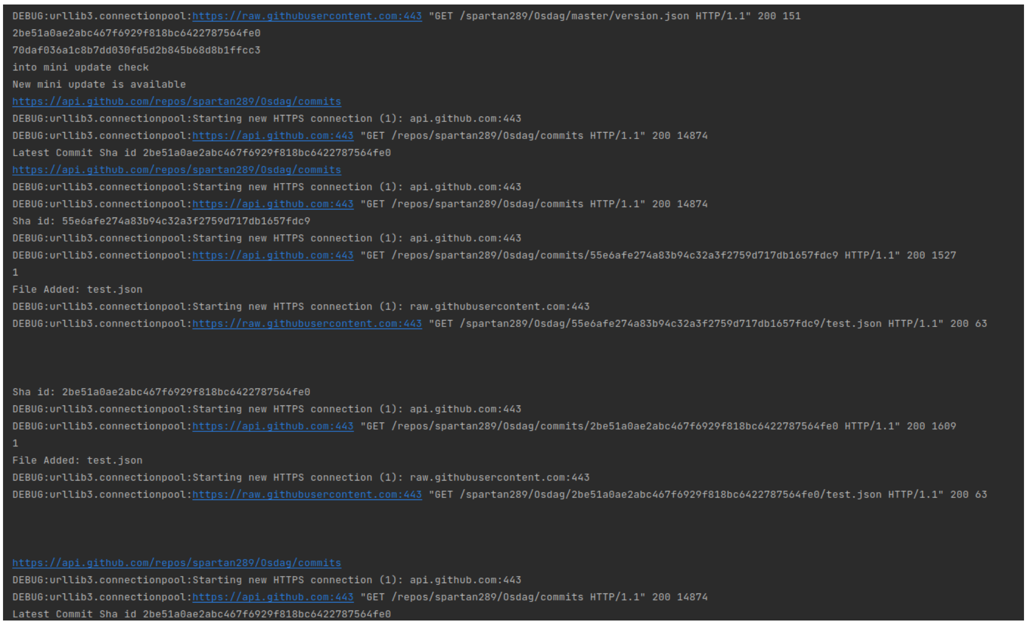
New mini update is available

When clicking on Ok it will update the osdag software by fetching files from Osdag Repository and do following operation:

1. Add a file (if new files added in repo)
2. Update a file (if a file is modified in repo)
3. Delete a file (if a file is deleted in repo)

To use it in the Osdag software with Osdag Repository as main source. One have to do some changes.

Base url is defined for github repo have “spartan289” id changing it to ”osdag-admin” and also changing the current commit id in version.json to currently commited id. This will set Osdag Repository as main from where latest commits. The Source file is here .



```
DEBUG:urllib3.connectionpool:https://raw.githubusercontent.com:443 "GET /spartan289/Osdag/master/version.json HTTP/1.1" 200 151
2be51a0ae2abc467f6929f818bc6422787564fe0
70daf036a1c8b7dd030fd5d2b845b68d8b1ffcc3
into mini update check
New mini update is available
https://api.github.com/repos/spartan289/Osdag/commits
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): api.github.com:443
DEBUG:urllib3.connectionpool:https://api.github.com:443 "GET /repos/spartan289/Osdag/commits HTTP/1.1" 200 14874
Latest Commit Sha id 2be51a0ae2abc467f6929f818bc6422787564fe0
https://api.github.com/repos/spartan289/Osdag/commits
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): api.github.com:443
DEBUG:urllib3.connectionpool:https://api.github.com:443 "GET /repos/spartan289/Osdag/commits HTTP/1.1" 200 14874
Sha id: 55e6afe274a83b94c32a3f2759d717db1657fdc9
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): api.github.com:443
DEBUG:urllib3.connectionpool:https://api.github.com:443 "GET /repos/spartan289/Osdag/commits/55e6afe274a83b94c32a3f2759d717db1657fdc9 HTTP/1.1" 200 1527
1
File Added: test.json
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): raw.githubusercontent.com:443
DEBUG:urllib3.connectionpool:https://raw.githubusercontent.com:443 "GET /spartan289/Osdag/55e6afe274a83b94c32a3f2759d717db1657fdc9/test.json HTTP/1.1" 200 63



Sha id: 2be51a0ae2abc467f6929f818bc6422787564fe0
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): api.github.com:443
DEBUG:urllib3.connectionpool:https://api.github.com:443 "GET /repos/spartan289/Osdag/commits/2be51a0ae2abc467f6929f818bc6422787564fe0 HTTP/1.1" 200 1609
1
File Added: test.json
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): raw.githubusercontent.com:443
DEBUG:urllib3.connectionpool:https://raw.githubusercontent.com:443 "GET /spartan289/Osdag/2be51a0ae2abc467f6929f818bc6422787564fe0/test.json HTTP/1.1" 200 63



https://api.github.com/repos/spartan289/Osdag/commits
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): api.github.com:443
DEBUG:urllib3.connectionpool:https://api.github.com:443 "GET /repos/spartan289/Osdag/commits HTTP/1.1" 200 14874
Latest Commit Sha id 2be51a0ae2abc467f6929f818bc6422787564fe0
```

Chapter 5

Development Work on Shell based Osdag

The current Osdag offers only GUI based system for designing structures. This software aims to create a design without the need for a GUI-based system and do it via Command Line / Shell/ Script.

The main idea is to develop a command line based and to use it to develop single designs and multiple designs in one go.

Currently it develops only a single design at a time and can create design report for 2 connections. You can see the demo here.

The implementation is divided into 3 steps

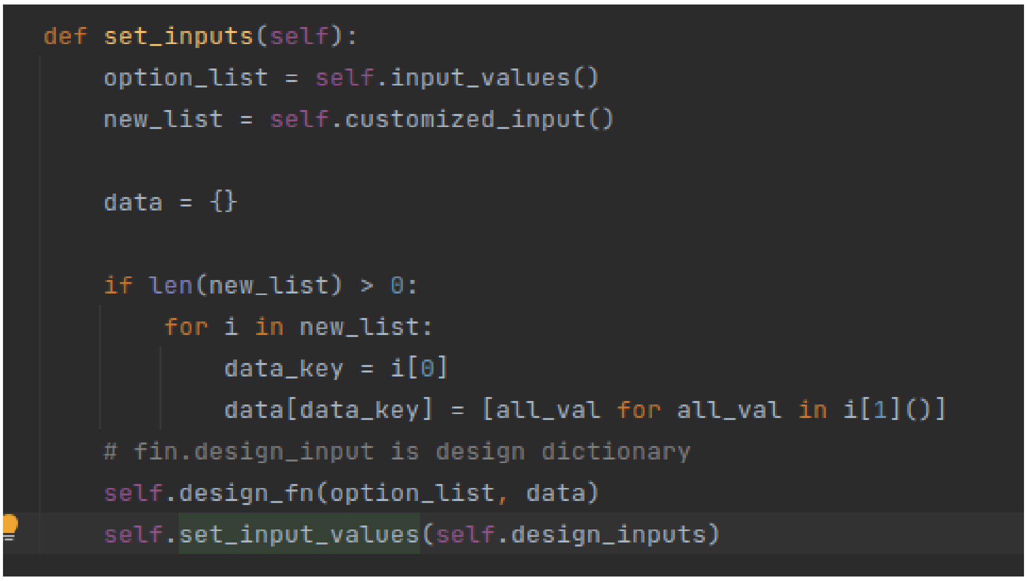
1. Collecting Inputs (same as shown in Input Dock)
2. Computing Data
3. Saving Output

5.0.1 Collecting Inputs

For Collecting Inputs, we need to observe what are inputs shown in input dock and design preference dialog box (Currenty for Preference Dialog Box is not implemented),and taking them via command line and add them in the input dictionary.

Every time a design is created there is an **design dictionary** created which contains all information about the design.

In Finplate.py there is a function designfn() which creates the design dictionary by combining the inputs entered by User and additional information.



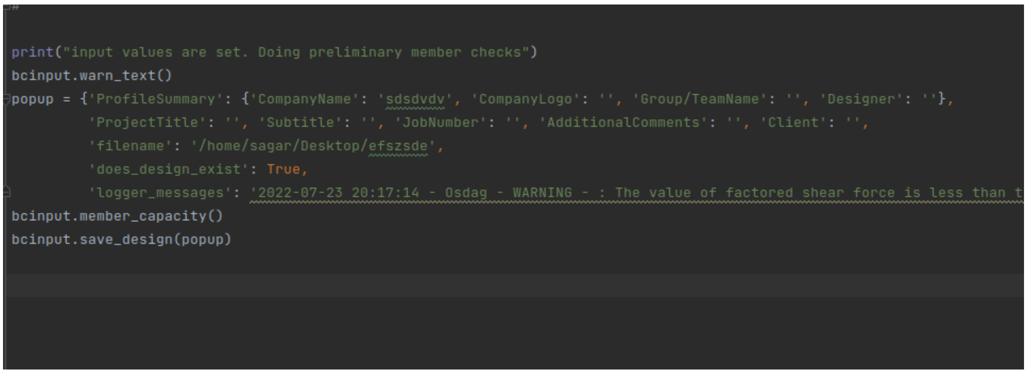
```
def set_inputs(self):
    option_list = self.input_values()
    new_list = self.customized_input()

    data = {}

    if len(new_list) > 0:
        for i in new_list:
            data_key = i[0]
            data[data_key] = [all_val for all_val in i[1]()]
    # fin.design_input is design dictionary
    self.design_fn(option_list, data)
    self.set_input_values(self.design_inputs)
```

5.0.2 Computing Data

For each connections there is a method **setinputvalues()** this method takes the design dict. as a parameter and computes all the data. (Method is included in Parent Class of Connections)

5.0.3 Saving, Showing Output

Similary there is a method for saving design report in each connections. The savedesign() method it takes popup summary [Type: Dict] as parameter.



```
#

print("input values are set. Doing preliminary member checks")
bcinput.warn_text()
popup = {'ProfileSummary': {'CompanyName': 'sdsdvdv', 'CompanyLogo': '', 'Group/TeamName': '', 'Designer': ''},
         'ProjectTitle': '', 'Subtitle': '', 'JobNumber': '', 'AdditionalComments': '', 'Client': '',
         'filename': '/home/sagar/Desktop/efszsde',
         'does_design_exist': True,
         'logger_messages': '2022-07-23 20:17:14 - Osdag - WARNING - : The value of factored shear force is less than t
bcinput.member_capacity()
bcinput.save_design(popup)
```

Notes

1. For every connection/te nsion member in designType folder calling a method is redefined as self argument was also passed in there (and an error was showing) so that's why there is different designType folder in commandline folder where self argument is not included

2. In designfn method for prefernce dialog values are not included and default values are taken for setting it up, we must have to take those parameters as well from user and redo some changes in designfn.

3. Log Output are not defined in Design Report.

4. Debugging uiTemplate.py file can help in identify designDictionary.

Chapter 6

Conclusion

On the whole, I had a great internship experience. It is my first internship and i am so grateful to be a part of it, working on an open source software gave me quite some experience. I have learnt many things along the path and helped me gain knowledge about working in a big project and also helped in gaining new skills which i will use along the path.

My internship has proven to be satisfactory and it has allowed as an opportunity to get an exposure of the practical implementation of theoretical fundamentals.

1. Osdag
2. Python and OOPs
3. Latex
4. Git and Github
5. Nsis Scripting
6. Batch/ VBscript
7. Bash scripting, Ubuntu directories

Reference

- stackoverflow.com
- NSIS Documentation
- Various articles form Medium.com, TowardsDataScience