Summer Fellowship Report

On

Developing Osdag Section Modeller module, Qt Template and Database Version Controlling

Submitted by

Satyam Singh Niranjan

Under the guidance of

Prof.Sidhartha Ghosh
Civil Engineering Department
IIT Bombay

Under the Mentorship of

Deepthi Reddy
Project Research Associate
Danish Ansari
Assistant Project Manager

July 1, 2020

Acknowledgment

I would like to thank FOSSEE for providing me a platform to work on something 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 Sourabh Das (Project Research Associate), Ajmal Babu MS (Project Research Associate), Danish Ansari (Project Research Assistant), Yash Lokhande (Project Research Assistant), Darshan Viswakarma (Project Research Associate), Anand Swaroop (Project Research Associate), Anjali Jatav (Project Research Assistant) and whole team, who made us feel welcome and planned all the tasks meticulously during this period.

I am grateful that I got a chance to work under Prof. Sidharth Ghosh, who took time to mentor us and monitored individual contributions as well.

Contents

Chapter 1

Introduction

1.1 Osdag Internship

Osdag internship is provided under the FOSSEE project. FOSSEE project promotes the use of FOSS (Free/Libre and 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 for this internship. And the selection will be based on a screening task.

1.2 What is Osdag?

Osdag is Free/Libre and Open Source Software being developed for design of steel structures. Its source code is written in Python, 3D CAD images are developed using PythonOCC. Github is used to ensure smooth workflow between different modules and team members. It is in a path where people from around the world would be able to contribute to its development. FOSSEE's "Share alike" policy would improve the standard of the software when the source code is further modified based on the industrial and educational needs across the country.

Design and Detailing Checklist (DDCL) for different connections, members and structure designs is one of the important bi-products of this project. It would create a repository and design guide book for steel construction based on Indian Standard codes and best industry practices.

1.3 Who can use ?

Osdag is created both for educational purpose and industry professionals. As Osdag is currently funded by MHRD, 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

Navigation Window Templatization

I have added a dictionary-parsing code which when provided with a dictionary with certain values will create the GUI using that information. This creates the Left-side Module Navigation buttons and connect these buttons to the respective Tab and further, sub-tabs. Also, Each tab/sub-tab can include any number of modules/sub-modules respectively. All modules will have a respective radio button and an image, which when clicked selects that particular module.On clicking the start button, finally, the respective module is launched in a separate window and the navigation window disappears. Checkout the Code here

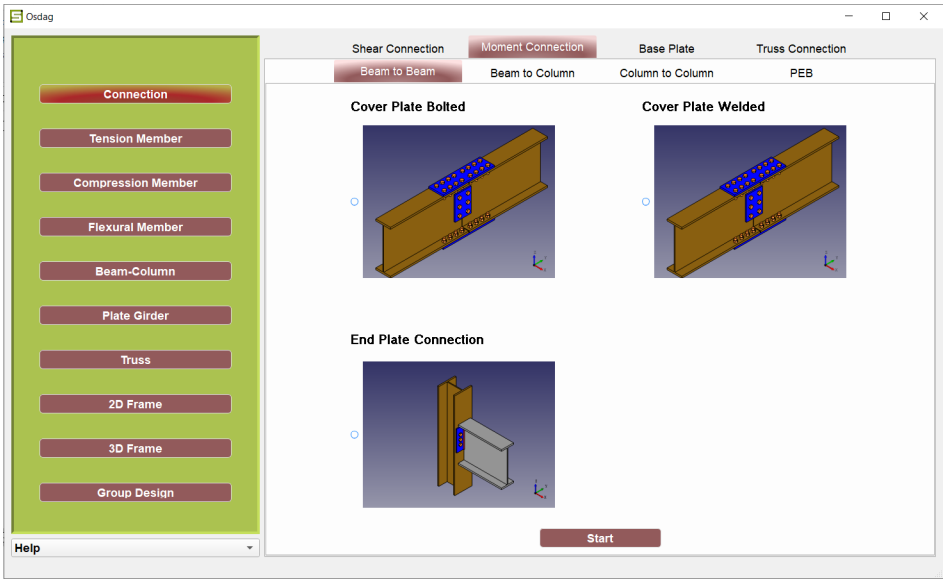


Figure 2.1: Navigation Window

2.1 3-Level Dictionary-Parsing

A pre-created variable is assigned atmost a 3-level dictionary.Now, there are 3 kinds of values allowed:

1. Dictionary(tab/sub-tab)
2. List/Tuple(modules)
3. Under Development Variable(under development notice)

```
'''
   INSTRUCTIONS TO USE OSDAG MAIN PAGE TEMPLATE(OsdagMainWindow):
-----------------------------------------------------------------------------------------------
   Note: This code is designed to handle a 3 level structure, e.g. ,


.                                                                                               .
.                          Modules (Main Dictionary)                                            .
.        ___________________|_____________________________________________.........              .   ((LEVEL 1))
.       |          |         |          |                  |             |                       .
.    Module_1    Module_2   Module_3    Module_4          Module_5     Module_6        (Keys of Main Dictionary)
........|...........|...........|..........|..................|.............|.........................
        |           |           |          |                  |             |
        |           |           |          |                  |             |
        |           |           | [List/Tuple of Module Variants]|         |
        |           |           |          |                  |             |
        |           |    [List/Tuple of Module Variants]      |             |
        |      (UNDER DEVELOPMENT)                        (UNDER DEVELOPMENT)   |
........|.....................................................................|.........................
.       |___|_____________________________.....                 _____________|_____.....     (Sub Dictionaries)
.       |                 |             |                      |             |                 .    ((LEVEL 2))
.    Submodule_1    Submodule_2    Submodule_3          Submodule_1    Submodule_2    Submodule_3    (Keys of Sub Dictionaries)
.       |             |             |                      |             |                       .
........|.............|.............|......................|.............|.......................
        |             |             |                      |             |
        |             |             |                      |             |
        |             |             |                      |             |
        |             |      (UNDER DEVELOPMENT)           |             |
        |             |                  [List/Tuple of Module Variants]  |
        |  [List/Tuple of Module Variants]                                 |  [List/Tuple of Module Variants]
        |                                                                  |
........|.....................................................................|.........................
.       |___|_____________________________.....                 _____________|_____.....     (Sub-Sub Dictionaries)
.       |                 |             |                      |             |                 .    ((LEVEL 3))
. Sub-Submodule_1  Sub-Submodule_2  Sub-Submodule_3  Sub-Submodule_1  Sub-Submodule_2  Sub-Submodule_3 (Keys of Sub-Sub Dictionaries)
.       |             |             |                      |             |                       .
........|.............|.............|......................|.............|.......................
        |             |             |                      |             |
        |    (UNDER DEVELOPMENT)    |    [List/Tuple of Module Variants]  |    [List/Tuple of Module Variants]
        |                  [List/Tuple of Module Variants]     [List/Tuple of Module Variants]
[List/Tuple of Module Variants]
```

Figure 2.2: 3-Level Dictionary Structure Example

The 1st level will contain keys as names of left-side module navigation buttons. The values of these keys can be either of the three values mentioned earlier.
The 2nd level is created on passing a dictionary as the value of 1st level dictionary key(s). The values of these 2nd level dictionary keys can be either of the three values mentioned earlier.
The 3rd level is created on passing a dictionary as the value of 2nd level dictionary key(s). The values of these 3nd level dictionary keys can be either List/tuple or under development variable.

The dictionary is then parsed through the code and both the front-end and back-end are created through code, saving the trouble of hardcoding everything and making a fuss on trying to make changes.

2.2 Modifying Existing Module Opening Methods

The list/tuple value on the 3-level dictionary will contain sub-lists/sub-tuples with 3 values each:

1. The Name of the Module
2. The location of the module image
3. Name of the radio button object for that module

Also, the last value of the main list/tuple will be a function that connects the start button on the respective tab to the respective module window; e.g.

'Beam to Beam' :[
('Cover Plate Bolted','images/coverplate.png','B2BCPBolted'),
('Cover Plate Welded','images/coverplate.png','B2BCPWelded'),
('End Plate Connection','images/endplate.png','B2BEPConnection'),
self.showmomentconnection,
]

Now, the function provided as the last element of the list/tuple should have conditions for searching the radio button object with the name provided and open the respective module window.

Chapter 3

Database Version Controlling

As far as git is concerned, database tables are just big binary blobs, and git has no way of tracking differences between versions in that case. You can track changes in your database by exporting your tables as flat files, and keeping those under git control. So, there's no optimal/ideal way of version controlling Databases with git rather than using commercial software like Redgate's SQL source control (Allows version controlling a database with your choice of VCS like git, TFS, etc). So to this was achieved by storing a SQL file in the repository instead of the database file itself. Since SQL files are flat file, thus version controlling was automatically done for them. The difficult task was keeping the database as well as SQL file updated on changes in one another.This whole concept is based on one single property that is the 'Date/Time modified' of the SQL or Database file. Checkout the Code here

3.1 Pre-Build Database Updation/Creation

A check is built into the code, to be performed on each run of the program. This check searches for the database. This is all done in lieu of maintaining the database on fresh install/pull or a consecutive pull. The code for this can be found in OsdagMainPage.py under "Pre-Build Database Updation/Creation"

- **If the database doesn't exist (Database Creation):**
  Then with the help of the SQL file, the database is created and the SQL file is touched, making its last modified date/time slightly higher(in milliseconds) than the database file.

- **If the database exists (Database Updation):**
  Then a check is performed to compare the last modification date/-time of the SQL and SQLite file(database). If the last modification date/time of SQL file minus one is more recent than that of SQLite file(database), then:

  1. The SQL file is used to create an updated database with a new name.
  2. If the operation (1) doesn't throw any error then, the original database is removed and the new one is named as the old one.
  3. The SQL file is touched, making its last modified date/time slightly higher(in milliseconds) than the database file.
  4. If an error is thrown then, any residue of the corrupt database created in the operation (1) is removed.

3.2 Post-Exit SQL file Updation

A check is built into the code, to be performed on each exit of the program. This check compares the last modified date/time of the SQL and the SQLite file(database) minus one. This is all done in lieu of maintaining the SQL (source code) file. If the SQLite file(database) is more recent than the SQL file, then another database dump is created and overwritten on the SQL file, thus updating the SQL file.

Chapter 4

LaTex Report Changes and Bug/Error Fixes

The LaTex report created and saved as Design report threw errors on certain circumstances and had some minor bugs which disabled the smooth and continuous run of the application. Most errors were minor and were handled with an error dialog displayed to the user.

4.1 LaTex Compilation Errors

One of the LaTex compilation error was thrown when the a file with the same name if already open , with which name the user is trying to save. Because since PDF viewers generally lock the PDF files from being edited when opened using them. Thus editing the file when it was open wasn't possible. This error was detected by searching if the log file created after error had this particular error in it. If this error was found in the log file then, the a respective Error dialog was displayed to the user. All other errors were handled the same, by asking the user to send the log file created to the developers to investigate the compilation error. Checkout the Code here

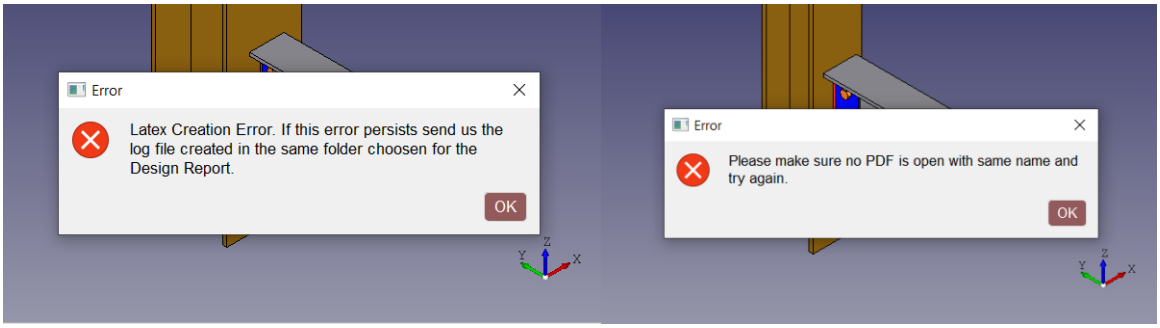


Figure 4.1: LaTex Error Dialog

4.2 Oddly-sized Image Bug for LaTex Report

In small resolutions and improper-sized window the screenshot taken of the OCC viewer for attaching in the LaTex design report was oddly-sized. To solve this bug:
A Screenshot is taken and saved.Checkout the Code here

- **If the screenshot is sized higher than a certain size:** Then, the screenshot is kept and report is created.
- **If the screenshot is sized lower than a certain size:** Then,
  1. The previous screenshot is deleted.
  2. The logger,Input and output docks are hidden.
  3. A screenshot is taken.
  4. The logger,Input and output docks are shown back.

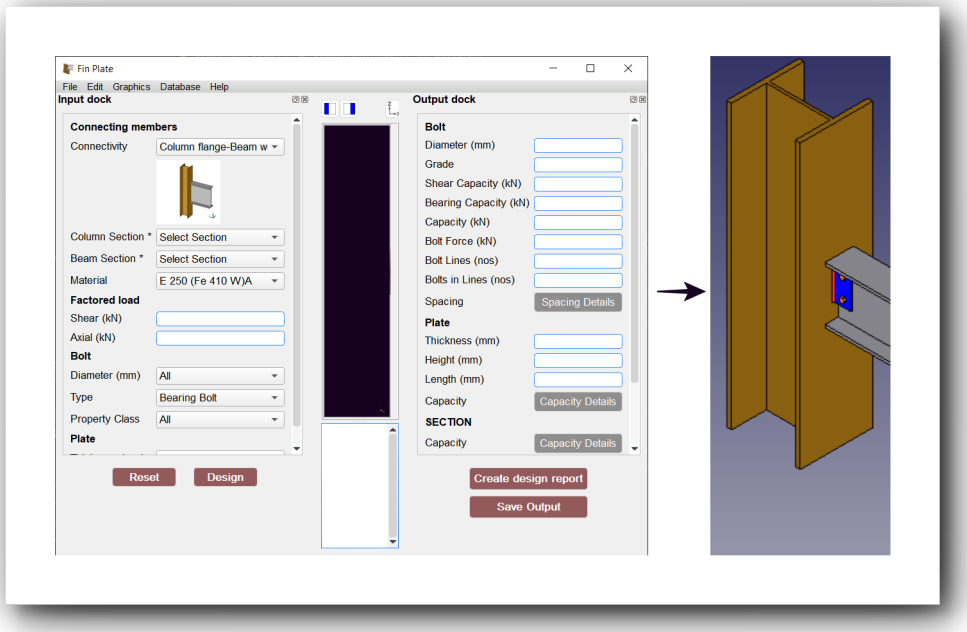


Figure 4.2: OCC Viewer Screenshot Bug

Chapter 5

Osdag Section Modeller Creation

Osdag Section Modeller is a new feature dialog that helps design,visualize and save Sections to be further used in the Main Application.The Section Modeller has 3 sections namely:

1. Define Section
2. CAD Viewer
3. Section Properties

Checkout the Code here

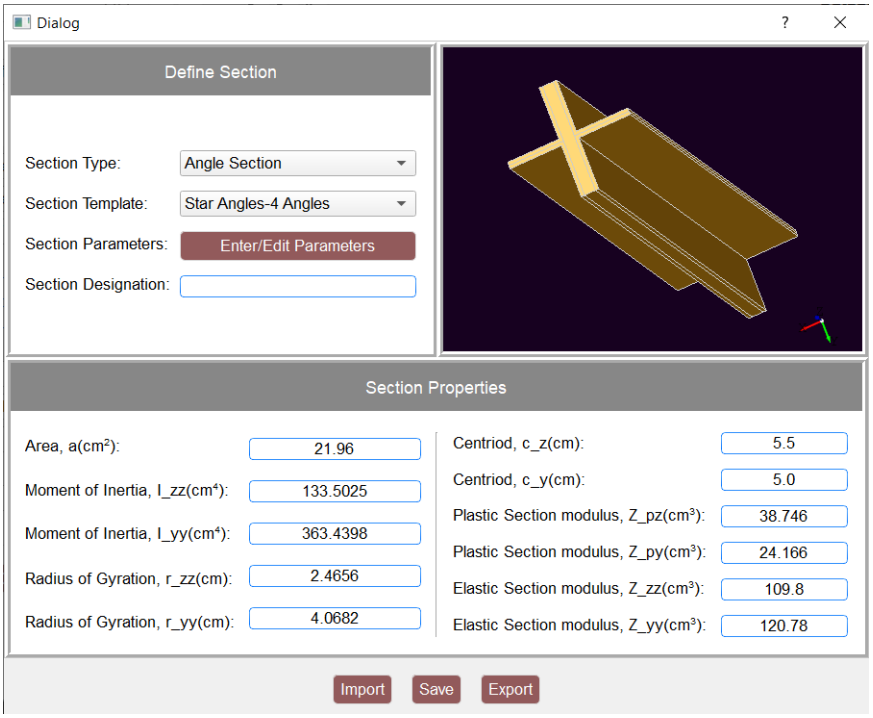


Figure 5.1: Osdag Section Modeller

5.1 Define Section with Inputs

The Define section is where the user enters/selects the required parameters.It has 4 Inputs:

1. Section Type
2. Section Template
3. Section Parameters
4. Section Designation

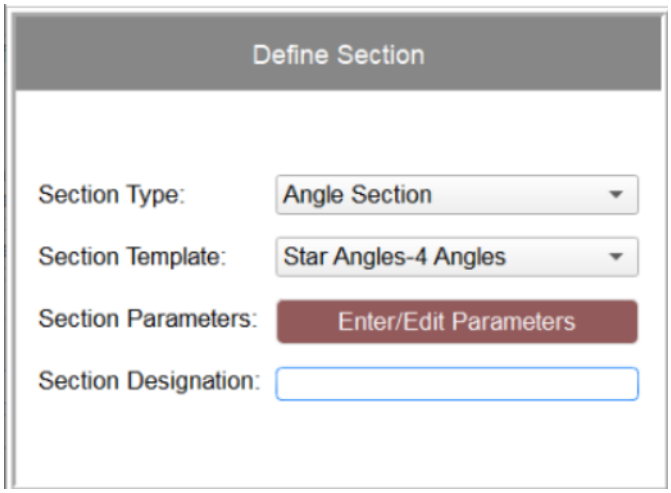


Figure 5.2: Define Section

There are 5 Types of section out of which user can select one. On selection of a type, the Section template drop down is automatically updated with the available templates for the selected section.Next, the user show enter the Section Parameters by clicking the 'Enter/Edit Parameters' button.This opens a separate dialog which has the required parameters for the selected template. As soon as the user clicks save on the Section Parameters dialog and if the parameters are all valid, then a CAD model is created and displayed in the CAD viewer and all the Section Properties are updated and displayed in their respective text boxes.

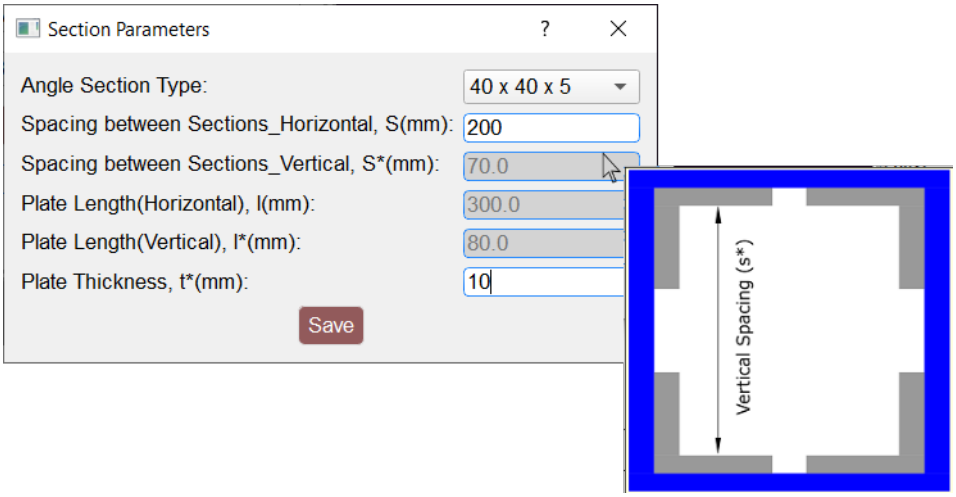


Figure 5.3: Section Parameters Dialog

5.2 Coded Formulas and Attached them to Section Properties

Coded formulas from manual calculations provided by fetching data from the user as well as the database.These formulas were then used to update the respective Section Property in the Section Properties Section.Each time a user saves the Section parameter, the Section properties are updated if the parameters entered are valid.

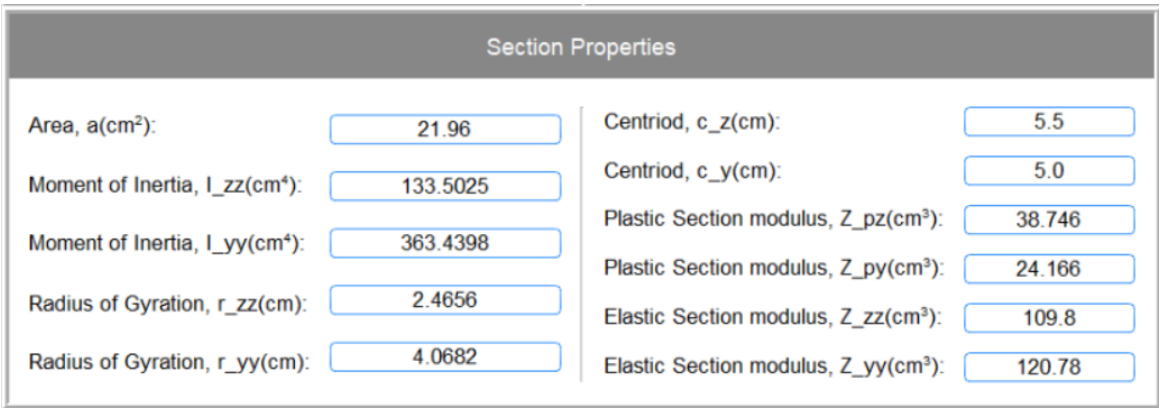


Figure 5.4: Section Properties

5.3 OCC viewer with Respective CAD model

Respective CAD model creation and display in OCC Viewer coded with input and database parameters for each model. Each time a user saves the Section parameter, respective CAD model based on parameters is created and displayed if the parameters entered are valid.

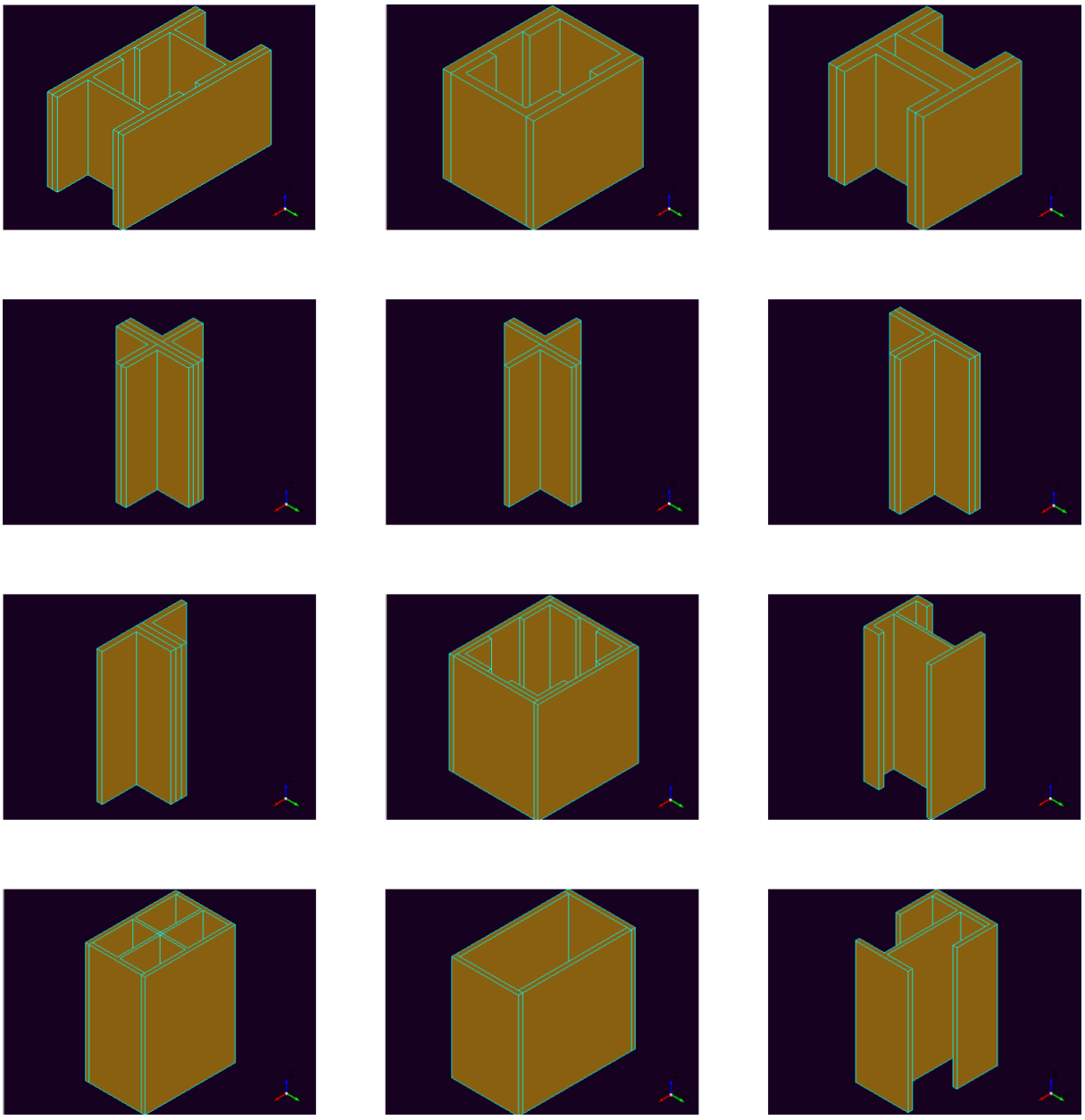
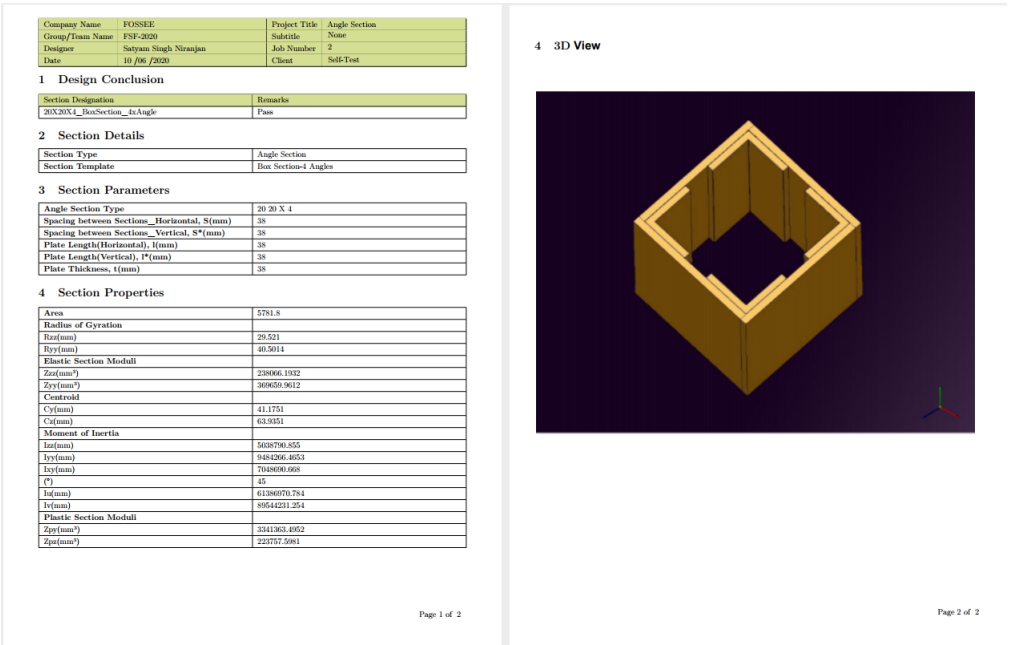


Figure 5.5: CAD Model Examples

5.4 Import,Export and Save Features

The Import feature helps import previously saved Sections from the location of choice in the system.
The Export feature allows the creation of a LaTex formatted Design report of the Designed Section.
The Save feature helps save Designed Sections into .osm file to be used further in the Application or the Modeller itself.



Company Name	FOSSEE	Project Title	Angle Section
Group/Team Name	FSF-2020	Subtitle	None
Designer	Satyam Singh Niranjan	Job Number	2
Date	10 /06 /2020	Client	Self-Test

1 Design Conclusion

Section Designation	Remarks
20X20X4_BoxSection_4xAngle	Pass

2 Section Details

Section Type	Angle Section
Section Template	Box Section-4 Angles

3 Section Parameters

Angle Section Type	20 20 X 4
Spacing between Sections_Horizontal, S(mm)	38
Spacing between Sections_Vertical, S*(mm)	38
Plate Length(Horizontal), l(mm)	38
Plate Length(Vertical), l*(mm)	38
Plate Thickness, t(mm)	38

4 Section Properties

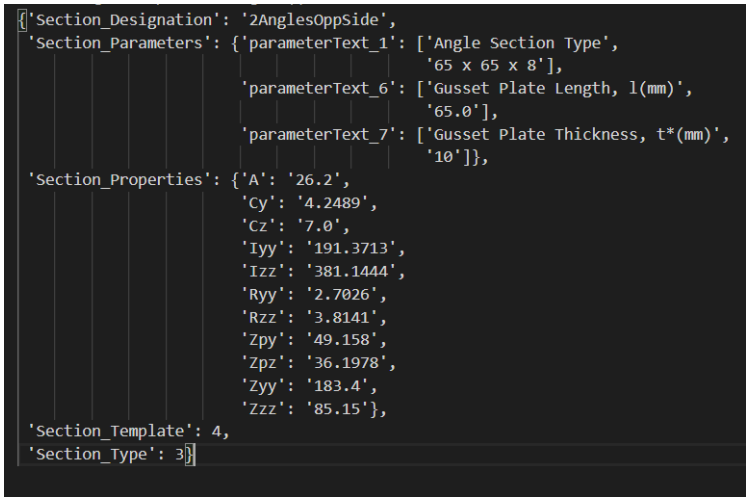
Area	5781.8
Radius of Gyration	
Rzz(mm)	29.521
Ryy(mm)	40.5014
Elastic Section Moduli	
Zzz(mm³)	238066.1932
Zyy(mm³)	369659.9612
Centroid	
Cy(mm)	41.1751
Cz(mm)	63.9351
Moment of Inertia	
Izz(mm)	5038790.855
Iyy(mm)	9484266.4653
Izy(mm)	7048690.668
(°)	45
Iu(mm)	61386970.784
Iv(mm)	89544231.254
Plastic Section Moduli	
Zpy(mm³)	3341363.4952
Zpz(mm³)	223757.5981

Page 1 of 2

4 3D View

Page 2 of 2

Figure 5.6: Design Report



```
{'Section_Designation': '2AnglesOppSide',
 'Section_Parameters': {'parameterText_1': ['Angle Section Type',
                                            '65 x 65 x 8'],
                        'parameterText_6': ['Gusset Plate Length, l(mm)',
                                            '65.0'],
                        'parameterText_7': ['Gusset Plate Thickness, t*(mm)',
                                            '10']},
 'Section_Properties': {'A': '26.2',
                        'Cy': '4.2489',
                        'Cz': '7.0',
                        'Iyy': '191.3713',
                        'Izz': '381.1444',
                        'Ryy': '2.7026',
                        'Rzz': '3.8141',
                        'Zpy': '49.158',
                        'Zpz': '36.1978',
                        'Zyy': '183.4',
                        'Zzz': '85.15'},
 'Section_Template': 4,
 'Section_Type': 3}
```

Figure 5.7: Saved Section