Structure Information Modeling (SIM)

Submitted for partial fulfilment of the Degree
of
Bachelor of Technology
(Information Technology)

Submitted By:
Amritpal Singh
1411234

Submitted To:

Department of Information Technology
Guru Nanak Dev Engineering College
Ludhiana 141006

Acknowledgement

I, student of Guru Nanak Dev Engineering College, Ludhiana, have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

The author is highly grateful to Dr. M.S. Saini Director, Guru Nanak Dev Engineering College, Ludhiana for providing him with the opportunity to carry out his Six Weeks Training at Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana.

The author would like to whole heartedly thank Dr. H.S. Rai Dean, Testing and Consultancy Cell, Guru Nanak Dev Engineering College, Ludhiana who is a vast sea of knowledge and without whose constant and never ending support and motivation, it would never have been possible to complete the project and other assignments so efficiently and effectively.

Finally, I would thanks My Mentors at FreeCAD organization Yorik van Havre. Without their encouragement and Guidence, it would not have been possible to complete this project in such an efficient manner.

Amritpal Singh

Abstract

CAD development project discuss the work done in computer-aided-design. Computer-aided design (CAD) is the use of computer systems to assist in the creation, modification, analysis, or optimization of a design.I explored FreeCADs source code. FreeCAD is Free and Open Source CAD Software. FreeCAD is a fully comprehensive 3D CAD application that you can download and install for free. There is a large base of satisfied FreeCAD users worldwide, and it is available in more than 20 languages and for all major operating systems, including Microsoft Windows, Mac OS X and Linux (De- bian, Ubuntu, Fedora, Mandriva, Suse ...). FreeCAD is an application for Computer Aided Design (CAD) in three dimensions (3d). With FreeCAD you can create technical drawings such as plans for buildings, interiors, mechanical parts or schematics and diagrams.

The software to be designed will create an index model for a 3 dimensional civil structure like a building or any other civil engineering project. This requires a software for designing such structures using a computer, the one used in SIM is STAAD PRO. In STAAD PRO the engineer draws the structure using various graphical tools and accordingly a .STD file is created that has all information about the building structure. Alternatively the engineer can choose to work from this file only and creating the elements of the building by writing pre-defined instructions into the file just like a script. It is this file that serves as the fundamental requirement for SIM. SIM parses this file and stores the parsed information into a database which is of object orientation naturea.

Also, this project makes the drawings of different views of the building on the drawing sheets. In this project, the user will be able to enter the specifications of the building through the web browser using Django (Python Web Framework) and on the back-end, the FreeCAD macros will use those input values to draw the drawings of different views of the building on different drawing sheets. The output of the same can then be taken by a user in SVG, PDF and fcstd formats.

This project is completely open source and the entire code is available to the user as and when required. There is also Complete developer's Documentation as well as User manual alongwith it that helps using it a lot easier.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION TO ORGANISATION

GURU NANAK DEV ENGINEERING COLLEGE
Contemplate and reflect upon knowledge, and you will become a
www.gnde

Figure 1.1: Guru Nanak Dev Engineering College

I had my Six Weeks Industrial Training at TCC-Testing And Consultancy Cell, GNDEC Ludhiana. Guru Nanak Dev Engineering College was established by the Nankana Sahib Education Trust Ludhiana. The Nankana Sahib Education Trust i.e NSET was founded in memory of the most sacred temple of Sri Nankana Sahib, birth place of Sri Guru Nanak Dev Ji. With the mission of Removal of Economic Backwardness through Technology Shiromani Gurudwara Parbandhak Committee i.e SGPC started a Poly technical was started in 1953 and Guru Nanak Dev Engineering College was established in 1956.

NSET resolved to uplift Rural areas by admitting 70% of students from these rural areas ever year. This commitment was made to nation on 8th April, 1956, the day foundation stone of the college building was laid by Dr. Rajendra Prasad Ji, the First President of India. The College is now ISO 9001:2000 certified.

Guru Nanak Dev Engineering College campus is spread over 88 acres of prime land about 5 Km s from Bus Stand and 8 Km s from Ludhiana Railway Station on Ludhiana-Malerkotla Road. The college campus is well planned with beautifully laid out tree plantation, pathways, flowerbeds besides the well maintained sprawling lawns all around. It has beautiful building for College,Hostels,Swimming Pool,Sports and Gymnasium Hall Complex, Gurudwara Sahib, Bank, Dispensary, Post Office etc. There are two hostels for boys and one for girls with total accommodation of about 550 students. The main goal of this institute is:

- To build and promote teams of experts in the upcoming specialisations.
- To promote quality research and undertake research projects keeping in view their relevance to needs and requirements of technology in local industry.
- To achieve total financial independence.
- To start online transfer of knowledge in appropriate technology by means of establishing multipurpose resource centres.

1.1 Testing and Consutancy Cell

My Six Weeks Institutional Training was done by me at TCC i.e Testing And Consultancy Cell, GNDEC Ludhiana under the guidance of Dr. H.S.Rai Dean Testing and Consultancy Cell. Testing and Consultancy Cell was established in the year 1979 with a basic aim to produce quality service for technical problems at reasonable and affordable rates as a service to society in general and Engineering fraternity in particular.

Figure 1.2: Testing and Consultancy Cell

Consultancy Services are being rendered by various Departments of the College to the industry, Sate Government Departments and Entrepreneurs and are extended in the form of expert advice in design, testing of materials & equipment, technical surveys, technical audit, calibration of instruments, preparation of technical feasibility reports etc. This consultancy cell of the college has given a new dimension to the development programmers of the College. Consultancy projects of over Rs. one crore are completed by the Consultancy cell during financial year 2009-10.

Ours is a pioneer institute providing Consultancy Services in the States of Punjab, Haryana, Himachal, J&K and Rajasthan. Various Major Clients of the Consultancy Cell are as under:

- Northern Railway, Govt. of India
- Indian Oil Corporation Ltd.
- Larson & Turbo.
- Multi National Companies like AFCON & PAULINGS.
- Punjab Water Supply & Sewage Board
- Power Grid Corporation of India.
- National Building Construction Co.
- Punjab State Electricity Board.
- Punjab Mandi Board.
- Punjab Police Housing Corporation.
- National Fertilizers Ltd.
- GLADA, Ludhiana

CHAPTER 2 INTRODUCTION TO PROJECT

2.1 Overview

Structure Information Modeling (SIM) is an open-source, free web-based software, developed by students of Testing and Consultancy Cell (TCC), under the guidance of Dr. H.S. Rai.The software to be designed will create an index model for a 3 dimensional civil structure like a building or any other civil engineering project. This requires a software for designing such structures using a computer, the one used in SIM is STAAD PRO. In STAAD PRO the engineer draws the structure using various graphical tools and accordingly a .STD file is created that has all information about the building structure. Alternatively the engineer can choose to work from this file only and creating the elements of the building by writing pre-defined instructions into the file just like a script. It is this file that serves as the fundamental requirement for SIM. SIM parses this file and stores the parsed information into a database which is of object orientation nature.

SIM will serve multiple users at a time. This is achieved by providing a web interface to things so that various user can access it from their web browsers. The user is required to make a decision at the home page of whether to upload a new file or open an already uploaded file. If the user chooses the former then the user must provide with a valid and complete STAAD PRO file for the system to further process.

Also in this project, the user makes the drawings of different views of the building on the drawing sheets. The user will be able to enter the specifications of the building through the web browser and on the back-end, the FreeCAD macros will use those input values to draw the drawings of different views of the building on different drawing sheets. The output of the same can then be taken by a user in SVG, PDF and fcstd formats. The user will also facilitate to render the 3D model of the building on the browser screen.

2.2 The Existing System

There are few existing systems for solving this particular problem like STAAD.Pro, SAP2000 but they dont have following features required by our mentor. These system were not open source and free web based software that were need.

All exiting system suffers from at least one of the following system.

Limitations of previous system

- No batch mode
- They are costly (STAAD.Pro costs nearly 80,000 rupees)
- Not open-source (User not modified the software)
- They need installation and a lot of system resources
- Platform dependent
- There is no any automations of drawing of 3D object (step-by-step approach is following for making drawings)
- Difficulty for taking sectional view of the object
- Doesn't support WebGl and IFC format

2.3 User Requirement Analysis

For User Requirement Analysis, users of this system have been asked about possible requirements that this software should have and we got following resultant list of outputs-:

1. Provide on-line way to analysis so that individual does not have to install anything.
2. Make it work like batch mode. so, that user can give inputs together and relax.
3. Help M.Tech and Civil Engineer to analysis structure.
4. Storing Staad Pro file to the database so that read data through queries.
5. All the data of Staad Pro file is store in database like a tree. By the result it is very easy to read data.
6. Rendering 3D model on browser without any plugin installation.
7. Make PDF of different views of the building

2.4 Feasibility Analysis

Feasibility analysis aims to uncover the strengths and weaknesses of a project. In its simplest term, the two criteria to judge feasibility are cost required and value to be attained. As such, a well-designed feasibility analysis should provide a historical background of the project, description of the project or service, details of the operations and management and legal requirements. Generally, feasibility analysis precedes technical development and project implementation. There is some feasibility factors by which we can determine that project is feasible or not:

- **Technical feasibility**: Technological feasibility is carried out to determine whether the project has the capability, in terms of software, hardware, personnel to handle and fulfill the user requirements. This whole project is based on parsing Staad Pro file to the database and on the FreeCAD and Django for user interface. Technical feasibility of this project revolves around the technical boundaries and limitations of the FreeCAD and Django. Structure Information Modeling (SIM) is technically feasible as it is built up in Open Source Environment and thus it can be run on any Open Source platform.

- **Economic feasibility**: Economic analysis is the most frequently used method to determine the cost/benefit factor for evaluating the effectiveness of a new system. In this analysis we determine whether the benefit is gain according to the cost invested to develop the project or not. If benefits outweigh costs, only then the decision is made to design and implement the system. It is important to identify cost and benefit factors, which can be categorized as follows:

  1. Development costs.
  2. Operating costs.

  Structure Information Modeling Software is also Economically feasible with 0 Development and Op- erating Charges as it is developed in Django framework and FreeCAD which is FOSS technology and the software is operated on Open Source platform.

- **Operational feasibility**: Operational feasibility is a measure of how well a project solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. All the Operations performed in the software are very quick and satisfies all the reuirements. This project is also operational feasible as it automates the work of solving the problem of analysising the structures which not only saves time but also saves money as most of the work is done by Employees and M.Tech students is done by this software.

2.5 Objective of Project

Structure Information Modeling Software is a web based software (that means it can run on any operation system) and the main objectives of this project is to:

1. To inspire M.Tech students to automate their work and do programming
2. Perform most of difficult Calculation work
3. Make it work like batch mode. so, that user can give inputs together and relax.
4. Accept inputs from the user in *.std file format
5. Help M.Tech and Civil Engineer to analysis structure.
6. Reduce the time for analysis.
7. Generates the final output in the form of PDF, SVG and .fcstd
8. Provide on-line way to analysis so that individual does not have to install anything.
9. Adminstration login page

CHAPTER 3 PROJECT DESIGN

3.1 Software Requirement Analysis

A Software Requirements Analysis for a software system is a complete description of the behavior of a system to be developed. It include functional Requirements and Software Requirements. In addition to these, the SRS also contains non-functional requirements. Non-functional requirements are requirements which impose constraints on the design or implementation.

- **Purpose**: Structure Information Modeling is a web based software and the main purpose of this project is to:
  1. Make it work like batch mode. so, that user can give inputs together and relax.
  2. Help M.Tech and Civil Engineer to analysis structure.
  3. Automatic calculation of modal force and modes.
  4. Reduce the time for analysis.
  5. Rendering 3D model on browser without any plugin installation.
  6. Storing Staad Pro file to the database so that read data through queries.
  7. Make PDF, SVG and .fcstd of different views of the building
- **Users of the System**:
  1. Client : Clients are the end users that benefit from this software. They just provide input and gets output in form of PDF.Client of this WEB Application can be of two types:
     (a) Civil Engineer -: They have knowledge of working of procedure and what input is being provided.
     (b) Layman -: They dont know anything about whats going on, their just work is to give input to system.

3.1.1 Functional Requirements

- **Specific Requirements**: This phase covers the whole requirements for the system. After understanding the system we need the input data to the system then we watch the output and determine whether the output from the system is according to our requirements or not. So what we have to input and then what well get as output is given in this phase. This phase also describe the software and non-function requirements of the system.

- **Input Requirements of the System**:

  1. Staad Pro file
  2. Number of stories
  3. Length of depth of foundation
  4. Plinth level
  5. Clear height
  6. Depth of slab
  7. Representation of span length
  8. Representation of span width
  9. Column item
  10. Length of column
  11. Width of column
  12. Radius of column
  13. Depth of beam
  14. Width of beam

- **Output Requirements of the System**:

  1. Generation of drawing in form of PDF and SVG
  2. Generation of FreeCAD project (.fcstd file)
  3. Rendering 3D model (WebGL format)

- **Special User Requirements**:

  1. Taking bulk input values in form of Staad Pro file
  2. Query

- **Software Requirements**:

  1. Programming language: Python 2.7, C++
  2. Software: FreeCAD, Inkscape
  3. Framework: Django 1.9, Materialize

4. Database: MySQL
5. Text Editor: Vim
6. Operating System: Ubuntu 12.04 or up
7. Revision System: Git

3.1.2 Non functional requirements

1. Scalability: System should be able to handle a number of users. For e.g., handling around thousand users at the same time.
2. Usability: Simple user interfaces that a layman can understand.
3. Speed: Processing input should be done in reasonable time i.e. we can say maximum 24 hrs.

3.2 Dependencies

Dependencies include softwares or framework that need to be installed for proper working of this software.

1. Programming language: Python 2.7, C++
2. Software: FreeCAD, Inkscape
3. Framework: Django 1.9
4. Operating System: Any on which above dependencies can be installed

3.3 DFDs

A data flow diagram (DFD) is a graphical representation of the ”flow” of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system, which can later be elaborated. DFDs of DoS is as following-:

1. Data flow LEVEL 0 figure 3.1
2. Data flow LEVEL 1 figure 3.2

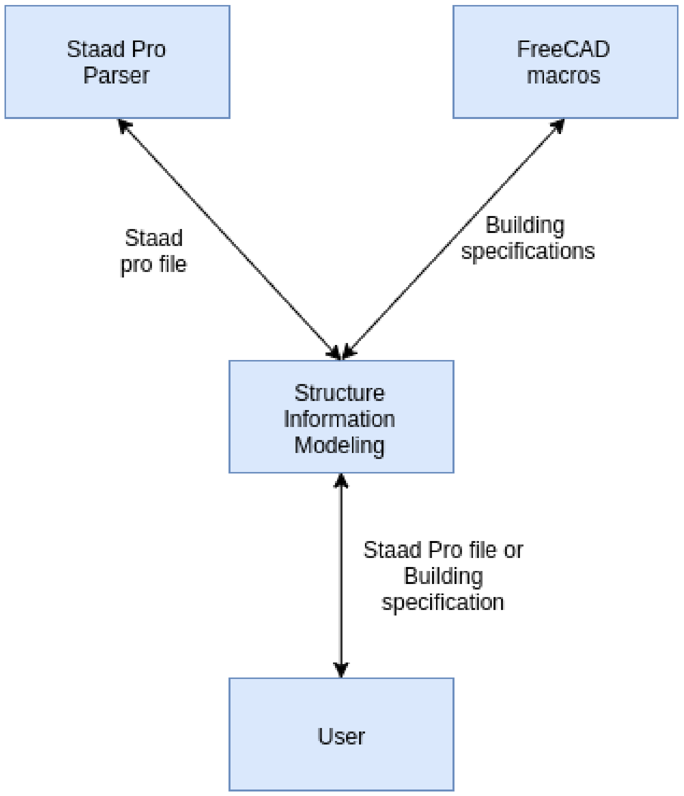


Figure 3.1: Data flow LEVEL 0

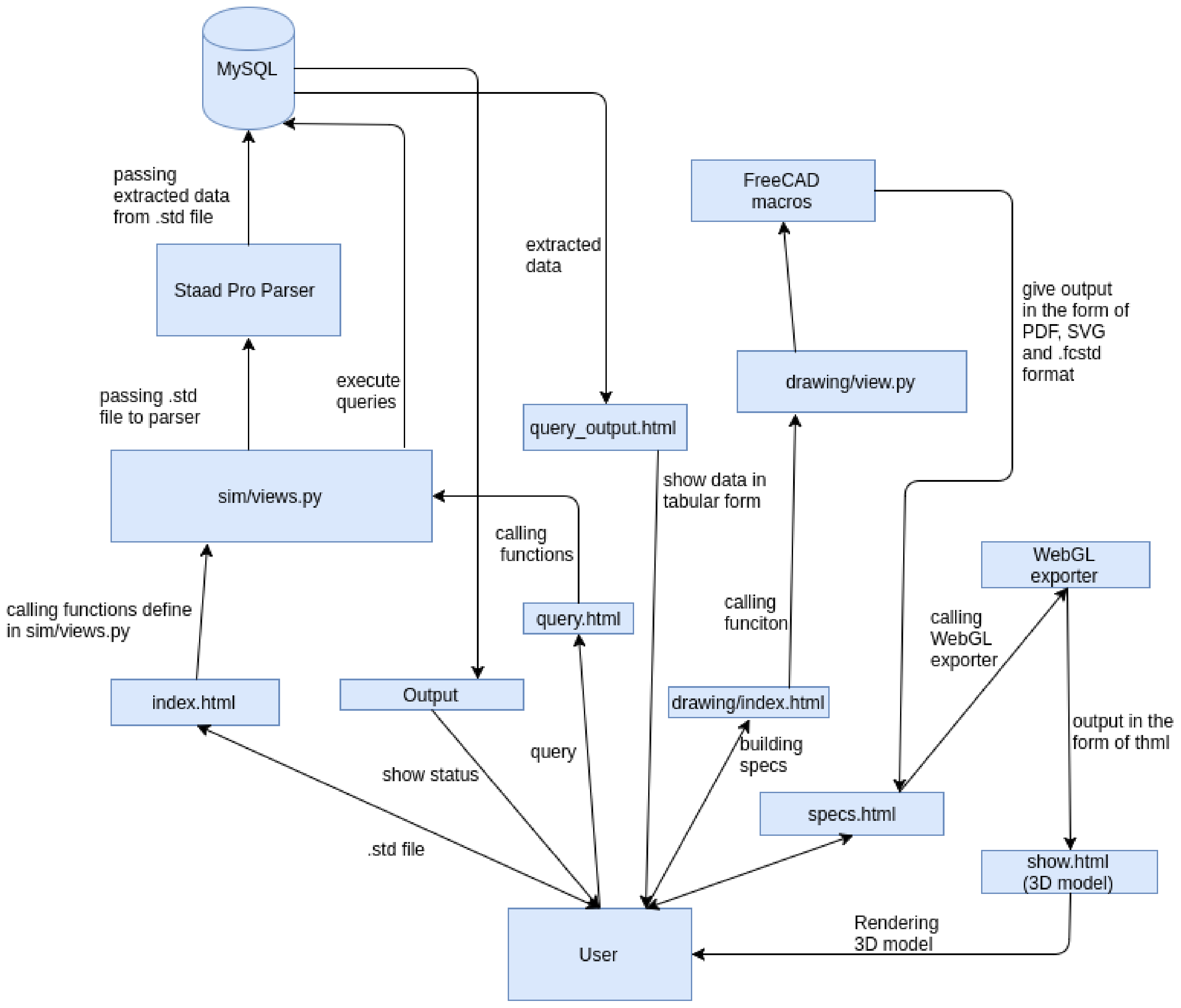


Figure 3.2: Data Flow LEVEL 1

CHAPTER 4 DEVELOPMENT AND IMPLEMENTATION

4.1 Python

Figure 4.1: Python logo

Python is a dynamic language, as in python coding is very easy and also it require less coding and about its interpreted nature it is just exellent. Python is a high level programming language and Django which is a web development framework is written in python language.

Python is an easy to learn, powerful programming language.Python runs on Windows, Linux/Unix, Mac OS X. Python is free to use, even for commercial products. Python can also be used as an extension language for existing modules and applications that need a programmable interface. Python is free to use, even for commercial products, because of its OSI-approved open source license.

4.1.1 Features of Python

- Very clear, readable syntax.
- Strong introspection capabilities.
- Intuitive object orientation.
- Natural expression of procedural code.
- Full modularity, supporting hierarchical packages.
- Exception-based error handling.
- Very high level dynamic data types.

- Extensive standard libraries and third party modules for virtually every task.
- Extensions and modules easily written in C, C++ (or Java for Jython, or .NET languages for IronPython).
- Embeddable within applications as a scripting interface.

4.1.2 Installation of Python

Installation of python is a very easy proccess. The current python versions are: Python 2.7.1 and Python 3.2. Type the commands in the terminal:

```
$ wget http://www.python.org/ftp/python/2.7/Python-2.7.tgz

$ tar xzf Python-2.7.tgz
```

This will install the python on your pc/laptop.

4.2 FreeCAD

FreeCAD is a general purpose parametric 3D CAD modeler. The development is completely Open Source (LGPL License). FreeCAD is aimed directly at mechanical engineering and product design but also fits in a wider range of uses around engineering, such as architecture or other engineering specialties.

FreeCAD features tools similar to Catia, SolidWorks or Solid Edge, and therefore also falls into the category of MCAD, PLM, CAx and CAE. It is a feature based parametric modeler with a modular software architecture which makes it easy to provide additional functionality without modifying the core system.

As with many modern 3D CAD modelers it has many 2D components in order to sketch 2D shapes or extract design details from the 3D model to create 2D production drawings, but direct 2D drawing (like AutoCAD LT) is not the focus, neither are animation or organic shapes (like Maya, 3ds Max, Blender or Cinema 4D), although, thanks to its wide adaptability, FreeCAD might become useful in a much broader area than its current focus.

FreeCAD makes heavy use of all the great open-source libraries that exist out there in the field of Scientific Computing. Among them are OpenCascade, a powerful CAD kernel, Coin3D, an incarnation of Open Inventor, Qt, the world-famous UI framework, and Python, one of the best scripting languages available. FreeCAD itself can also be used as a library by other programs.

FreeCAD is also fully multi-platform, and currently runs flawlessly on Windows and Linux/Unix and Mac OSX systems, with the exact same look and functionality on all platforms.

4.2.1 FreeCAD's features

4.2.1.1 Key features

- A complete Open CASCADE Technology-based geometry kernel allowing complex 3D operations on complex shape types, with native support for concepts like brep, nurbs curves and surfaces, a wide range of geometric entities, boolean operations and fillets, and built-in support of STEP and IGES formats.

- A full parametric model. All FreeCAD objects are natively parametric, which means their shape can be based on properties or even depend on other objects, all changes being recalculated on demand, and recorded by the undo/redo stack. New object types can be added easily, that can even be fully programmed in Python.
- A modular architecture that allow plugins (modules) to add functionality to the core application. Those extensions can be as complex as whole new applications programmed in C++ or as simple as Python scripts or self-recorded macros. You have complete access from the Python built-in interpreter, macros or external scripts to almost any part of FreeCAD, being geometry creation and transformation, the 2D or 3D representation of that geometry (scenegraph) or even the FreeCAD interface.
- Import/export to standard formats such as STEP, IGES, OBJ, STL, DXF, SVG, STL, DAE, IFC or OFF, NASTRAN, VRML in addition to FreeCAD's native Fcstd file format. The level of compatibility between FreeCAD and a given file format can vary, since it depends on the module that implements it.
- A Sketcher with constraint-solver, allowing to sketch geometry-constrained 2D shapes. The sketcher currently allows you to build several types of constrained geomerty, and use them as a base to build other objects throughout FreeCAD.
- A Robot simulation module that allows to study robot movements. The robot module already has an extended graphical interface allowing GUI-only workflow.
- A Drawing sheets module that permit to put 2D views of your 3D models on a sheet. This modules then produces ready-to-export SVG or PDF sheets. The module is still sparse but already features a powerful Python functionality.
- A Rendering module that can export 3D objects for rendering with external renderers. Currently only supports povray and LuxRender, but is expected to be extended to other renderers in the future.
- An Architecture module that allows BIM-like workflow, with IFC compatibility. The making of the Arch module is heavily discussed by the community here.

4.2.2 FreeCAD workbenches used in this project

4.2.3 Part module

The CAD capabilities of FreeCAD are based on the OpenCasCade kernel. The Part module allows FreeCAD to access and use the OpenCasCade objects and functions. OpenCascade is a professional-level CAD kernel, that features advanced 3D geometry manipulation and objects. The Part objects, unlike Mesh Module objects, are much more complex, and therefore permit much more advanced operations, like coherent boolean operations, modifications history and parametric behaviour.

4.2.3.1 Draft module

The Draft workbench allows to quickly draw simple 2D objects in the current document, and offers several tools to modify them afterwards. Some of these tools also work on all other FreeCAD

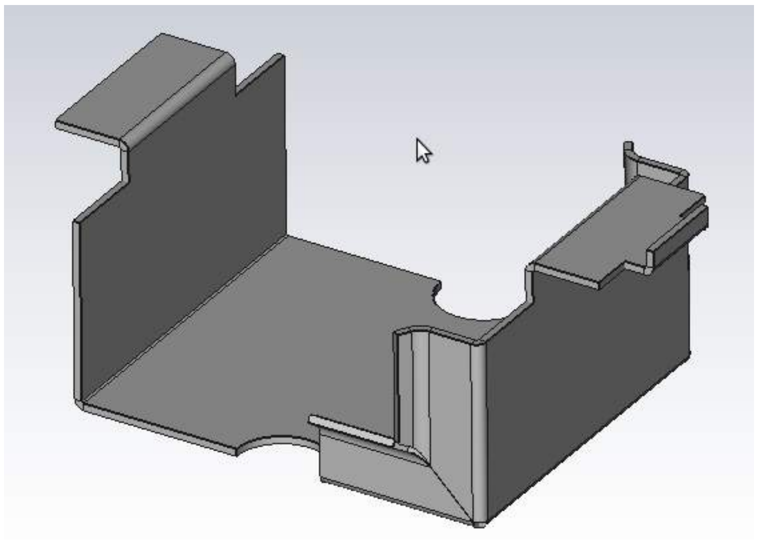


Figure 4.2: Example of Part shapes in FreeCAD

objects, not only those created with the Draft workbench. It also provides a complete snapping system, and several utilities to manage objects and settings.

4.2.4 Drawing module

The Drawing module allows you to put your 3D work on paper. That is, to put views of your models in a 2D window and to insert that window in a drawing, for example a sheet with a border, a title and your logo and finally print that sheet. The Drawing module is currently under construction and more or less a technology preview!

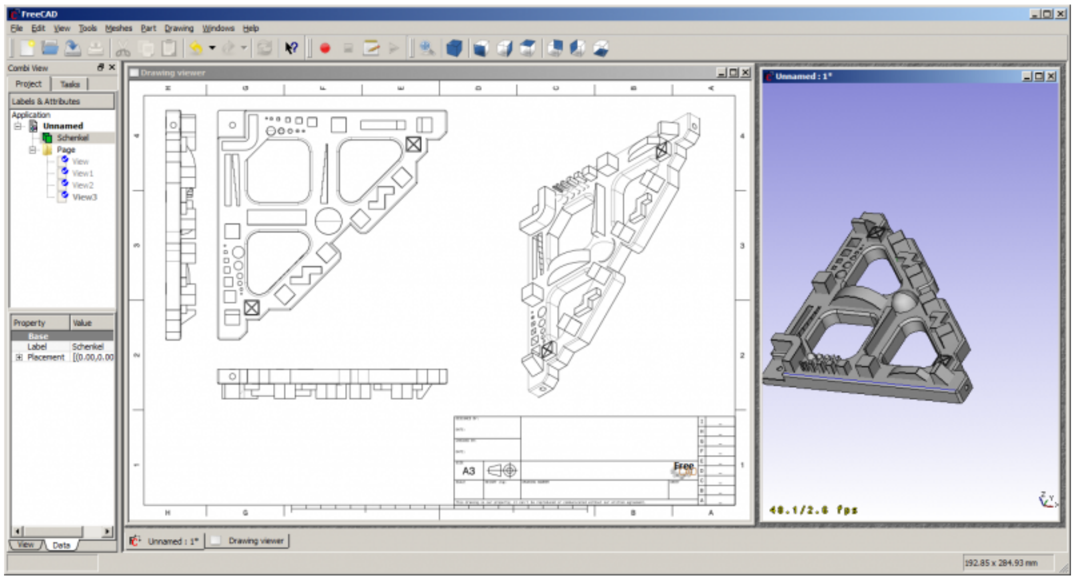


Figure 4.3: Example of Drawing module in FreeCAD

4.2.5 Design implementation using Python scripting language

Python is a programming language, very simple to use and very fast to learn. It is open-source, multi-platform, and can be used alone for a wide array of things, from programming simple shell scripts to very complex programs. But one of its most widespread uses is as a scripting language, since it is easy to embed in other applications. That's exactly how it is used inside FreeCAD. From the python console, or from your custom scripts, you can pilot FreeCAD, and make it perform very complex actions for which there is still no graphical user interface tool.
For example, from a python script, you can:

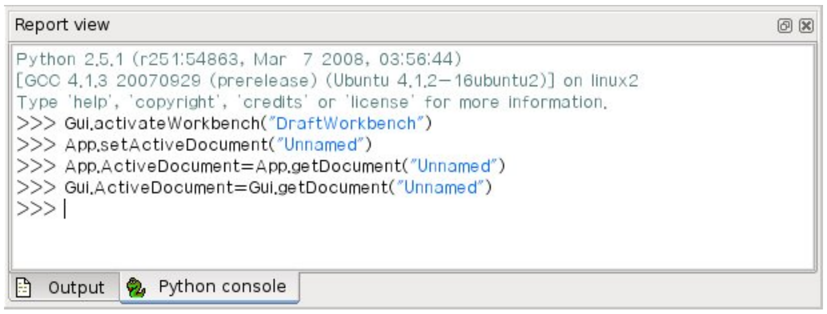
- create new objects
- modify existing objects
- modify the 3D representation of those objects
- modify the FreeCAD interface

There are also several different ways to use python in FreeCAD:

- From the FreeCAD python interpreter, where you can issue simple commands like in a "command line"-style interface
- From macros, which are a convenient way to quickly add a missing tool to the FreeCAD interface
- From external scripts, which can be used to program much more complex things. like entire Workbenches.

4.2.5.1 Writing python code

There are two easy ways to write python code in FreeCAD: From the python console (available from the View -¿ Panels -¿ Python console menu) or from the Macro editor (Tools -¿ Macros). In the console, you write python commands one by one, which are executed when you press return, while the macros can contain a more complex script made of several lines, which is executed only when the macro is executed.



```
Python 2.5.1 (r251:54863, Mar  7 2008, 03:56:44)
[GCC 4.1.3 20070929 (prerelease) (Ubuntu 4.1.2-16ubuntu2)] on linux2
Type 'help', 'copyright', 'credits' or 'license' for more information.
>>> Gui.activateWorkbench("DraftWorkbench")
>>> App.setActiveDocument("Unnamed")
>>> App.ActiveDocument=App.getDocument("Unnamed")
>>> Gui.ActiveDocument=Gui.getDocument("Unnamed")
>>> |
```

Figure 4.4: Example of python scripting

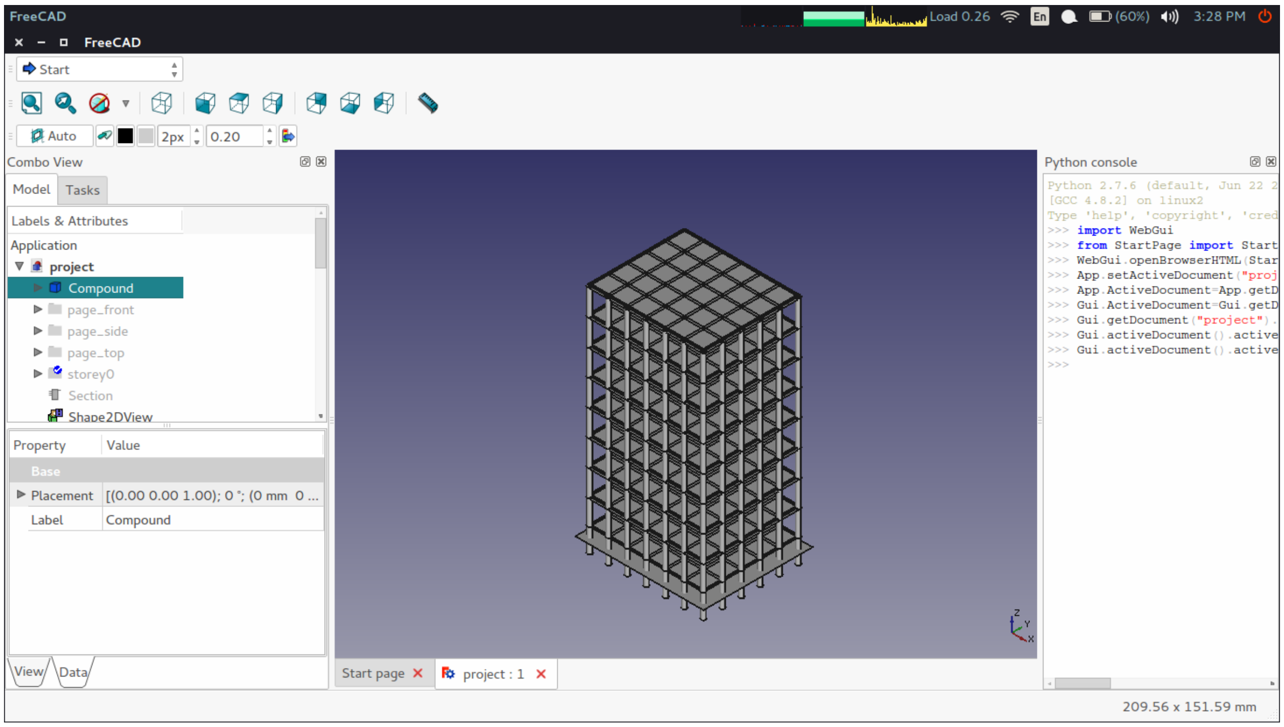
4.2.6 Macros

Macros are a convenient way to create complex actions in FreeCAD. You simply record actions as you do them, then save that under a name, and replay them whenever you want. Since macros are in reality a list of python commands, you can also edit them, and create very complex scripts.

In this project I have made many macros which are using for different purposes.

- **Builing macro:** This macro will create the model of building by using the user input in 3D space of FreeCAD. For running this macro in the project I have used the below command.

  ```
  freecadcmd builing_macro.py
  ```



```
Python 2.7.6 (default, Jun 22 2
[GCC 4.8.2] on linux2
Type 'help', 'copyright', 'cred
>>> import WebGui
>>> from StartPage import Start
>>> WebGui.openBrowserHTML(Star
>>> App.setActiveDocument("proj
>>> App.ActiveDocument=App.getD
>>> Gui.ActiveDocument=Gui.getD
>>> Gui.getDocument("project").
>>> Gui.activeDocument().active
>>> Gui.activeDocument().active
>>>
```

Figure 4.5: Example of building macro

- **Drawing macro:** This macro will draw the drawing of different views (top, front and side view) of the builing along with sectional view of each building storey on the drawing sheet. (3D model).

  ```
  freecadcmd drawing.py
  ```

- **Saving drawing macro:** This macro will save the drawing of the building in SVG (Scalable Vector Graphics) and then convert SVG file into PDF (Portable Document Format).

  ```
  freecadcmd save_drawing.py
  ```

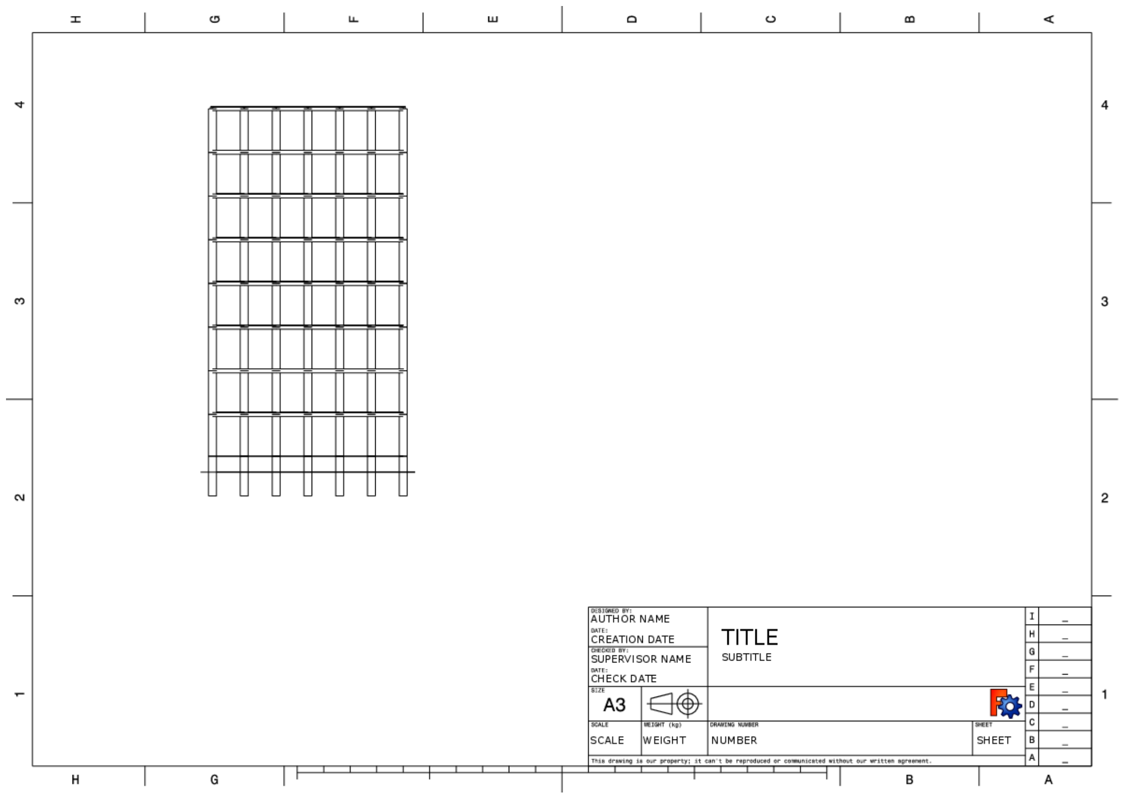


Figure 4.6: Example of drawing macro

4.3 Inkscape

Inkscape is an open source drawing tool for creating and editing SVG graphics. More than just a text vector editor, Inkscape provides a WYSIWYG interface for manipulation of vector images, allowing the artist to express himself freely. While other free and proprietary software exists with similar capabilities, Inkscape provides an interface to directly manipulate the underlying SVG code, which allows one to be certain that the code is in compliance with W3C standards. Since the beginning of its development, the Inkscape project has been a very active, providing stability for the current software and growth of capacities in the future. Like other drawing programs, Inkscape offers creation of basic shapes (such as ellipses, rectangles, stars, polygons and spirals) as well as the ability to transform and manipulate these basic shapes by rotation, stretching and skewing.

Inkscape also offers functionality to manipulate objects more precisely by adjusting node points and curves. These functions are indispensable to useful drawing software, and allow the advanced artist to freely create what he imagines.

The properties of objects can either be manipulated individually and precisely through the XML editor, or more generally in an intuitive fashion by input devices such as mice, pen tablets or even touch screen. In addition, Inkscape allows one to insert text and bitmaps (such as PNG, another W3C recommended bitmap image format) into an image as well as perform some basic editing functions on them. If further bitmap editing is required, other tools may be used (such as the GIMP) on images before importing them or after. In fact, if a linked bitmap is edited in another program, Inkscape will reflect these changes once the SVG is reloaded.

All of these characteristics make Inkscape a model drawing application, especially considering its flexibility and many other capabilities. Its strict compliance with the W3C SVG standards allow excellent portability of images to many applications and platforms on which these applications are used.

4.3.1 About SVG

Those who work with graphics for internet use are familiar with the problems tied to publication of images on the web. Traditionally, bitmap images (such as JPG or GIF) have been the only option for use in such documents, with the disadvantage that these images are either too large for quick transfer or, if they are small or highly compressed to reduce file-size, of poor quality.

As a solution to this problem, Macromedia created the Flash image format. While Flash satisfactorily solved the main problems inherent to bitmap images, there has been discontent for some users that the common vector format for the web is dependent solely on Macromedia for development of the file format and software. In order to address this discontent and provide an open option for vector graphics, the W3C created the SVG file format, making a freely usable vector format available to everyone.

Most image files are only able to be read by specific software that renders the image. SVG, however, is described in XML and CSS, and its files can be opened and edited in any ASCII text editor. While it is possible to create SVG images in this manner, it is highly unproductive and unintuitive. SVG editors and renderers have the ability to easily open and manipulate SVG files without a special interpreter.

4.3.2 Objectives of the SVG Format

The advantages of SVG are the same as for any vector image: high-quality images that are smooth and crisp ability to resize the image to any dimensions without diminishing quality, which is impossible with bitmap images. The SVG standard also defines animation, and with a little use of Javascript, one can make SVG interactive. Finally, since SVG is written in XML, it is possible to create graphics based on data that is stored in other XML-based formats, such as graphs, charts and maps. Despite its benefits, there is a lack of usable software to create and edit SVG files and take full advantage of its capacities; for this reason, SVG is not as usable at the moment as Flash.

4.3.3 Export PDF macro

It uses Inkscape to do the SVG to PDF conversion. Customize ToolsBar instructions can be used to add tool button in Drawing WB and therefore to be able to use Export PDF macro with ease when needed.

```
import os
import subprocess
from PySide import QtGui

obj = FreeCADGui.Selection.getSelection()[0]

# Path to Inkscape executable
inkscape_path = "/usr/bin/inkscape"
```

```
# Exported PDF file location
file_location = os.path.expanduser("~" + os.sep + obj.Label + ".pdf")

# ----------------------------------------------------------------------
try:
  page = getattr(obj, 'PageResult')
  except AttributeError:
    QtGui.QMessageBox.warning(None,"Export PDF", "Page object has \
        not been selected.")
    else:
        file_exists = os.path.isfile(file_location)
            if file_exists == True:
                QtGui.QMessageBox.warning(None,"Export PDF", "A \
                    file with the name " + obj.Label + ".pdf already \
                        exists:\n\n" + file_location)
                    else:
                         call_inkscape = [inkscape_path, "-f", \
                            obj.PageResult,"-A", file_location]
                         subprocess.call(call_inkscape)
                         QtGui.QMessageBox.information(None,"Export \
                            PDF", "Successfully exported:\n\n" + \
                                file_location)
```

4.3.3.1 What does it do?

It exports drawing page to PDF file using Inkscape. Instead of bitmap image inserted in PDF file elements like text are searchable and exported PDF file size can be smaller when the drawings do not have big number of elements in it.

4.3.3.2 How to use it

- Select Page object in Tree View.
- Run Export PDF macro.

4.4 C++

C++ is a general-purpose programming language. It has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation.

It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, web search or SQL servers), and performance-critical applications (e.g. telephone switches or space probes). C++ is a compiled language, with implementations of it available on many platforms and provided by various organizations, including the Free Software Foundation (FSF's GCC), LLVM, Microsoft, Intel and IBM.

Figure 4.7: C++ Logo

C++ is standardized by the International Organization for Standardization (ISO), with the latest standard version ratified and published by ISO in December 2014 as ISO/IEC 14882:2014 (informally known as C++14). The C++ programming language was initially standardized in 1998 as ISO/IEC 14882:1998, which was then amended by the C++03, ISO/IEC 14882:2003, standard. The current C++14 standard supersedes these and C++11, with new features and an enlarged standard library. Before the initial standardization in 1998, C++ was developed by Bjarne Stroustrup at Bell Labs since 1979, as an extension of the C language as he wanted an efficient and flexible language similar to C, which also provided high-level features for program organization.

Many other programming languages have been influenced by C++, including C#, D, Java, and newer versions of C (after 1998).

Features of Language:

1. Object storage
   (a) Static storage duration objects
   (b) Thread storage duration objects
   (c) Automatic storage duration objects
   (d) Dynamic storage duration objects
2. Templates
3. Objects
   (a) Encapsulation
   (b) Inheritance
4. Operators and operator overloading
5. Polymorphism
   (a) Static polymorphism
   (b) Dynamic polymorphism
      i. Inheritance
      ii. Virtual member functions

6. Lambda expressions
7. Exception handling

4.5 Front End Languages and Framework [1]

Front End languages are language that are used to give better user experince and user interface. These mainly include HTML, CSS, Javascript. Some Frameforks like Bootstrap are also used with these basic languages.

4.5.1 HTML

Figure 4.8: HTML5 logo

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications. Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <p>Hello world!</p>
  </body>
</html>
```

[1] Used in project but not by me accept basic HTML

4.5.2 CSS

Figure 4.9: CSS logo

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language.Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions

```
p {
    color: red;
    text-align: center;
}
```

4.5.3 Javascript

JavaScript (/dvskrpt/) is a high-level, dynamic, untyped, and interpreted programming language. It has been standardized in the ECMAScript language specification. Alongside HTML and CSS, it is one of the three essential technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern web browsers without plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage or graphics facilities, relying for these upon the host environment in which it is embedded.

Figure 4.10: Javascript logo

4.5.4 Materialize

Materialize is a UI component library created with CSS, JavaScript, and HTML. Materialize UI components helps in constructing attractive, consistent, and functional web pages and web apps while adhering to modern web design principles like browser portability, device independence, and graceful degradation. It helps in creating faster, beautiful, and responsive websites. It is inspired from Google Material Design.

4.6 Introduction to Django

Figure 4.11: Django logo

Django is an open source web application framework written in python. It lets you build high-performing, elegant Web applications quickly. Django focuses on automating as much as possible. Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the DRY principal. Python is used throughout, even for settings, files, and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.

Django takes it's name from the early jazz guitarist Django Reinhardt, a gypsy savant who managed to play dazzling and electrifying runs on his instrument even though two of the fingers on his left hand were paralyzed in an accident when he was young.

Thus its a fitting name for the framework. Django can do some very complex things with less code and a simpler execution than youd expect. It doesn't take a heavy hand to build with Django. The framework does the repetitive work for you, allowing you to get a working website up quickly and easily.

4.6.1 Features of Django

- Clean URLs
- Object- Relational Mapping
- Loosely coupled components
- Designer-friendly templates
- Cache framework
- MVC architecture
- Jython support
- DRY (Don't Repeat Yourself)

4.6.2 Installation of Django

Installation of Django is very easy. To install Django version 1.4, type the following commands:

```
$ wget http://www.djangoproject.com/download/1.4/tarball

$ tar xzvf Django-1.4.tar.gz

$ cd Django-1.4

$ sudo python setup.py install
```

This will install the django on your system.

4.6.3 MTV

Django adopts the standard MVC (Model-View-Controller) design pattern. But instead, their naming convention is the MTV (Model-Template-View).

- **Model** is an object relational mapping to your database schema. So each model is a class which represents a table in your database. Django models provide easy access to an underlying data storage mechanism, and can also encapsulate any core business logic, which must always remain in effect, regardless of which application is using it. Models exist independent of the rest of the system, and are designed to be used by any application that has access to them. In fact, the database manipulation methods that are available on model instances can be utilized even from the interactive interpreter, without loading a Web server or any application-specific logic.
- **Template** is simply HTML for your views. It also allows you to display different messages depending on whether or not a user is logged in. Templates are Django's provided way of generating text-based output, such as HTML or emails, where the people editing those documents may not have any experience with Python. Therefore, templates are designed

to avoid using Python directly, instead favoring an extensible, easy-to-use custom language built just for Django.

- **View** could be a homepage or a page to display a user's information, for instance. A view accepts user input, including simple requests for information; behaves according to the application's interaction logic; and returns a display that is suitable for user's to access the data represented by models.

4.6.4 Creating Prject in Django

If this is your first time using Django, you'll have to take care of some initial setup. Namely, you'll need to auto-generate some code that establishes a Django project- a collection of settings for an instance of Django, including database configuration, Django-specific options and application-specific settings. From the command line, cd into a directory where you'd like to store your code, then run the command

```
$ django-admin.py startproject mysite
```

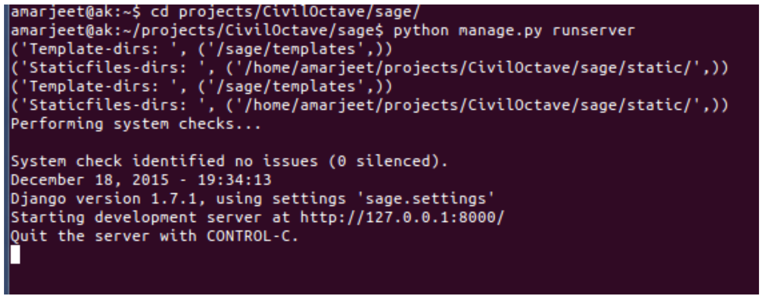
This will create a mysite directory in your current directory.

4.6.5 Development Server in Django

Change into the outer mysite directory, if you haven't already, and run the command

```
$ python manage.py runserver
```

You'll see the following output on the command line:



```
amarjeet@ak:~$ cd projects/CivilOctave/sage/
amarjeet@ak:~/projects/CivilOctave/sage$ python manage.py runserver
('Template-dirs: ', ('/sage/templates',))
('Staticfiles-dirs: ', ('/home/amarjeet/projects/CivilOctave/sage/static/',))
('Template-dirs: ', ('/sage/templates',))
('Staticfiles-dirs: ', ('/home/amarjeet/projects/CivilOctave/sage/static/',))
Performing system checks...

System check identified no issues (0 silenced).
December 18, 2015 - 19:34:13
Django version 1.7.1, using settings 'sage.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Figure 4.12: Output of runserver

4.6.6 Database setup

In this, we need to edit the settings.py file of the Project, that is the configuration file. It's a normal Python module with module-level variables representing Django settings. Change the following keys in the DATABASES 'default' item to match your database connection settings.

- ENGINE – Either 'django.db.backends.postgresql_psycopg2', 'django.db.backends.mysql', 'django.db.backends.sqlite3' or 'django.db.backends.oracle'. Other backends are also available.
- NAME – The name of your database. If you're using SQLite, the database will be a file on your computer; in that case, NAME should be the full absolute path, including filename, of that file. If the file doesn't exist, it will automatically be created when you synchronize the database for the first time (see below). When specifying the path, always use forward slashes, even on Windows (e.g. C:/homes/user/mysite/sqlite3.db).
- USER – Your database username (not used for SQLite).
- PASSWORD – Your database password (not used for SQLite).
- HOST – The host your database is on. Leave this as an empty string if your database server is on the same physical machine (not used for SQLite).

If you're new to databases, we recommend simply using SQLite by setting ENGINE to 'django.db.backends.sqlite3' and NAME to the place where you'd like to store the database. SQLite is included as part of Python 2.5 and later, so you won't need to install anything else to support your database.

While you're editing settings.py, set TIME_ZONE to your time zone. The default value is the Central time zone in the U.S. (Chicago).

Also, note the INSTALLED_APPS setting toward the bottom of the file. That holds the names of all Django applications that are activated in this Django instance. Apps can be used in multiple projects, and you can package and distribute them for use by others in their projects.

By default, INSTALLED_APPS contain the following apps, all of which come with Django:

- django.contrib.auth – An authentication system.
- django.contrib.contenttypes – A framework for content types.
- django.contrib.sessions – A session framework.
- django.contrib.sites – A framework for managing multiple sites with one Django installation.
- django.contrib.messages – A messaging framework.
- django.contrib.staticfiles – A framework for managing static files.

These applications are included by default as a convenience for the common case.

Each of these applications makes use of at least one database table, though, so we need to create the tables in the database before we can use them. To do that, run the following command:

```
$ python manage.py syncdb
```

The syncdb command looks at the INSTALLED_APPS setting and creates any necessary database tables according to the database settings in your settings.py file. You'll see a message for each database table it creates, and you'll get a prompt asking you if you'd like to create a superuser account for the authentication system. Go ahead and do that.

4.7 Introduction to LaTeX

LaTeX, I had never heard about this term before doing this project, but when I came to know about its features, found it excellent. LaTeX (pronounced /letk/, /letx/, /ltx/, or /ltk/) is a document markup language and document preparation system for the TeX typesetting program. Within the typesetting system, its name is styled as LaTeX.

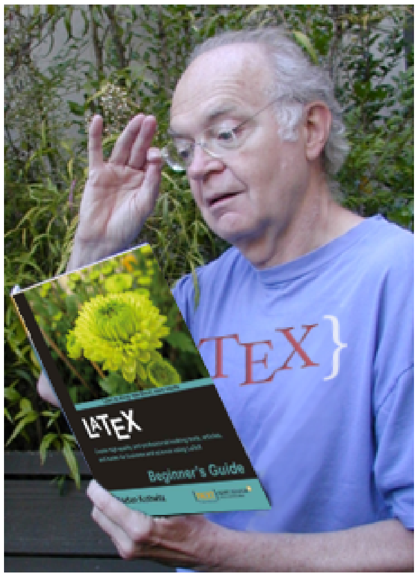


Figure 4.13: Donald Knuth, Inventor Of TeX typesetting system

Within the typesetting system, its name is styled as LaTeX. The term LaTeX refers only to the language in which documents are written, not to the editor used to write those documents. In order to create a document in LaTeX, a .tex file must be created using some form of text editor. While most text editors can be used to create a LaTeX document, a number of editors have been created specifically for working with LaTeX.

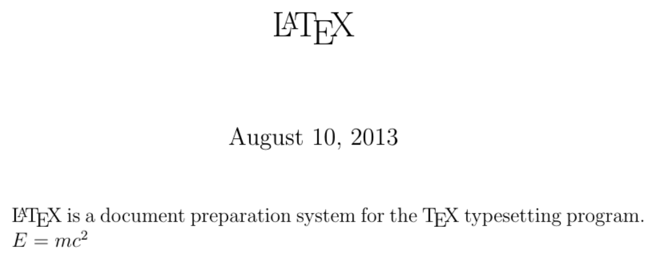
LaTeX is most widely used by mathematicians, scientists, engineers, philosophers, linguists, economists and other scholars in academia. As a primary or intermediate format, e.g., translating DocBook and other XML-based formats to PDF, LaTeX is used because of the high quality of typesetting achievable by TeX. The typesetting system offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout and bibliographies.

LaTeX is intended to provide a high-level language that accesses the power of TeX. LaTeX essentially comprises a collection of TeX macros and a program to process LaTeXdocuments. Because the TeX formatting commands are very low-level, it is usually much simpler for end-users to use LaTeX.

4.7.1 Typesetting

LaTeX is based on the idea that authors should be able to focus on the content of what they are writing without being distracted by its visual presentation. In preparing a LaTeX document, the author specifies the logical structure using familiar concepts such as chapter, section, table, figure, etc., and lets the LaTeX system worry about the presentation of these structures. It therefore encourages the separation of layout from content while still allowing manual typesetting adjustments where needed.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\LaTeX}
\date{}
\begin{document}
  \maketitle
  \LaTeX{} is a document preparation system
  for the \TeX{} typesetting program.
   \par
   $E=mc^2$
\end{document}
```



LaTeX

August 10, 2013

LaTeX is a document preparation system for the TeX typesetting program.
$E = mc^2$

Figure 4.14: LaTeX output of above program.

4.7.2 Installing LaTeX on System

Installation of LaTeX on personal system is quite easy. As i have used LaTeX on Ubuntu 13.04 so i am discussing the installation steps for Ubuntu 13.04 here:

- Go to terminal and type

  $ sudo apt-get install texlive-full

- Your Latex will be installed on your system and you can check for manual page by typing.

  $ man latex
  in terminal which gives manual for latex command.

- To do very next step now one should stick this to mind that the document which one is going to produce is written in any type of editor whether it may be your most common usable editor Gedit or you can use vim by installing first vim into your system using command.

  $ sudo apt-get install vim

- After you have written your document it is to be embedded with some set of commands that Latex uses so as to give a structure to your document. Note that whenever you wish your document to be looked into some other style just change these set of commands.

- When you have done all these things save your piece of code with .tex format say test.tex. Go to terminal and type

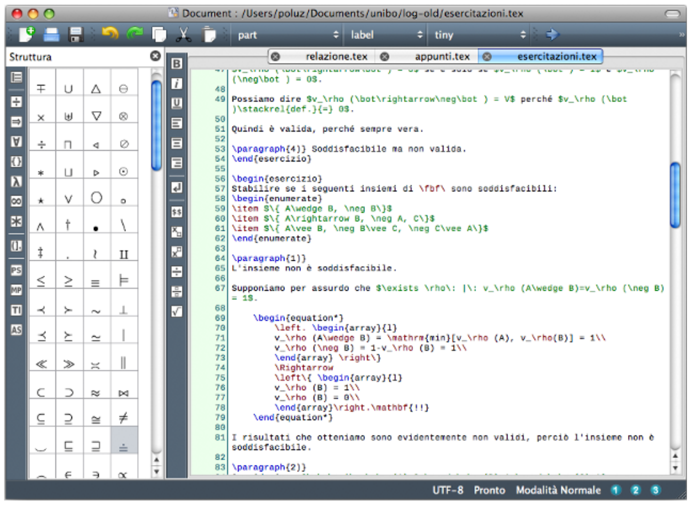
  latex path of the file test.tex Or pdflatex path of the file test.tex
  eg: pdflatex test.tex
  for producing pdf file simultaneously.
  After compiling it type command

  $ evince filename.pdf
  eg: evince test.pdf
  To see output pdf file.

4.7.3 Graphical Editors for LaTeX

LaTeX is not restricted to command line only there are so many graphical based editors available to be used. These GUi based editors provide an easy interface to user so as to do typesetting in an efficient manner. Some of them are listed below:

- Texmaker

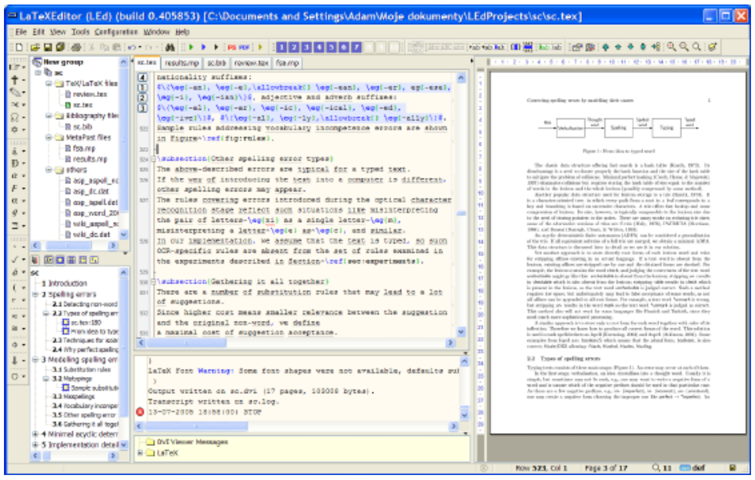


```
(\neg\bot ) = 0$.

Possiamo dire $v_\rho (\bot\rightarrow\neg\bot ) = V$ perché $v_\rho (\bot
)\stackrel{def.}{=} 0$.

Quindi è valida, perché sempre vera.

\paragraph{4)} Soddisfacibile ma non valida.
\end{esercizio}

\begin{esercizio}
Stabilire se i seguenti insiemi di \fbf\ sono soddisfacibili:
\begin{enumerate}
\item $\{ A\wedge B, \neg B\}$
\item $\{ A\rightarrow B, \neg A, C\}$
\item $\{ A\vee B, \neg B\vee C, \neg C\vee A\}$
\end{enumerate}

\paragraph{1)}
L'insieme non è soddisfacibile.

Supponiamo per assurdo che $\exists \rho\: |\: v_\rho (A\wedge B)=v_\rho (\neg B)
= 1$.

    \begin{equation*}
        \left. \begin{array}{l}
        v_\rho (A\wedge B) = \mathrm{min}[v_\rho (A), v_\rho(B)] = 1\\
        v_\rho (\neg B) = 1-v_\rho (B) = 1\\
        \end{array} \right\}
        \Rightarrow
        \left\{ \begin{array}{l}
        v_\rho (B) = 1\\
        v_\rho (B) = 0\\
        \end{array}\right.\mathbf{!!}
    \end{equation*}

I risultati che otteniamo sono evidentemente non validi, perciò l'insieme non è
soddisfacibile.

\paragraph{2)}
```

Figure 4.15: Texmaker, A Graphical LaTeX Editor

- LEd

And many more but the preferred method to produce LaTeX document is through console mode only.


LaTeXEditor (LEd) (build 0.405853) [C:\Documents and Settings\Adam\Moje dokumenty\LEdProjects\sc\sc.tex]
sc.tex
results.mp
sc.bib
review.tex
fsa.mp
DVI Viewer Messages
LaTeX

```
nationality suffixes:
$\(\eg{-an}, \eg{-e},\allowbreak{} \eg{-ean}, \eg{-er}, eg{-ese},
\eg{-i}, \eg{-ian}\)$, adjective and adverb suffixes:
$\(\eg{-al}, \eg{-ar}, \eg{-ic}, \eg{-ical}, \eg{-ed},
\eg{-ive}\)$, $\(\eg{-al}, \eg{-ly},\allowbreak{} \eg{-ally}\)$.
Sample rules addressing vocabulary incompetence errors are shown
in Figure~\ref{fig:rules}.

\subsection{Other spelling error types}
The above-described errors are typical for a typed text.
If the way of introducing the text into a computer is different,
other spelling errors may appear.
The rules covering errors introduced during the optical character
recognition stage reflect such situations like misinterpreting
the pair of letters~\eg{rn} as a single letter~\eg{m},
misinterpreting a letter~\eg{e} as~\eg{c}, and similar.
In our implementation, we assume that the text is typed, so such
OCR-specific rules are absent from the set of rules examined in
the experiments described in Section~\ref{sec:experiments}.

\subsection{Gathering it all together}
There are a number of substitution rules that may lead to a lot
of suggestions.
Since higher cost means smaller relevance between the suggestion
and the original non-word, we define
a maximal cost of suggestion acceptance.
```

```
)
LaTeX Font Warning: Some font shapes were not available, defaults sub
 )
Output written on sc.dvi (17 pages, 102008 bytes).
Transcript written on sc.log.
13-07-2008 18:58:00: STOP
```

Figure 4.16: LEd, A Graphical LATEX Editor

4.7.4 Pdfscreen LATEX

There are some packages that can help to have unified document using LATEX. Example of such a package is pdfscreen that let the user view its document in two forms-print and screen. Print for hard copy and screen for viewing your document on screen. Download this package from www.ctan.org/tex-archive/macros/latex/contrib/pdfscreen/.
Then install it using above mention method.

Just changing print to screen gives an entirely different view. But for working of pdfscreen another package required are comment and fancybox.

The fancybox package provides several different styles of boxes for framing and rotating content in your document. Fancybox provides commands that produce square-cornered boxes with single or double lines, boxes with shadows, and round-cornered boxes with normal or bold lines. You can box mathematics, floats, center, flushleft, and flushright, lists, and pages.

Whereas comments package selectively include/excludes portions of text. The comment package allows you to declare areas of a document to be included or excluded. One need to make these declarations in the preamble of your file. The package uses a method for exclusion that is pretty robust, and can cope with ill-formed bunches of text.

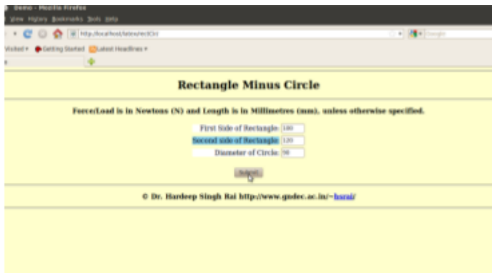
So these extra packages needed to be installed on system for the proper working of pdfscreen package.

4.7.5 Web based graphic generation using LATEX

LATEX is also useful when there is need of generating the graphics from browser. For example to draw a circle by just entering its radius in html input box. So this kind A of project can be conveniently handled using LATEX. Basic idea behind this generation process is that when user clicks on submit button after entering radius a script will run that enter the radius in already made .tex file and recompiles it on server and makes its pdf and postscript file. After that user can view those files by clicking on link provided to view the files. See some screen shots of such a graphic generation project made by Dr. H.S. Rai:

So here in the above input page which is also the index page user can enter input for length of rectangle, breadth of rectangle and for radius of circle after that user can submit the values. After the values get submitted a script get runs by php code at server side. This script first enters the dimensions of rectangle and circle that were selected by user in to an already existing .tex file and replace with the older dimensions there. After that script recompiles the the tex file and make it available for user.

In above figure it gets clear that .tex file has been compiled and pdf and postscript files are available to user and user can download the graphics so produced. Hence graphics can be generated in LaTeX through web interface.


Rectangle Minus Circle

Force/Load is in Newtons (N) and Length is in Millimetres (mm), unless otherwise specified.

First Side of Rectangle: 180

Second side of Rectangle: 120

Diameter of Circle: 98

© Dr. Hardeep Singh Rai http://www.gndec.ac.in/~hsrai/

Figure 4.17: Web based graphic generation using LaTeX(input page)

4.8 Introduction to Github

Figure 4.18: Github Logo

GitHub is a Git repository web-based hosting service which offers all of the functionality of Git as well as adding many of its own features. Unlike Git which is strictly a command-line tool, Github provides a web-based graphical interface and desktop as well as mobile integration. It also provides access control and several collaboration features such as wikis, task management, and bug tracking and feature requests for every project.

GitHub offers both paid plans for private repto handle everything from small to very large projects with speed and efficiency. ositories, and free accounts, which are usually used to host open source software projects. As of 2014, Github reports having over 3.4 million users, making it the largest code host in the world.

GitHub has become such a staple amongst the open-source development community that many developers have begun considering it a replacement for a conventional resume and some employers require applications to provide a link to and have an active contributing GitHub account in order to qualify for a job.

The Git feature that really makes it stand apart from nearly every other Source Code Management (SCM) out there is its branching model.

Git allows and encourages you to have multiple local branches that can be entirely independent of each other. The creation, merging, and deletion of those lines of development takes seconds.

This means that you can do things like:

- Frictionless Context Switching.
  Create a branch to try out an idea, commit a few times, switch back to where you branched from, apply a patch, switch back to where you are experimenting, and merge it in.
- Role-Based Codelines.
  Have a branch that always contains only what goes to production, another that you merge work into for testing, and several smaller ones for day to day work.
- Feature Based Workflow.
  Create new branches for each new feature you're working on so you can seamlessly switch back and forth between them, then delete each branch when that feature gets merged into your main line.
- Disposable Experimentation.
  Create a branch to experiment in, realize it's not going to work, and just delete it - abandoning the workwith nobody else ever seeing it (even if you've pushed other branches in the meantime).

Notably, when you push to a remote repository, you do not have to push all of your branches. You can choose to share just one of your branches, a few of them, or all of them. This tends to free people to try new ideas without worrying about having to plan how and when they are going to merge it in or share it with others.

There are ways to accomplish some of this with other systems, but the work involved is much more difficult and error-prone. Git makes this process incredibly easy and it changes the way most developers work when they learn it.

4.8.1 What is Git?

Git is a distributed revision control and source code management (SCM) system with an emphasis on speed, data integrity, and support for distributed, non-linear workflows. Git was initially designed and developed by Linus Torvalds for Linux kernel development in 2005, and has since

Figure 4.19: Git Logo

become the most widely adopted version control system for software development.

As with most other distributed revision control systems, and unlike most clientserver systems, every Git working directory is a full-fledged repository with complete history and full version-tracking capabilities, independent of network access or a central server. Like the Linux kernel, Git is free and open source software distributed under the terms of the GNU General Public License version 2 to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

4.8.2 Installation of Git

Installation of git is a very easy process. The current git version is: 2.0.4. Type the commands in the terminal:

```
$ sudo apt-get update

$ sudo apt-get install git
```

This will install the git on your pc or laptop.

4.8.3 Various Git Commands

Git is the open source distributed version control system that facilitates GitHub activities on your laptop or desktop. The commonly used Git command line instructions are:-

4.8.3.1 Create Repositories

Start a new repository or obtain from an exiting URL

$ git init [project-name]
Creates a new local repository with the specified name

$ git clone [url]
Downloads a project and its entire version history

4.8.3.2 Make Changes

Review edits and craft a commit transaction

$ git status
Lists all new or modified files to be committed

$ git diff
Shows file differences not yet staged

$ git add [file]
Snapshots the file in preparation for versioning

$ git reset [file]
Unstages the file, but preserve its contents

$ git commit -m "[descriptive message "]
Records file snapshots permanently in version history

4.8.3.3 Group Changes

Name a series of commits and combine completed efforts

$ git branch
Lists all local branches in the current repository

$ git branch [branch-name]
Creates a new branch

$ git checkout [branch-name]
Switches to the specified branch and updates the working directory

$ git merge [branch]
Combines the specified branchs history into the current branch

$ git branch -d [branch-name]
Deletes the specified branch

4.8.3.4 Save Fragments

Shelve and restore incomplete changes

$ git stash
Temporarily stores all modified tracked files

$ git stash pop
Restores the most recently stashed files

$ git stash list
Lists all stashed changesets

$ git stash drop
Discards the most recently stashed changeset

4.8.3.5 Synchronize Changes

Register a repository bookmark and exchange version history

$ git fetch [bookmark]
Downloads all history from the repository bookmark

$ git merge [bookmark /[branch]]
Combines bookmarks branch into current local branch

$ git push [alias [branch]]
Uploads all local branch commits to GitHub

$ git pull
Downloads bookmark history and incorporates changes

4.9 Implementation

Development of SIM started with development in phases which focus on particular need of project. Various phases and their detail are given below -:

- Phase I (C++) -:
  During Phase I, we wrote code in C++ to parse Staad Pro file to the database (Mysql).
- Phase II (Mysql) -:
  During Phase II, we make the schema of Mysql tables in which all the attributes of Staad pro are present. Phase I and II are very much interlinked because C++ code i.e. act as a parser and Mysql used to store the output data of the parser.
- Phase III (Django) -:
  During phase III, we record small macros (interpreted code) in FreeCAD and then transfer that macro in the form of class and objects.

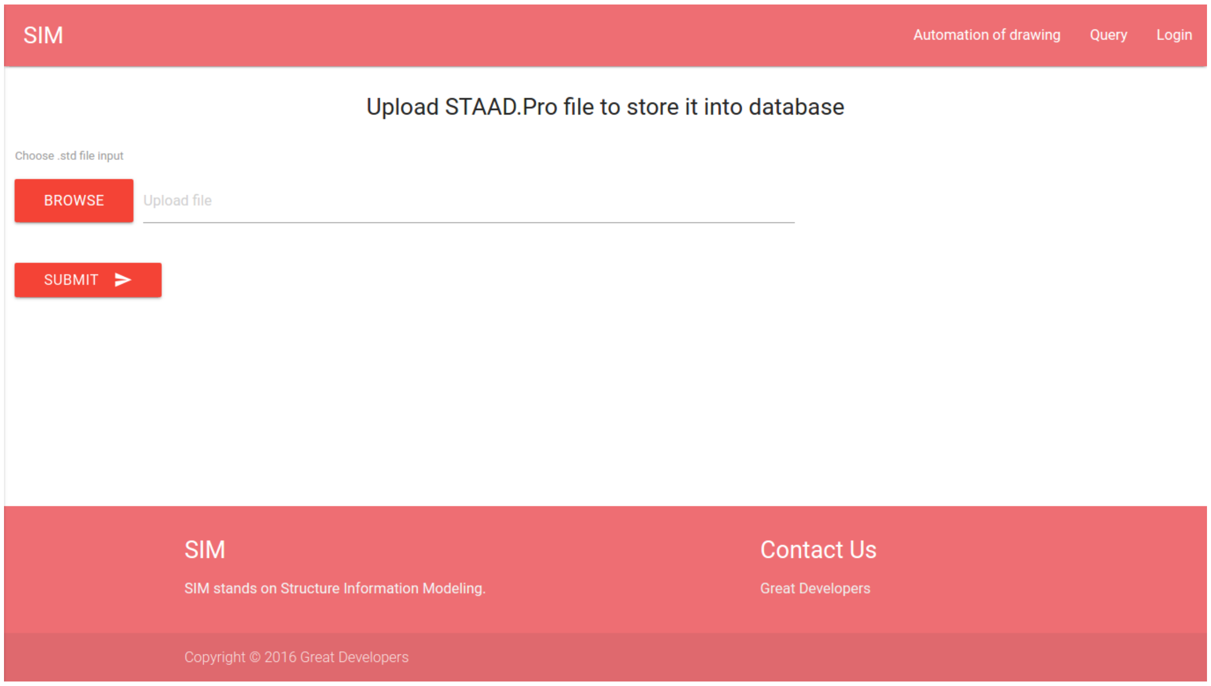
- Phase IV (Django) -:
  During phase IV, we provided web interface to this software using Django. Djanog was used to get input from user and write some.csv file and after that FreeCAD macro will run automatically and give its respectively output in the form of PDF, SVG, .fcstd.

- Phase V (Rendering 3D model) -:
  During phase V, we convert the .fcstd file to WebGL format file to rendering 3D model on the browser screen. Sed command is used for changing the value some constraint.

- Phase VI (Testing and documentation) -:
  During final phase, we tested the software for various conditions and then applied required error control and messaging mechanism. Also in this phase, we documented the project(developers documentation and README.md) using doxygen and wrote the report for this software.

4.9.1 Upload STAAD.Pro file

SIM communicates with the database back at the server end and interact to store and retrieve information. A processing of the file is considered complete only if all of the file has been parsed, it database objects has been created by the corresponding function and the objects have been properly transferred to the database at the back end. If this is not the case then no changes shall be made to the database and its consistent state must be maintained.



Upload STAAD.Pro file to store it into database

Choose .std file input

SIM

SIM stands on Structure Information Modeling.

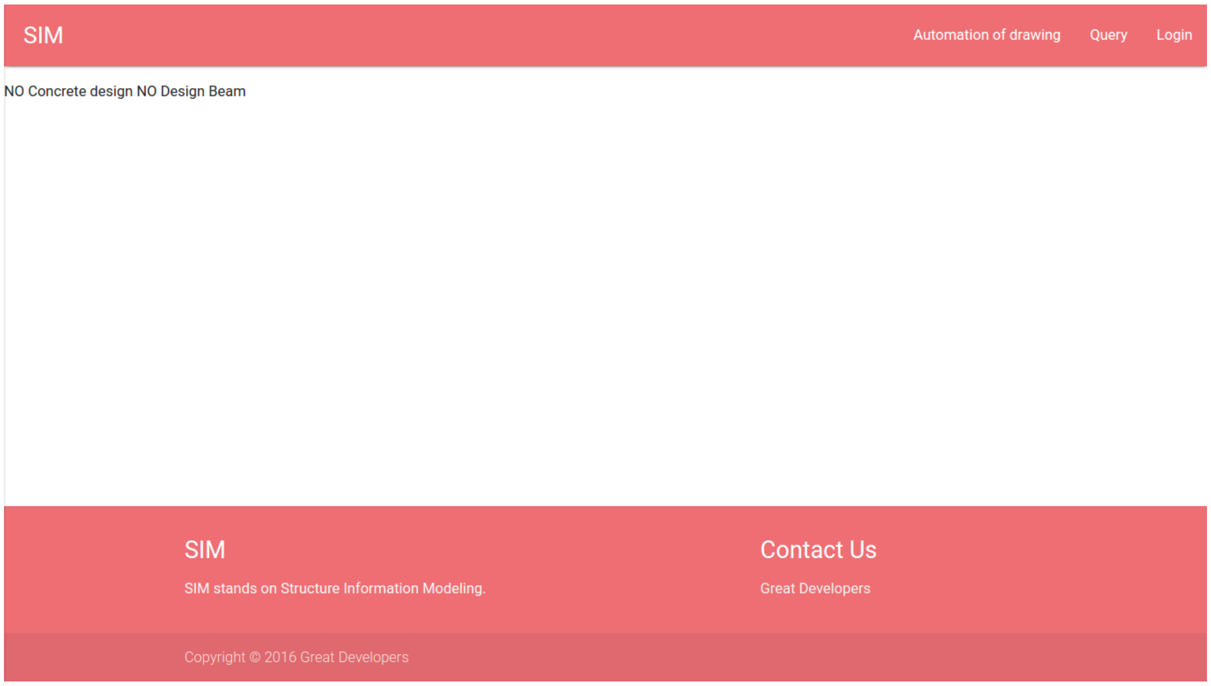
Contact Us

Great Developers

Copyright © 2016 Great Developers

Figure 4.20: Front page

If the system recognizes that the file in consideration is not correct then it must point that out to the user and also be able to tell that in what manner is the file wrong. For example, if the file is wrong syntactically then the correct syntactic suggestion must be provided to the user. If the file is not complete then simply a message must be passed to the user.


SIM
Automation of drawing
Query
Login

NO Concrete design NO Design Beam

SIM
SIM stands on Structure Information Modeling.
Contact Us
Great Developers
Copyright © 2016 Great Developers

Figure 4.21: Result show on submitted the Staad.Pro file

4.9.2 Query

The user must be able to apply queries on the information stored in the databases. All the data stored in the database is shown in the tabular form with all the fields that are related to the respective element as shown in Fig 4.22.

Result of the query is shown Fig. 4.23.

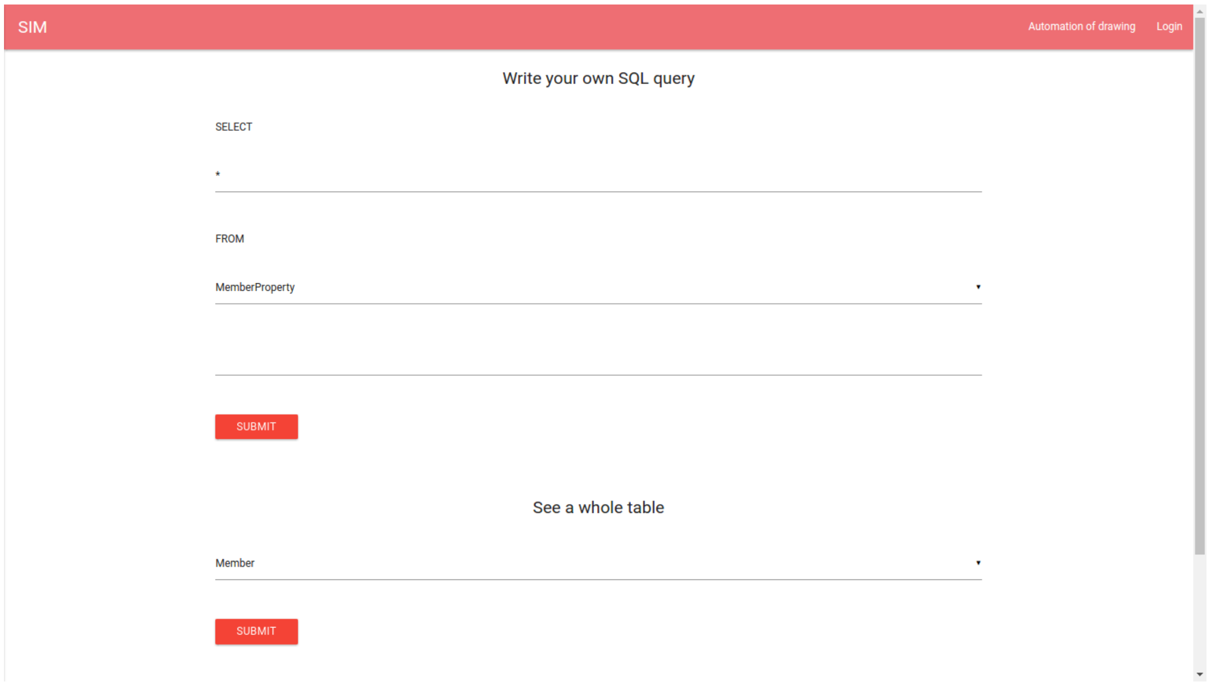
4.9.3 Automation of Drawings

4.9.3.1 Enter Building specifications

The user makes the drawings of different views of the building on the drawing sheets. The user will be able to enter the specifications of the building through the web browser (Fig. 4.24) and on the back-end, the FreeCAD macros will use those input values to draw the drawings of different views of the building on different drawing sheets. The output of the same can then be taken by a user in SVG, PDF and fcstd formats.

4.9.3.2 Download project as zip

After submitting the form this page will redirect to download page(Fig. 3.6). Here the user will see all the submitted value of in form (Fig. 4.25) and download its project. The zip file will contain different views (like side view, front view, a top view and many sectional views according to the building stories) of the building in PDF as well in SVG format. It also contains .fcstd file (FreeCAD project file) Fig. 4.26. By using .fcstd file the user will modify the existing project and can perform a wide range of operations.



Write your own SQL query

SELECT

*

FROM

MemberProperty

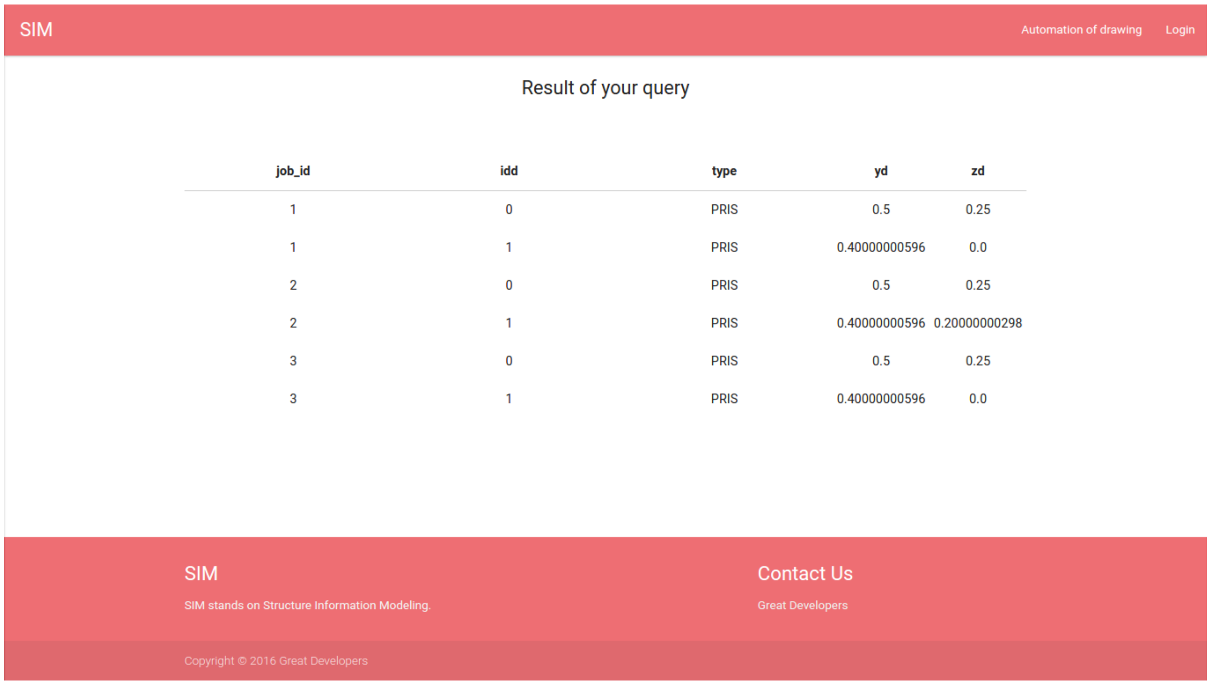
SUBMIT

See a whole table

Member

SUBMIT

Figure 4.22: Query page



Result of your query

job_id	idd	type	yd	zd
1	0	PRIS	0.5	0.25
1	1	PRIS	0.40000000596	0.0
2	0	PRIS	0.5	0.25
2	1	PRIS	0.40000000596	0.20000000298
3	0	PRIS	0.5	0.25
3	1	PRIS	0.40000000596	0.0

SIM

SIM stands on Structure Information Modeling.

Contact Us

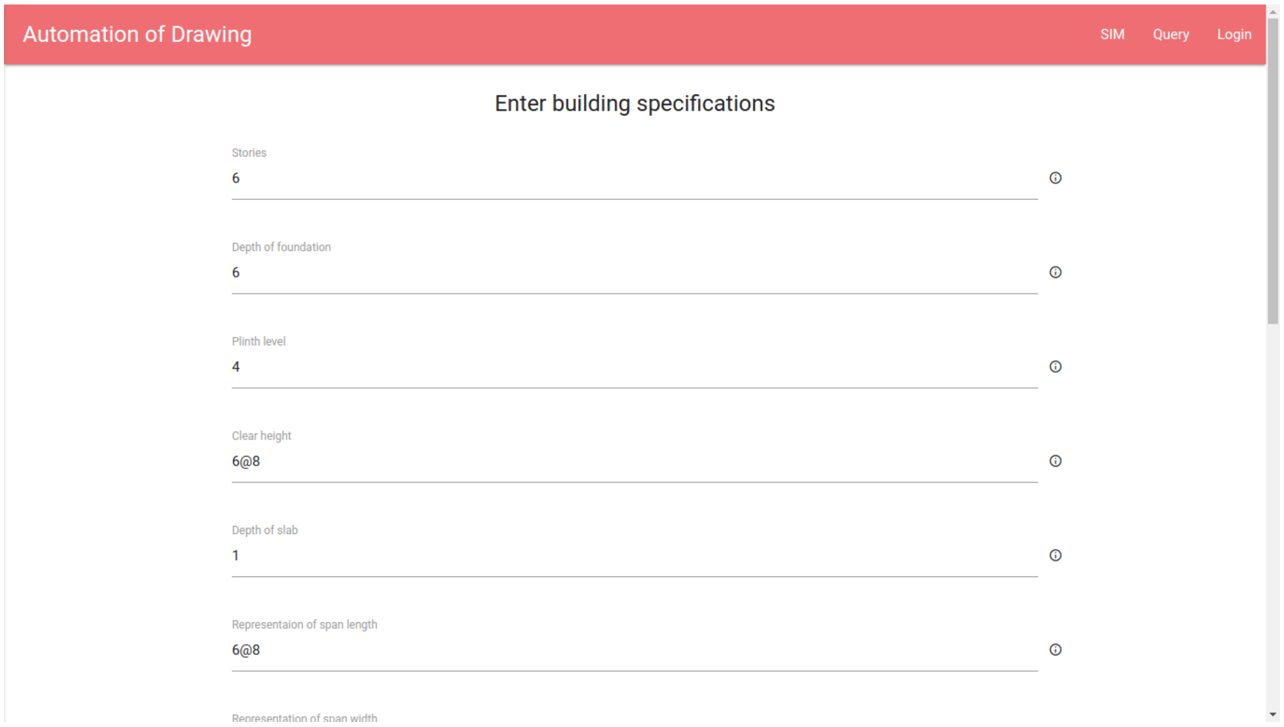
Great Developers

Copyright © 2016 Great Developers

Figure 4.23: Result of query

4.9.3.3 View 3D model

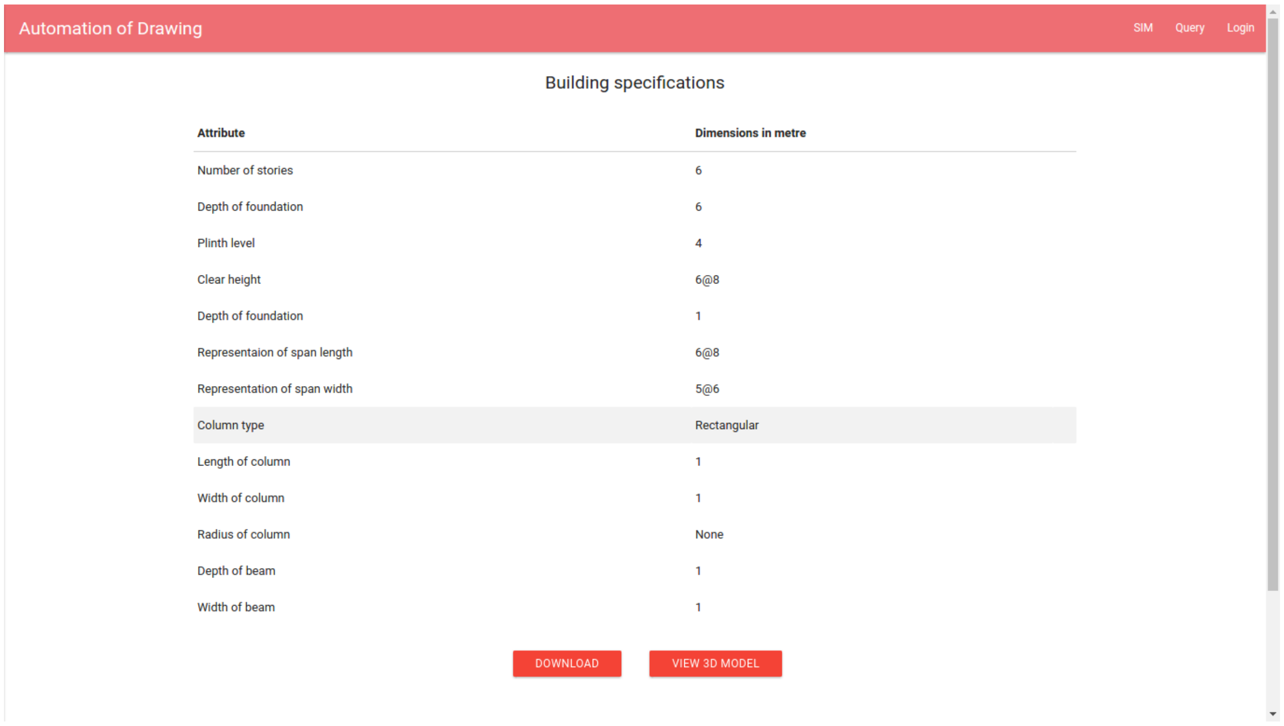
This feature will render the 3D model of the building on the browser screen. In this process, FreeCAD exporter convert .fcstd file to WebGL format (Web Graphics Library is a JavaScript



Enter building specifications

Stories	6
Depth of foundation	6
Plinth level	4
Clear height	6@8
Depth of slab	1
Representaion of span length	6@8
Representation of span width	

Figure 4.24: Form to enter the specifications of the building

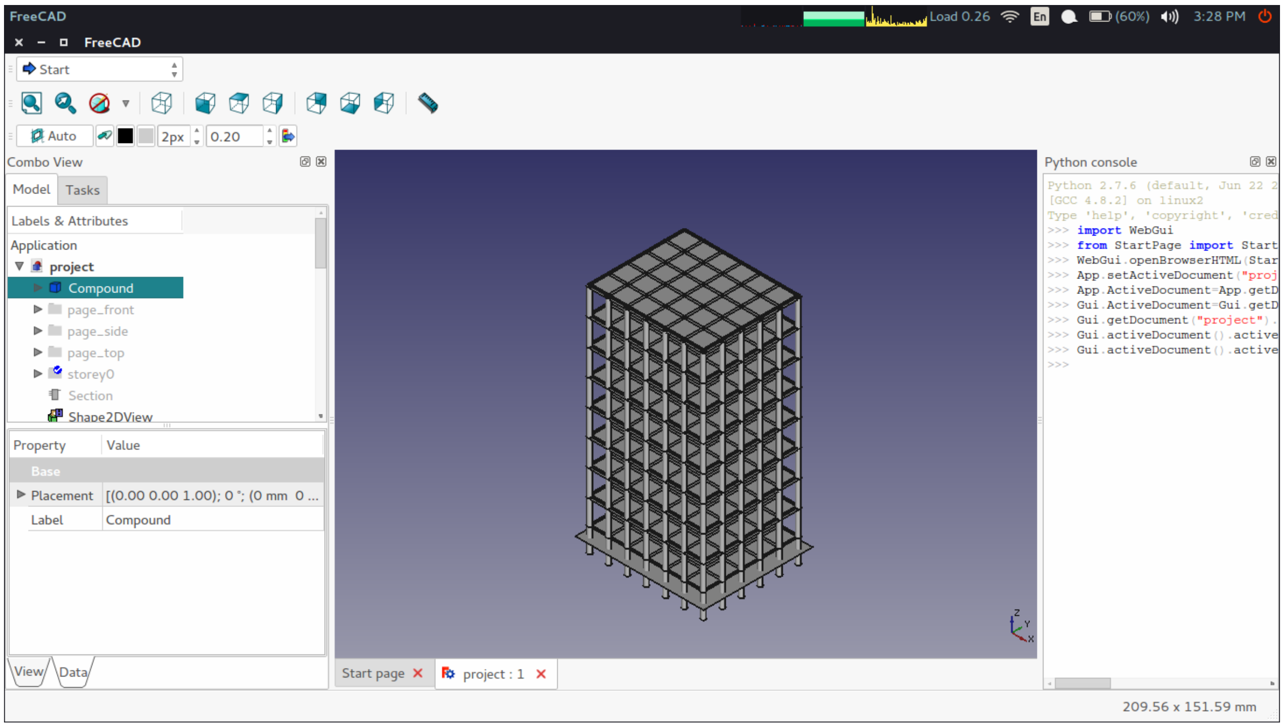


Building specifications

Attribute	Dimensions in metre
Number of stories	6
Depth of foundation	6
Plinth level	4
Clear height	6@8
Depth of foundation	1
Representaion of span length	6@8
Representation of span width	5@6
Column type	Rectangular
Length of column	1
Width of column	1
Radius of column	None
Depth of beam	1
Width of beam	1

Figure 4.25: Form to enter the specifications of the building

API for rendering interactive 3D and 2D graphics within any compatible web browser without the use of plug-ins).



Property	Value
Base	
Placement	[(0.00 0.00 1.00); 0 °; (0 mm 0 ...
Label	Compound

```
Python 2.7.6 (default, Jun 22 2
[GCC 4.8.2] on linux2
Type 'help', 'copyright', 'cred
>>> import WebGui
>>> from StartPage import Start
>>> WebGui.openBrowserHTML(Star
>>> App.setActiveDocument("proj
>>> App.ActiveDocument=App.getD
>>> Gui.ActiveDocument=Gui.getD
>>> Gui.getDocument("project").
>>> Gui.activeDocument().active
>>> Gui.activeDocument().active
>>>
```

Figure 4.26: Opening .fcstd file in FreeCAD

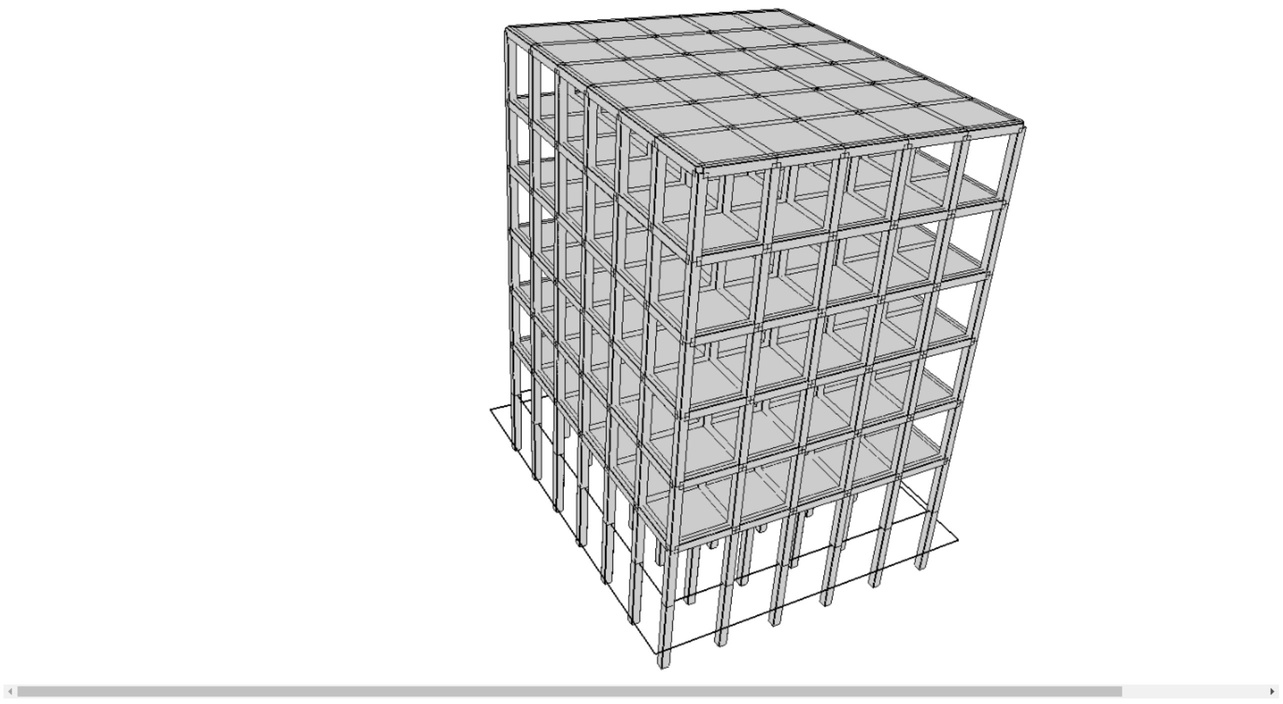


Figure 4.27: Admin page

4.10 Adminstration Login

One of the most powerful parts of SIM is the admin interface Fig . It reads metadata from your models to provide a quick, model-centric interface where trusted users can manage content on your site. The admins recommended use is limited to an organisations internal management tool. Its not intended for building your entire front end around.

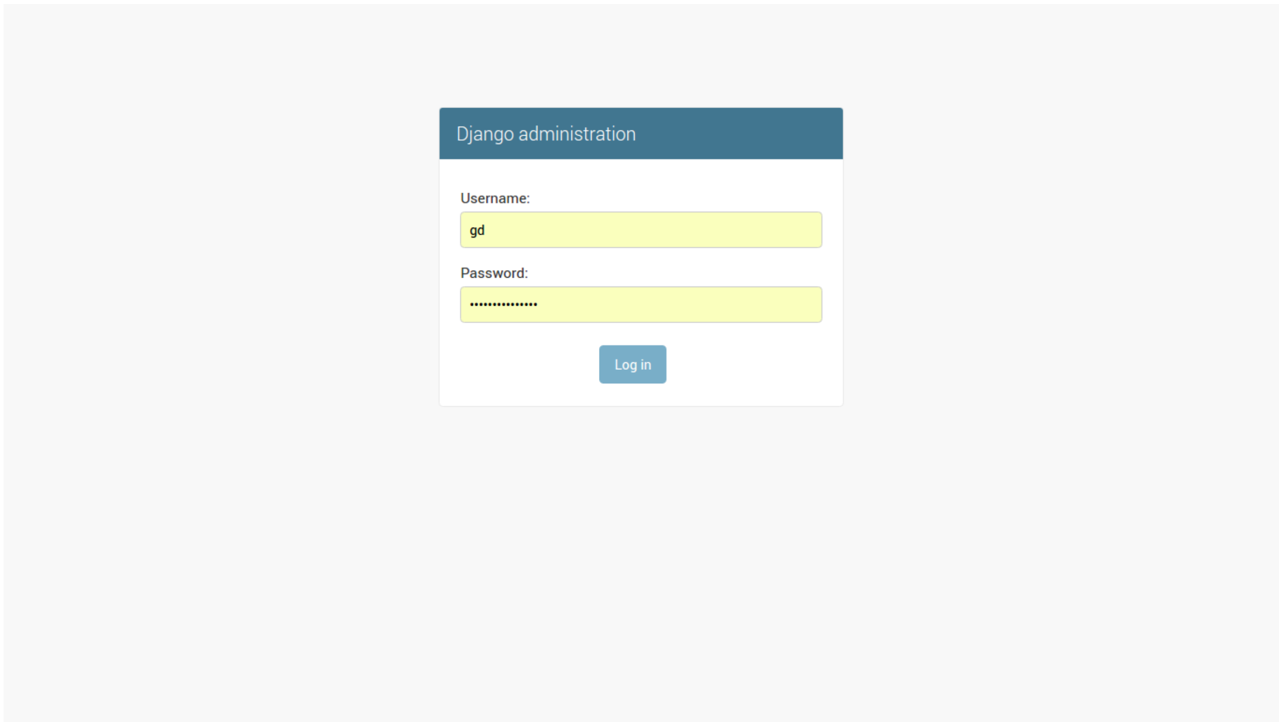
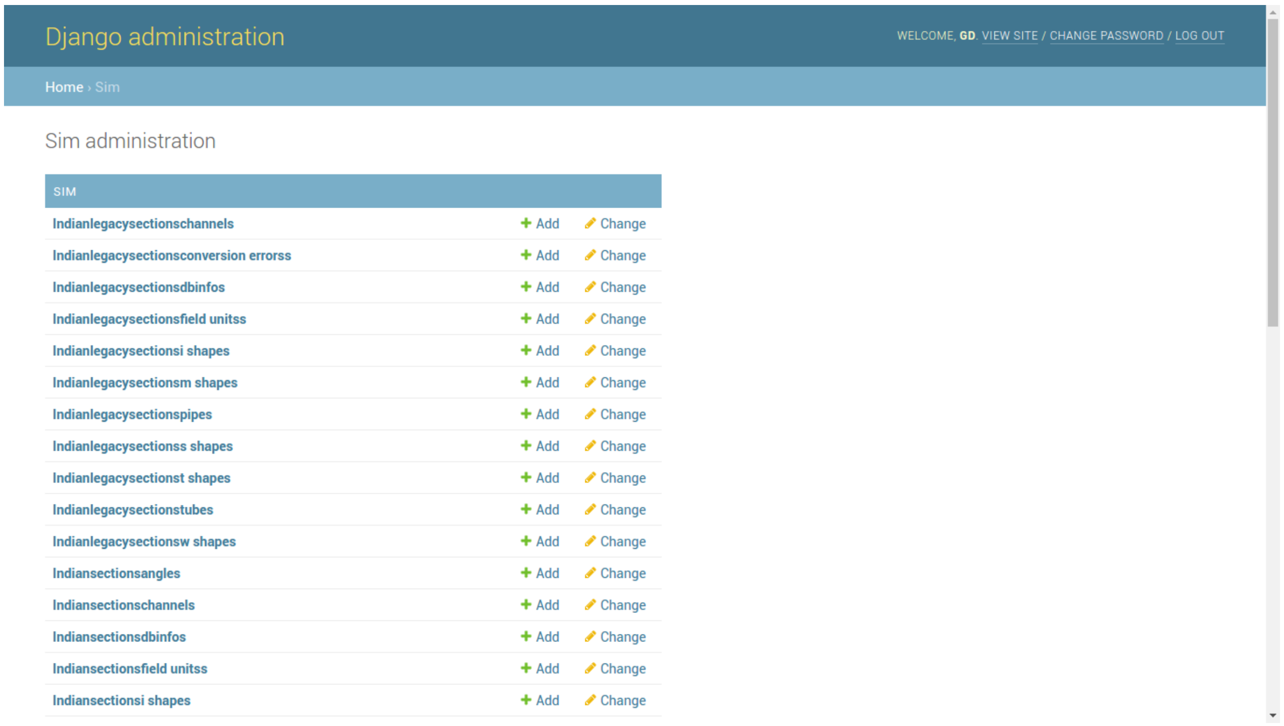


Figure 4.28: Login page

The admin has many hooks for customization, but beware of trying to use those hooks exclusively. If you need to provide a more process-centric interface that abstracts away the implementation details of database tables and fields, then its probably time to write your own views.



Django administration

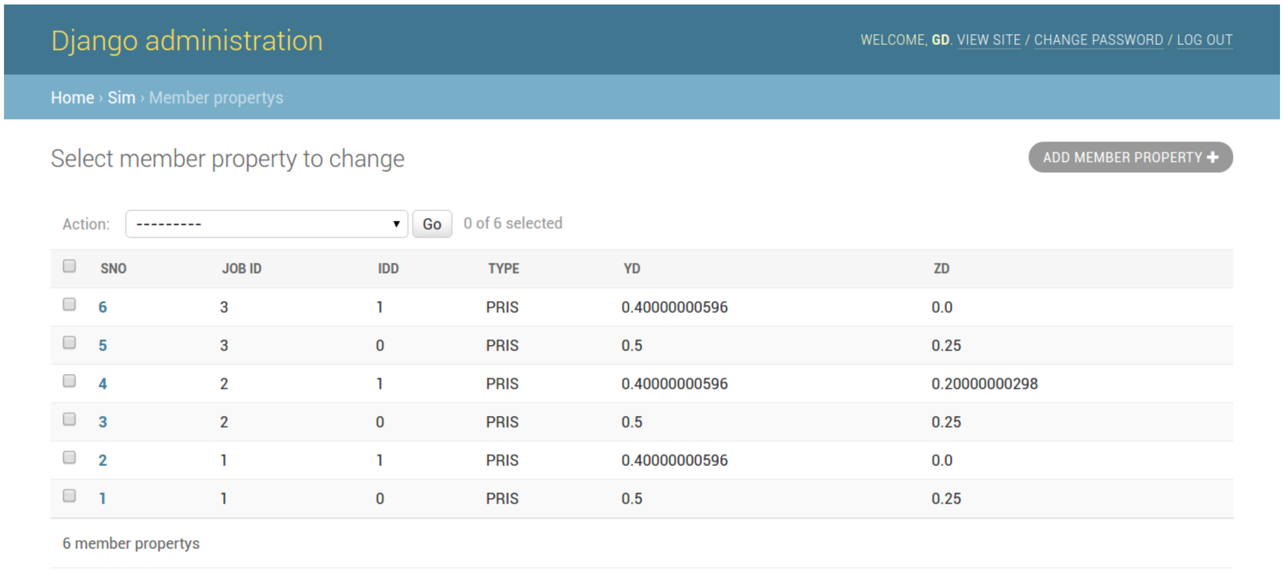
WELCOME, GD. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Sim

Sim administration

SIM		
Indianlegacysectionschannels	Add	Change
Indianlegacysectionsconversion errorss	Add	Change
Indianlegacysectionsdbinfos	Add	Change
Indianlegacysectionsfield unitss	Add	Change
Indianlegacysectionsi shapes	Add	Change
Indianlegacysectionsm shapes	Add	Change
Indianlegacysectionspipes	Add	Change
Indianlegacysectionss shapes	Add	Change
Indianlegacysectionst shapes	Add	Change
Indianlegacysectionstubes	Add	Change
Indianlegacysectionsw shapes	Add	Change
Indiansectionsangles	Add	Change
Indiansectionschannels	Add	Change
Indiansectionsdbinfos	Add	Change
Indiansectionsfield unitss	Add	Change
Indiansectionsi shapes	Add	Change

Figure 4.29: Entities of Staad Pro



Django administration

WELCOME, GD. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Sim › Member propertys

Select member property to change

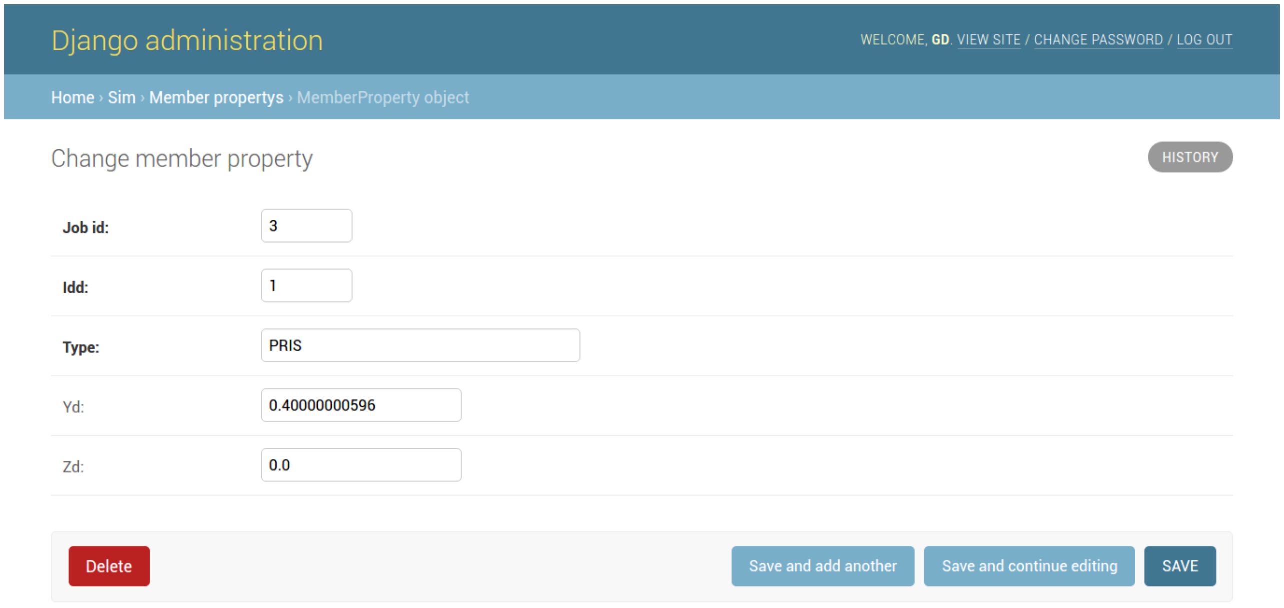
ADD MEMBER PROPERTY

Action: --------- Go 0 of 6 selected

SNO	JOB ID	IDD	TYPE	YD	ZD
6	3	1	PRIS	0.40000000596	0.0
5	3	0	PRIS	0.5	0.25
4	2	1	PRIS	0.40000000596	0.20000000298
3	2	0	PRIS	0.5	0.25
2	1	1	PRIS	0.40000000596	0.0
1	1	0	PRIS	0.5	0.25

6 member propertys

Figure 4.30: Attributes of each entity


Django administration
WELCOME, GD. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Sim › Member propertys › MemberProperty object
HISTORY
Delete
Save and add another
Save and continue editing
SAVE

Change member property

Job id:	3
Idd:	1
Type:	PRIS
Yd:	0.40000000596
Zd:	0.0

Figure 4.31: Modification, creation and deleting entity

4.11 Testing

Testing a program consists of providing the program with a set of test inputs (or test cases) and observing if the program behaves as expected. If the program fails to behave as expected, then the conditions under which failure occurs are noted for later debugging and correction.

This software had been taken through rigorous test to fully found potential causes of error and system failure and full focus have been given to cover all possible exceptions that can occur and cause failure of the software. As this software is based on intensive background process it have been taken care that if correct input and email address are given then processing of user job can even continue or a least automatically restart even after server shuts down or even crash.

SIM communicates with the database back at the server end and interact to store and retrieve information. A processing of the file is considered complete only if all of the file has been parsed, it database objects has been created by the corresponding function and the objects have been properly transferred to the database at the back end. If this is not the case then no changes shall be made to the database and its consistent state must be maintained.

If the system recognizes that the file in consideration is not correct then it must point that out to the user and also be able to tell that in what manner is the file wrong. For example, if the file is wrong syntactically then the correct syntactic suggestion must be provided to the user. If the file is not complete then simply a message must be passed to the user.

Test Case ID	Test Case Name	Description	Expected Result	Status
MA001	Change file	The user must be able to change the working file without having to restart the system every time. This means that there must be navigational features in the system to move back and forth with different files in the same session.	Without having to create a new session the user must be able to change the working files.	Passed
MA002	valid file	The file provided by the user maybe half complete and/or with errors or outright empty. So the system must be able recognize these issues and offer the correct error message in response.	The system must accept those files that are correct and complete otherwise proper redirection to the home page with an error message must be provided.	Passed
MA003	query proccessing	The user must be able to apply queries on the information stored in the databases.	All user queries that are valid must be answered properly	Passed
MA004	File retrival	Just like in a query when the user retrieves certain information about the structure from the database, the user should be able if he/she wishes to, construct a whole, complete file from the information stored in the database.	The whole file must be created from the informtion in the database	Passed

Table 4.1: Integration testing

CHAPTER 5 CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

SIM is a very efficient application but it does nothing but prepare notes for the design and making drawing of the building. Yes, it reads the script, parses data out of it and then stores it in a format that the user can easily retreat, recall and query. It can be used by Civil Engineers and M.Tech. students and even layman. It's less time consuming and user-friendly which let the User work in batch mode and free him from all the installation process. It is a web application that can be accessed from a number of devices. The responsive User Interface makes it easy for the users to operate it. Many efforts were made to ease the usage for the users. Hence, it is expected to be work properly in different conditions. But any future bug reports or improvements are always welcomed and will be processed happily.

I learn a lot by working on this project. During this period I got to learn a vast number of technologies. These are listed below: Operating system: Ubuntu Language used: Python, C++, HTML, CSS, shellscript Framework: Django, Materialize Technogloy: LaTeX, Djanog, Git Software: FreeCAD, Inkscape So during this project I learn all the above things. Above all I got to know how software is developed and how much work and attention to details is required in building even the most basic of components of any project. Planning, designing, developing code, working in a team, testing, etc. These are all very precious lessons in themselves. Aside from all above I got go know about various methods like -:

1. Embedding and using different tech in one software.
2. How to work like in group for development of software.

Beside these technology used in project I also get to know some other tech also like -:

1. Flex and Bison
2. opensshserver
3. reveal.md, impress.js (for making presentations)

5.2 Future Scope

This software being a open source have allot of scope for future improvements and additions as other individuals can also contibute in it and add additional functionality like-:

1. Convert parser code (presently it is in C++) to Flex and Bison language
2. Making builing from Staad Pro file
3. Convert Staad Pro file to IFC (Industry Foundation Classes) and then rendering IFC with different opensource viewer on the browser.
4. BIM integration
5. Adding dimensions to the output drawing

BIBLIOGRAPHY

[1] Structure Information Modeling (SIM), https://github.com/amrit3701/Sim

[2] FreeCAD, https://github.com/FreeCAD/FreeCAD

[3] LaTeX Beginner's Guide By Stefan Kottwitz [Pact Publishing]

[4] My Blog, https://amritpals.com

[5] My Github Profile, http://github.com/amrit3701