FOSSEE Winter Internship Report

On

Logic Development of Purlin Module and Lacing Module for Osdag

Submitted by

Koustav Bhattacharjee

B.E. 3rd Year, Department of Mechanical Engineering

Jadavpur University

Kolkata

Under the Guidance of

Prof. Siddhartha Ghosh

Department of Civil Engineering

Indian Institute of Technology Bombay

Mentors:

Ajmal Babu M S

Parth Karia

Ajinkya Dahale

January 4, 2025

Acknowledgments

- I am grateful to my guide and mentors who hovered my mind through the experience, motivating me to contribute to open-source through my engineering knowledge.
- I have my sincere gratitude to **Ajmal Babu M. S.**, **Ajinkya Dahale**, and **Parth Karia**, for their constant support during the internship and for teaching us new skills that have greatly enriched our learning experience.
- I am thankful to Osdag Principal Investigator (PI) **Prof. Siddhartha Ghosh**, Department of Civil Engineering at IIT Bombay, , for his invaluable guidance and support throughout the project.
- I extend my sincere thanks to **Prof. Kannan M. Moudgalya,** Principal Investigator (PI) of the FOSSEE Project, Department of Chemical Engineering, IIT Bombay, for providing me with this opportunity and for his continuous encouragement.
- I extend my special thanks to the FOSSEE Managers, **Ms. Usha Viswanathan** and **Ms. Vineeta Parmar**, along with their entire team, for their assistance and coordination, which significantly contributed to the project's advancement.
- I would like to acknowledge the support of the **National Mission on Education** through **Information and Communication Technology (ICT)**, under the **Ministry of Education (MoE), Government of India**. Their contributions have been instrumental in facilitating this project, enhancing access to quality educational resources, and promoting innovative teaching methodologies across various institutions.

- Lastly, I want to convey my sincere gratitude to my colleagues Debayan Ghosh and Shubham Kumar for their collaboration and support during this internship.
- I would like to express my sincere gratitude to my professors from the **Mechanical Engineering Department** at **Jadavpur University** for their invaluable guidance and support throughout my academic journey. Their expertise and dedication have significantly contributed to my learning experience and personal growth.

Contents

Chapter 1

Introduction

1.1 National Mission in Education through ICT

The National Mission on Education through ICT (NMEICT) is a scheme under the Department of Higher Education, Ministry of Education, Government of India. It aims to leverage the potential of ICT to enhance teaching and learning in Higher Education Institutions in an anytime-anywhere mode.

The mission aligns with the three cardinal principles of the Education Policy—**access, equity, and quality**—by:

- Providing connectivity and affordable access devices for learners and institutions.
- Generating high-quality e-content free of cost.

NMEICT seeks to bridge the digital divide by empowering learners and teachers in urban and rural areas, fostering inclusivity in the knowledge economy. Key focus areas include:

- Development of e-learning pedagogies and virtual laboratories.
- Online testing, certification, and mentorship through accessible platforms like EduSAT and DTH.
- Training and empowering teachers to adopt ICT-based teaching methods.

For further details, visit the official website: www.nmeict.ac.in.

1.1.1 ICT Initiatives of MoE

The Ministry of Education (MoE) has launched several ICT initiatives aimed at students, researchers, and institutions. The table below summarizes the key details:

No.	Resource	For Students/Researchers	For Institutions
		Audio-Video e-content	
1	SWAYAM	Earn credit via online courses	Develop and host courses; accept credits
2	SWAYAMPRABHA	Access 24x7 TV programs	Enable SWAYAMPRABHA viewing facilities
		Digital Content Access	
3	National Digital Library	Access e-content in multiple disciplines	List e-content; form NDL Clubs
4	e-PG Pathshala	Access free books and e-content	Host e-books
5	Shodhganga	Access Indian research theses	List institutional theses
6	e-ShodhSindhu	Access full-text e-resources	Access e-resources for institutions
		Hands-on Learning	
7	e-Yantra	Hands-on embedded systems training	Create e-Yantra labs with IIT Bombay
8	FOSSEE	Volunteer for open-source software	Run labs with open-source software
9	Spoken Tutorial	Learn IT skills via tutorials	Provide self-learning IT content
10	Virtual Labs	Perform online experiments	Develop curriculum-based experiments
		E-Governance	
11	SAMARTH ERP	Manage student lifecycle digitally	Enable institutional e-governance
		Tracking and Research Tools	
12	VIDWAN	Register and access experts	Monitor faculty research outcomes
13	Shodh Shuddhi	Ensure plagiarism-free work	Improve research quality and reputation
14	Academic Bank of Credits	Store and transfer credits	Facilitate credit redemption

Table 1.1: Summary of ICT Initiatives by the Ministry of Education

1.2 FOSSEE Project

The FOSSEE (Free/Libre and Open Source Software for Education) project promotes the use of FLOSS tools in academia and research. It is part of the National Mission on Education through Information and Communication Technology (NMEICT), Ministry of Education (MoE), Government of India.

1.2.1 Projects and Activities

The FOSSEE Project supports the use of various FLOSS tools to enhance education and research. Key activities include:

- **Textbook Companion**: Porting solved examples from textbooks using FLOSS.
- **Lab Migration**: Facilitating the migration of proprietary labs to FLOSS alternatives.
- **Niche Software Activities**: Specialized activities to promote niche software tools.
- **Forums**: Providing a collaborative space for users.
- **Workshops and Conferences**: Organizing events to train and inform users.

1.2.2 Fellowships

FOSSEE offers various internship and fellowship opportunities for students:

- Winter Internship
- Summer Fellowship
- Semester-Long Internship

Students from any degree and academic stage can apply for these internships. Selection is based on the completion of screening tasks involving programming, scientific computing, or data collection that benefit the FLOSS community. These tasks are designed to be completed within a week.

For more details, visit the official FOSSEE website.

Figure 1.1: FOSSEE Projects and Activities

1.3 Osdag Software

Osdag (Open steel design and graphics) is a cross-platform, free/libre and open-source software designed for the detailing and design of steel structures based on the Indian Standard IS 800:2007. It allows users to design steel connections, members, and systems through an interactive graphical user interface (GUI) and provides 3D visualizations of designed components. The software enables easy export of CAD models to drafting tools for construction/fabrication drawings, with optimized designs following industry best practices [1, 2, 3]. Built on Python and several Python-based FLOSS tools (e.g., PyQt and PythonOCC), Osdag is licensed under the GNU Lesser General Public License (LGPL) Version 3.

1.3.1 Osdag GUI

The Osdag GUI is designed to be user-friendly and interactive. It consists of

- **Input Dock**: Collects and validates user inputs.
- **Output Dock**: Displays design results after validation.
- **CAD Window**: Displays the 3D CAD model, where users can pan, zoom, and rotate the design.
- **Message Log**: Shows errors, warnings, and suggestions based on design checks.

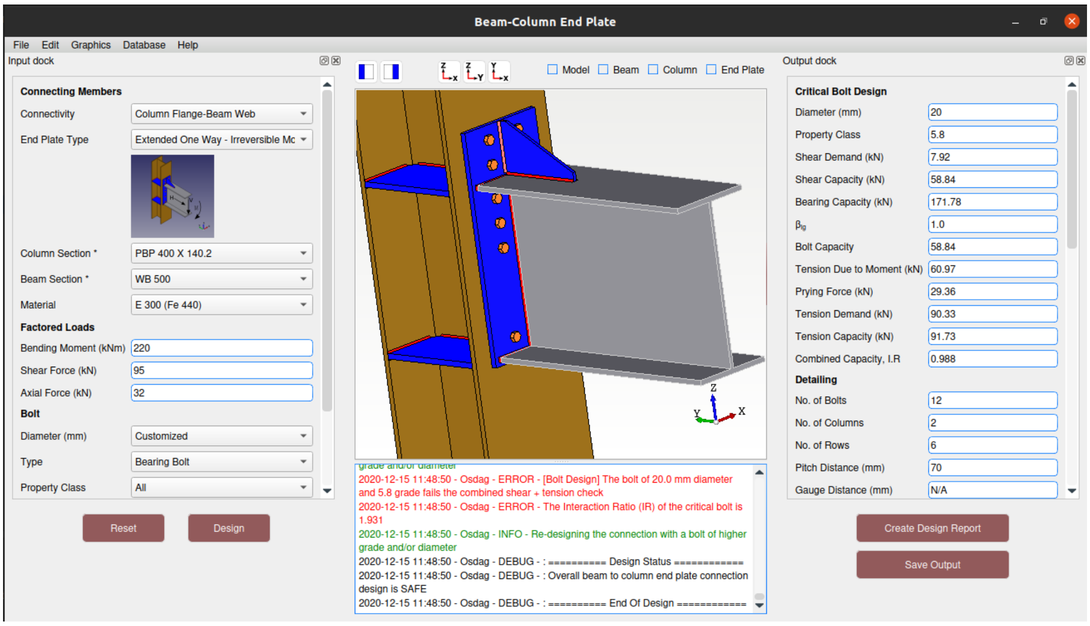

Figure 1.2: Osdag GUI

1.3.2 Features

- **CAD Model**: The 3D CAD model is color-coded and can be saved in multiple formats such as IGS, STL, and STEP.
- **Design Preferences**: Customizes the design process, with advanced users able to set preferences for bolts, welds, and detailing.
- **Design Report**: Creates a detailed report in PDF format, summarizing all checks, calculations, and design details, including any discrepancies.

For more details, visit the official Osdag website.

Chapter 2

Screening Task

2.1 Problem Statement

The screening task involved developing a Python program for the design of flexural members that are laterally supported and subjected to low shear forces. We were supposed to create a program that accurately calculates the necessary design parameters while adhering to relevant engineering standards. Along with that, a comprehensive report was to be prepared, detailing the design methodologies employed, calculations performed, and justifications for the chosen approaches. This task aims to evaluate candidates' programming skills and their understanding of structural design principles, ensuring they can contribute effectively to engineering solutions.

2.2 Tasks Done

2.2.1 Introduction

In this assignment, the design of flexural members has been explored, emphasizing its critical role in structural engineering. The objective has been to ensure that beams can withstand various loads while maintaining safety and serviceability. The fundamental concepts and procedures involved in the design of flexural members, particularly focusing on steel structures, have been outlined.

2.2.2 Understanding the Loads

The types of loads that a member will experience have been understood as essential for structural design. The following categories have been identified:

- **Dead Loads**: Permanent static loads, such as the weight of the structure itself.
- **Live Loads**: Temporary dynamic loads, including occupancy and furniture.
- **Environmental Loads**: Forces due to wind, snow, or seismic activity.

2.2.3 Load Calculations

Load calculations have been performed to determine the magnitude and distribution of these forces on the structure. This process included:

- Analyzing load combinations as per relevant codes and standards.
- Considering factors such as load duration and impact effects.

2.2.4 Material Selection

The selection of appropriate materials has been deemed crucial for structural integrity. Factors considered in this process include:

- **Strength**: The material must withstand applied loads without failure.
- **Durability**: Resistance to environmental factors such as corrosion.
- **Cost-effectiveness**: Balancing performance with budget constraints.

2.2.5 Sectional Design

The sectional design has involved selecting a cross-section that can adequately resist bending moments and shear forces. Key considerations included:

- **Shape Efficiency**: The use of shapes like I-beams or channel sections for optimal performance.
- **Dimensions**: Ensuring adequate depth and width to resist bending.

Shear and Moment Capacity Checks

Checks for shear and moment capacity have been conducted to verify that the selected section can handle calculated shear forces and bending moments. This process involved:

- Calculating nominal strengths based on material properties.
- Ensuring that design strengths exceed applied loads.

Serviceability Criterion

Serviceability checks have been implemented to ensure that deflections and vibrations remain within acceptable limits for user comfort and functionality. Key points considered include:

- Limiting deflections to prevent structural damage or discomfort.
- Considering dynamic effects from live loads.

Detailing and Reinforcement

Proper detailing has been emphasized for construction and long-term performance. This includes:

- Specifying reinforcement in areas subjected to high stress.
- Ensuring connections are designed to transfer loads effectively.

2.2.6 Design Procedure of a Flexural Member

Designing a Laterally Unsupported Beam

The design of beams that are laterally unsupported has required special considerations due to potential lateral-torsional buckling.

Lateral Torsional Buckling

Lateral torsional buckling has been identified as a phenomenon occurring when a beam twists about its longitudinal axis under load, necessitating checks against buckling failure.

Calculations

Calculations have accounted for unbraced lengths and effective section properties to ensure stability.

2.2.7 Designing a Laterally Supported Beam

For beams that are laterally supported, the design has focused on ensuring adequate shear strength and moment capacity.

Low Shear

In low shear scenarios, typical design checks have concentrated primarily on bending strength.

High Shear

In high shear conditions, additional checks for shear capacity have been conducted to prevent failure.

Calculations

Detailed calculations have included shear force distributions and corresponding capacities.

Design Criterion for Web Buckling

Web buckling checks have been performed to ensure that thin web members do not buckle under compressive forces.

Design Criterion for Web Crippling

Web crippling checks have been established to evaluate concentrated loads; thus, specific assessments have ensured web strength is adequate.

Classification of Sections

Sections have been classified based on their ability to resist buckling and overall stability, significantly impacting design choices.

2.2.8 Solution Code

```
# -*- coding: utf-8 -*-
"""KoustavB_Osdag_FOSSEE.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1I49xdAreisXJg6v9ZpSqy5W3SFogbrya
"""

import pandas as pd

url='https://docs.google.com/spreadsheets/d/1gKWa1mIAuvzyZybtDLSTmxi0qPAFaKfIaKD8SkrTc8k/export?format=xlsx'
all_section_df=pd.read_excel(url)
all_section_df  =all_section_df.dropna()
all_section_df['Z_p'] = pd.to_numeric(all_section_df['Z_p'], errors='coerce')
def correct_format(value):
    # Split the string by space and join them back with an empty string
    parts = str(value).split()
    if len(parts) == 2:
        return f"{parts[0]}{parts[1]}"

all_section_df['s']=all_section_df['s'].apply(correct_format)
all_section_df['s'] = pd.to_numeric(all_section_df['s'], errors='coerce')

all_section_df['r_y']=pd.to_numeric(all_section_df['r_y'],errors='coerce')
```

```
all_section_df['t_f']=pd.to_numeric(all_section_df['t_f'],errors=
    'coerce')
all_section_df['d']=pd.to_numeric(all_section_df['d'],errors='
    coerce')
all_section_df['t_w']=pd.to_numeric(all_section_df['t_w'],errors=
    'coerce')
all_section_df['h']=pd.to_numeric(all_section_df['h'],errors='
    coerce')
all_section_df['Z_e']=pd.to_numeric(all_section_df['Z_e'],errors=
    'coerce')



all_section_df = all_section_df.sort_values('Z_e')



all_section_df  = all_section_df.dropna()

#Choose section type: Write the class of sections you want to
    surf through
#'ISMB', 'ISJB', 'ISHB','ISWB','ISLB'

filter_section='ISWB'
all_section_df = all_section_df[all_section_df['Designation'].str
    .contains(filter_section, case=False, na=False)]

#all_section_df

#Additional Column Creation for Validation
all_section_df['b/2*t_f']=all_section_df['b']/(2*all_section_df['
    t_f'])
all_section_df['d/t_w']=(all_section_df['h']-2*(all_section_df['
    t_f']+all_section_df['r_y']))/all_section_df['t_w']
all_section_df['n2']=2.5*(all_section_df['t_f']+all_section_df['
    r_y'])
```

```
#NEEDFUL CONSTANTS DEFINITION

import numpy as np
#Defining the constants-
gamma_mo=1.1
f_y=250 #kN/mm2 (Yield Stress of Steel)
ep=np.sqrt(250/f_y)
E=210*1e3
I=3441.8*1e4

b1=75 #Support Beam Span for Crippling Calculation

#Slenderness Ratio to Design Compressive Force Table

class_c_250={
    'f_cd':[227, 226, 220, 213, 205, 195, 182, 167, 149, 132,
       115, 101, 88.3, 77.8, 68.9,
            61.4, 55.0, 49.5, 44.7, 40.7, 37.1, 34.0, 31.2, 28.8,
                26.6],
    'KL/r':[10,20,30,40,50,60,70,80,90,100,110,120,130,140,
            150,160,170,180,190,200,210,220,230,240,250]
}

class_c_250_df=pd.DataFrame(class_c_250)

#Sectional Classification Function

def section_class(checkval_flange,checkval_web):
  #Condition taken for Rolled Section - flange

  if(checkval_flange<=9.4*ep and checkval_web<=84*ep):
    section_class='plastic'
```

```
  elif(checkval_flange<=10.5*ep and checkval_web<=105*ep):
    section_class='semi-Compact'

  elif(checkval_flange<=15.7*ep and checkval_web<=126*ep):
    section_class='compact'

  else:
    section_class='composite'

  return section_class

#Design Shear and Moment Reference Value Calculation

def design_shear(h,t_w):
  A_v=h*t_w #in mm2
  V_d=A_v*f_y/(np.sqrt(3)*gamma_mo*1e3)
  return V_d


def design_moment(Z_p,beta_b):
  M_d=beta_b*Z_p*f_y*1e-3/gamma_mo
  return M_d

#INPUTS

V_u=float(input("Enter the Maximum Shear Force (in kN): ")) #wl/2
M_u=float(input("Enter the Maximum Bending Moment (in kN-m): "))
   #wl^2/8
l=float(input("Enter the Beam Effective Span (in m): "))

#DERIVED CALCULATIONS

w=V_u**2/(2*M_u*1.5) #1.5 divided to remove the factorisation of
   load
```

```
Z_p_req=M_u*1e3*gamma_mo/f_y #Required Plastic Section Modulus in
    cm3

#Final Table Search and Suitable CS Finding

for index_ismb, row in all_section_df.iterrows():

    if row['Z_p'] >= Z_p_req:
        # Retrieve Z_p and corresponding properties
        Z_p = row['Z_p']
        Grade = row['Designation']
        h=row['h']
        t_f = row['t_f']
        t_w= row['t_w']
        d_by_t_w=row['d/t_w']
        b= row['b']
        r_1=row['r_y']
        s=row['s']
        checkval_flange=row['b/2*t_f']
        checkval_web=row['d/t_w']
        n2=row['n2']
        n1=h/2
        Z_e= row['Z_e']
        d=h-2*(t_f+r_1)
        deflection=5*w*(l*1e3)**4/(384*E*I)

        V_d=design_shear(h,t_w)
        M_d=design_moment(Z_p,1)



        Z_fd=Z_p-(h*t_w)*t_w/4
        M_fd=design_moment(Z_fd,1)


        beta=(2*V_u/V_d-1)**2
```

```
    M_dv=M_d-beta*(M_d-M_fd)

    Z_e_req=Z_p_req/s #in cm3

    M_ref=1.2*Z_e*1e-3*f_y/gamma_mo #Reference Clause Pg. 53


    if(V_d>=V_u):
      #print(0.6*V_d) #For crosscheck

      #Moment Calc for Low Shear
      if(V_u<=0.6*V_d and M_d<=M_ref and M_d>0):

        print(f"This is a Low Shear Load Case; M_d={np.round(
           M_d,2)}<{np.round(M_ref,2)}")


      #Moment Calc for High Shear
      elif(V_u>0.6*V_d and M_dv<=M_ref and M_dv>0):

        print(f"This is a High Shear Load Case; M_dv={np.
           round(M_dv,2)}<{np.round(M_ref,2)}")


      else:
        continue

      #Deflection Limit Check
      # if (deflection<=l*1e3/300):
      #   print(f"The deflection of value {np.round(
           deflection,2)} mm is within range of {l*10/3} mm -
           suitable for application")
      # else:
```

```
#   print(f"The deflection {deflection} mm exceeds for
    {Grade}")
#   continue


#Web Buckling for 45 deg load dispersion criteria

A_b=(100+n1)*t_w #in mm2
KL=0.7*d #in mm
I_eff=b*t_w**2/12 #in mm3
A_eff=b*t_w #in mm2
r=np.sqrt(I_eff/A_eff) #in mm
slendrat=KL/r
f_cd_val=0

#F_cd Interpolation Condition
for index,row in class_c_250_df.iterrows():
  if(slendrat>row['KL/r']):
    if index+1<len(class_c_250_df):
      next_row=class_c_250_df.iloc[index+1]
      f_cd=row['f_cd']
      f_cd_next=next_row['f_cd']

      f_cd_val+=f_cd+(f_cd_next-f_cd)*(slendrat-row['KL
         /r'])/(next_row['KL/r']-row['KL/r'])
    break

f_bw=A_b*f_cd_val*1e-3 #in kN

if (d_by_t_w>=67*ep):
  continue
else:
  if (f_bw>=V_u):
```

```
            print(f"The design is suitable for web buckling, with shear buckling force equal to {np.round(f_bw,2)} kN > {np.round(V_u,2)} kN")

           else:
            continue



        #Web Crippling
        F_w=(b1+n2)*t_w*f_y*1e-3/gamma_mo
        if(F_w>=V_u):
          print(f"The design is suitable for web crippling, with shear crippling force equal to  {np.round(F_w,2)} kN > {np.round(V_u,2)} kN")

        else:
          continue

        section_class_result=section_class(checkval_flange, checkval_web)
        print(f"The {Grade} designated beam is suitable for the design, the section being classified as {section_class_result}.")

        if (deflection<=l*1e3/300):
         print(f"The deflection of value {np.round(deflection,2)}  mm is within range of {l*10/3} mm - suitable for application")
        else:
         print(f"The design is physically possible, but it violates limits of serviceability, hampering user ergonomics")
         break
```

```
          break

        else: #Cancellation Criteria Final
          if index_ismb+1<len(all_section_df):
           continue
          else:
           print("No suitable solution found from given
              designations.")
           break
```

2.2.9 Explanation of Code

Input Dataset

The algorithm has begun with defining an input dataset comprising material properties and loading conditions.

Input Parameters

Parameters such as beam dimensions, load types, and material strengths have been specified for calculations.

Derived Parameters

Derived parameters have included calculated values such as moment capacities and shear strengths based on inputs.

Functions

Functions have been implemented to perform specific calculations related to bending, shear, and buckling checks.

Logic Implementation

The logic implementation has outlined how inputs will be processed through various functions to achieve design outputs.

Inputs

Inputs have been collected from user-defined parameters or pre-established datasets.

V_d and M_{dv} Check

Checks for deflection (V_d) and moment (M_{dv}) have ensured serviceability criteria are met.

Web Buckling Check

A function has evaluated web buckling potential based on applied loads.

Web Crippling Check

This check has ensured that web crippling does not occur under concentrated loads by evaluating stress distributions.

Deflection Check

Deflection checks have confirmed that beam deflections remain within allowable limits under service loads.

2.2.10 Results

Finally, results have been compiled to provide a comprehensive overview of the beam's performance under specified loading conditions, ensuring all design criteria are satisfied.

Chapter 3

Internship Task 1: Development of Purlin Module in Osdag

3.1 Task 1: Problem Statement

The task involved developing an algorithm for the purlins in the Osdag software. The purlins are a crucial flexural members most widely used in case of roof trusses. These are similar to other flexural members such as simply-supported and cantilever beams but differ with respect to the fact that they have a biaxial shear force and bending moment conditions.

3.2 Task 1: Tasks Done

3.2.1 Study of Previous Implementation

The DDCL(Design Details Check List) for the already-developed flexural module was studied - to understand how the logical implementation finally looked like in the software itself.

3.2.2 Study of Purlin DDCL

The purlin module DDCL was studied and some prominent changes in the input dock of the UI were noticed due to change in the input parameters were there with respect to the earlier flexural members. This made it simpler to proceed to the flowchart designing.

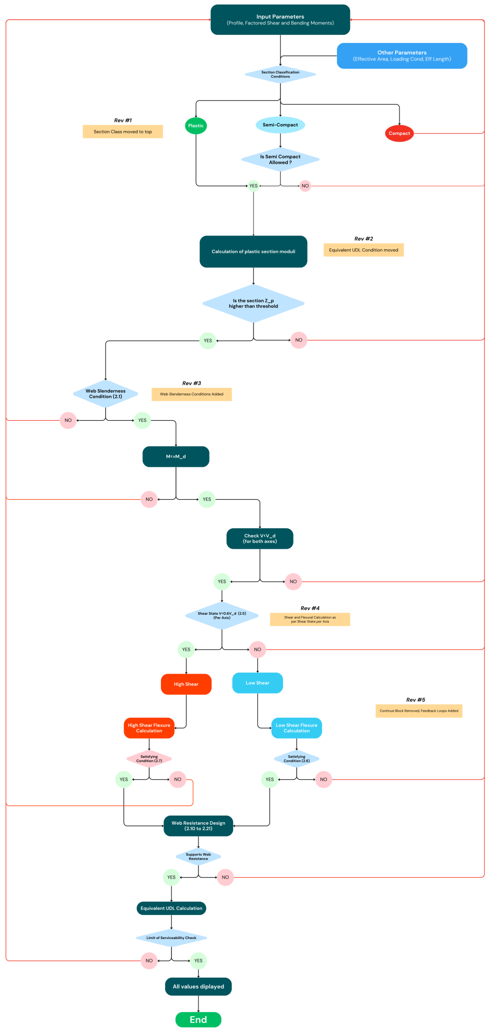

Figure 3.1: Purlin Logic Flowchart

3.2.3 Flowchart Development

The flowchart design was started with, keeping the clear, communicative visuals that helps my teammate understand the logic flow while implementing in the code. This was done using a flowchart visualization suite - which used different colours and hues for different conditions.

3.2.4 Assistance to Codal Implementation

The code implementation was in alignment to the proforma of the flexural modules developed earlier. This was evolved to newly added biaxial input parameters - and the code was developed for purlin with modified input and output docks and print statements after each step of the check.

Chapter 4

Internship Task 2: Development of Lacing Module in Osdag

4.1 Task 2: Problem Statement

The task involved the development of an algorithm for the lacing module within the Osdag software. Lacing refers to a system of diagonal or inclined members utilized to connect two or more primary components within a structural framework. This system is crucial for enhancing the stability and support of the main members, allowing them to effectively resist external loads and stresses.

4.2 Task 2: Tasks Done

4.2.1 Study of Lacing

The lacing structures were studied and the classifications of the various type of lacing and column arrangement combinations

4.2.2 Study of Lacing DDCL

The lacing module DDCL was studied and some prominent changes in the input dock of UI, with many additional logical steps were employed as compared to the earlier compression members that possessed a simpler logic. A total of 120 prepared end logical classifications were found out - based upon cladding, section type, bracing arrangement

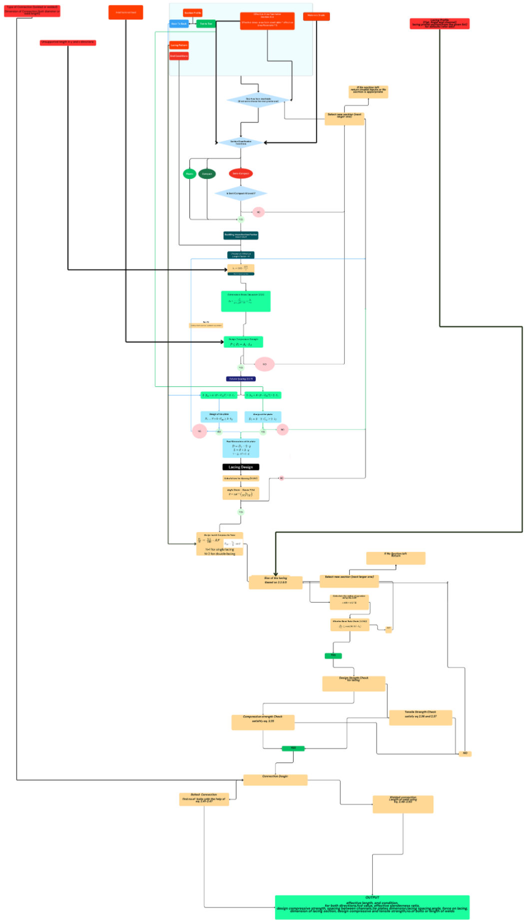

Figure 4.1: Lacing Logic Flowchart

and column arrangement. Some cases have not been considered yet, and are supposed to be taken care of later on.

4.2.3 Flowchart Development

The flowchart design was started with, keeping the clear, communicative visuals that helps my teammate understand the logic flow while implementing in the code. This was done using a flowchart visualization suite - which used different colours and hues for different conditions. The DDCL-based flowchart was put forth and further carried out by other teammates, leading to an entirety of logic.

Chapter 5

Conclusions

5.1 Tasks Accomplished

The following accomplishments have been achieved through my internship tenure.

1. The purlin module has been developed with entire UI changes and logic and merged with Osdag Github repository's developer branch.
2. Logical errors in certain feedback loops have been found with respect to the bending moment check to be catered later on.
3. The lacing module has also been fully developed and updated in the Osdag Github repository in the developer branch.

5.2 Skills Developed

This internship has been much more than few lines of my resume and added up all-round skills for me. Here I enlist the skills that I have obtained through my internship tenure.

1. **Understanding of IS 800:2007 Code**: The design of steel structures in the Indian civil landscape with adherence to the code was the key takeaway from this internship. As a mechanical engineering student with an overall idea of machine designing according to ASME codes, this was indeed a fresh knowledge.
2. **Visual Representation of Engineering Problems**: Presenting a problem in a manner that is logical, visually appealing, and apt to collaborative work was an

additional skill I took up through my internship experience.

3. **Time Management and Scheduling**: The internship was a great occasion of brushing up the time management skill and scheduling tasks of various sorts right into a tight schedule

4. **Software-level Implementation of Engineering Theory**: In addition to all the projects I have done before- working on a really useful open-source tool and seeing the connection of engineering and software development right from each lines of code.

   I am indeed grateful to my guides, mentors who hovered my mind through the experience, motivating me to contribute to open-source through my engineering knowledge. My teammates have also been supportive -steering to the end goals.

Chapter A

Appendix

A.1 Work Reports

Internship Report			
Name:		**Koustav Bhattacharjee**	
Project:		**Osdag**	
Internship:		**FOSSEE Winter Fellowship Report 2024**	
Date	**Day**	**Task**	**Work Hours**
12-Nov-2024	Tuesday	Internship Introduction + Review of IS 800:2007 Codes	4
13-Nov-2024	Wednesday	Revisit to Simply Supported Beam Design - Reconceptualisation of basic design checks	4
14-Nov-2024	Thursday	Purlin Theory Meet + Osdag Purlin and Flexural Member DDCL Study	4
15-Nov-2024	Friday	Osdag Purlin and Flexural Member DDCL Study	3
16-Nov-2024	Saturday	Conceptualisation of the Necessary Equation Flow for Purlin Design Check	3
17-Nov-2024	Sunday	Building of Purlin Flow Logic On-Paper	1
18-Nov-2024	Monday	Building Flow Logic on Flowchart Visualisation Suite	4
19-Nov-2024	Tuesday	Review Meeting + Corrections to Flow Logic	4
20-Nov-2024	Wednesday	Flow Logic Refinement + Assistance to Basic Code Development	4
21-Nov-2024	Thursday	Review Meeting + Doubt Clearance regarding new parts of Physics-Code Corelation	4
22-Nov-2024	Friday	Self-Study of the code and database structure of the previously designed modules	4
23-Nov-2024	Saturday	Flow Logic Correction	4
24-Nov-2024	Sunday	Flow Logic Correction	1
25-Nov-2024	Monday	UI Updates as per Input and Output Requirements for Purlin Module with teammate	4
26-Nov-2024	Tuesday	Review Meeting for Code Development	4
27-Nov-2024	Wednesday	Study of Lacing Module DDCL and Engineering Understanding	4
28-Nov-2024	Thursday	Lacing Module Logic Flow Study + Purlin Code Logical Assist	3
29-Nov-2024	Friday	Lacing Module Logic Flow Study + Purlin Code Logical Assist	3
30-Nov-2024	Saturday	Lacing Module Logic Flow Study + Purlin Code Logical Assist	3
1-Dec-2024	Sunday	Lacing Module Logic Flow Study + Purlin Code Logical Assist	2
2-Dec-2024	Monday	Exam Break	0
3-Dec-2024	Tuesday	Exam Break	0
4-Dec-2024	Wednesday	Exam Break	0
5-Dec-2024	Thursday	Exam Break	0
6-Dec-2024	Friday	Exam Break	0
7-Dec-2024	Saturday	Exam Break	0
8-Dec-2024	Sunday	Exam Break	0
9-Dec-2024	Monday	Exam Break	0
10-Dec-2024	Tuesday	Exam Break	0
11-Dec-2024	Wednesday	Exam Break	0
12-Dec-2024	Thursday	Exam Break	0
13-Dec-2024	Friday	Exam Break	0
14-Dec-2024	Saturday	Exam Break	0
15-Dec-2024	Sunday	Exam Break	0
16-Dec-2024	Monday	Exam Break	0
17-Dec-2024	Tuesday	Exam Break	0
18-Dec-2024	Wednesday	Exam Break	0
19-Dec-2024	Thursday	Exam Break	0
20-Dec-2024	Friday	Review of Lacing Module Coompletion by teammate and Purlin code	4
21-Dec-2024	Saturday	Purlin Code Fine Tuning + Logical Debugging	4
22-Dec-2024	Sunday	Purlin Code Fine Tuning + Logical Debugging	1
23-Dec-2024	Monday	Purlin Code Fine Tuning + Logical Debugging	4
24-Dec-2024	Tuesday	Purlin Code Fine Tuning + Logical Debugging	4
25-Dec-2024	Wednesday	Purlin Code Fine Tuning + Logical Debugging	4
26-Dec-2024	Thursday	Internship Report Meeting + Purlin Final Development	4
27-Dec-2024	Friday	Final Code Logical Review	3
28-Dec-2024	Saturday	Code Deployment with Full Logical Checks	3
		Total Work Hours	**98**

Bibliography

[1] Siddhartha Ghosh, Danish Ansari, Ajmal Babu Mahasrankintakam, Dharma Teja Nuli, Reshma Konjari, M. Swathi, and Subhrajit Dutta. Osdag: A Software for Structural Steel Design Using IS 800:2007. In Sondipon Adhikari, Anjan Dutta, and Satyabrata Choudhury, editors, *Advances in Structural Technologies*, volume 81 of *Lecture Notes in Civil Engineering*, pages 219–231, Singapore, 2021. Springer Singapore.

[2] FOSSEE Project. FOSSEE News - January 2018, vol 1 issue 3. Accessed: 2024-12-05.

[3] FOSSEE Project. Osdag website. Accessed: 2024-12-05.