FOSSEE Semester Internship Report

On

Development of LaTeX-Based Report Generation Module for 3psLCCA (Osdag)

Submitted by

Shubh Raj Gupta

2nd year B.Tech Student, Department of Computer Science and Technology

Rungta College Of Engineering & Technology

Bhilai

Under the Guidance of

Prof. Siddhartha Ghosh

Department of Civil Engineering

Indian Institute of Technology Bombay

Mentors:

Parth Karia

Swastik Gupta

May 28, 2026

Abstract

3psLCCA (Three-Pillar Sustainability Life Cycle Cost Analysis) is a desktop software developed under the FOSSEE initiative at IIT Bombay. It enables engineers to evaluate the total cost of a bridge over its entire service life across three sustainability pillars: Economic, Social, and Environmental. At the end of each analysis, the software automatically compiles and generates a comprehensive PDF report containing input data, cost results, charts, and supporting methodology.

This report documents the work undertaken during the semester-long internship, which focused on developing the LaTeX source code for two appendix sections of the 3psLCCA automated PDF report: **Appendix A – Assumptions** and **Appendix B – Calculation Methodology**.

Appendix B, the more technically demanding of the two, required typesetting a glossary of over 50 mathematical symbols in proper LaTeX notation, followed by all cost equations spanning three lifecycle stages—Initial, Use Stage, and End-of-Life—each covering economic, social, and environmental cost sub-components. This included ten multi-row tables (Tables B-1 through B-10) containing vehicle-class-specific empirical equations for fuel consumption, vehicle speed, tyre life, engine oil consumption, grease consumption, spare part ratios, fixed costs, congestion factors, and more.

The resulting LaTeX code was integrated into the existing Python-based report generation pipeline within the `src/three_ps_lcca_gui/report/` module, where it is programmatically assembled and compiled into the final PDF output delivered to the user.

Acknowledgments

I would like to express my sincere gratitude to everyone who supported me throughout this internship.

I am thankful to **Prof. Siddhartha Ghosh**, Principal Investigator of the Osdag project, Department of Civil Engineering, IIT Bombay, for providing the opportunity to contribute to this meaningful open-source initiative in structural engineering infrastructure.

I am grateful to my mentors, **Parth Karia** and **Swastik Gupta**, for their consistent guidance, technical direction, and patience in explaining the engineering concepts underlying the 3psLCCA Life Cycle Cost Analysis framework. Their feedback was instrumental in shaping the quality and accuracy of the work.

I sincerely acknowledge **Prof. Kannan M. Moudgalya**, FOSSEE Project Investigator, Department of Chemical Engineering, IIT Bombay, and the FOSSEE management team—**Usha Viswanathan** and **Vineeta Parmar**—for administering the internship program and creating a productive open-source learning environment.

I acknowledge the support from the **National Mission on Education through Information and Communication Technology (ICT)**, Ministry of Education (MoE), Government of India, for enabling this project through FOSSEE.

Finally, I thank my college faculty and family for their encouragement throughout the duration of this internship.

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.

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 [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.

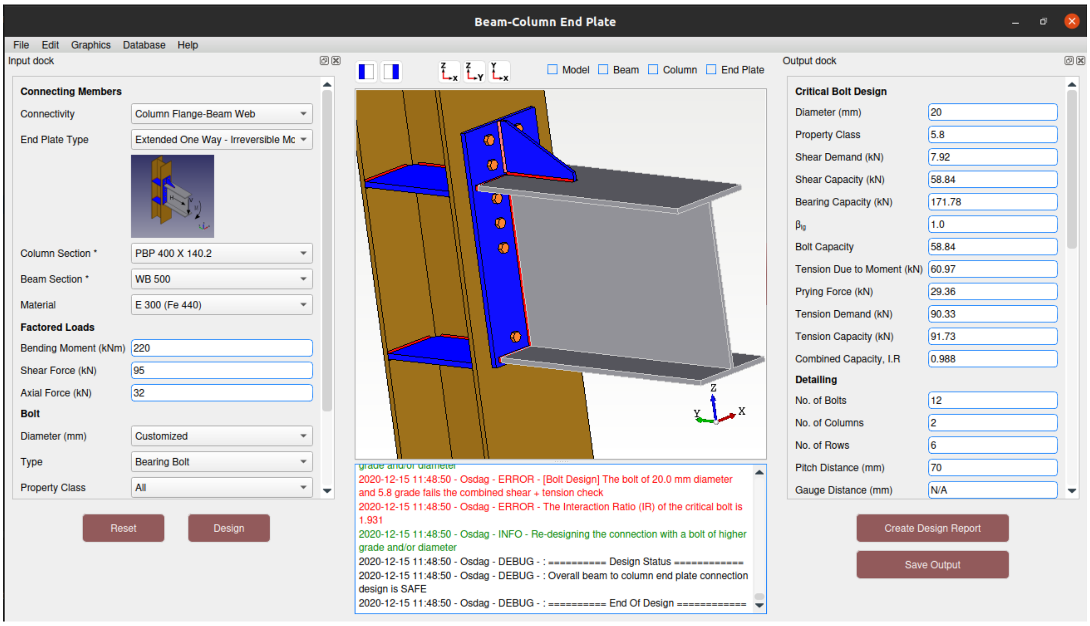

Figure 1.2: Osdag GUI

1.4 3psLCCA Software

3psLCCA (Three-Pillar Sustainability Life Cycle Cost Analysis) is a desktop software developed under the Osdag/FOSSEE umbrella at IIT Bombay, available at osdag.iitb.ac.in/3pslcca. While Osdag focuses on structural steel design, 3psLCCA addresses a complementary problem: evaluating the total cost of a bridge infrastructure project across its entire service life.

1.4.1 Purpose and Scope

Bridge infrastructure involves costs that extend far beyond initial construction. Maintenance, periodic inspections, traffic disruption during repairs, and eventual demolition all contribute to the true cost of a bridge. 3psLCCA quantifies all of these under a unified Life Cycle Cost Analysis (LCCA) framework.

The software organizes all costs under three sustainability pillars:

- **Economic Pillar**: Initial construction costs, inspection costs, maintenance costs, bearing/expansion joint replacement costs, demolition and recycling costs.
- **Social Pillar**: Road user costs during all disruption periods, including vehicle operating costs (VOC), value of time costs (VOT), and accident costs incurred

during construction, repairs, replacement, and demolition.

- **Environmental Pillar**: Carbon emission costs arising from construction materials, on-site equipment, transportation of materials, and vehicle rerouting across the full lifecycle.

1.4.2 Lifecycle Stages

The analysis covers three well-defined stages:

1. **Initial Stage**: Construction, time cost of capital, road user disruption during construction.

2. **Use and Reconstruction Stage**: Routine inspections, periodic maintenance, major inspections, major repairs, bearing/expansion joint replacement, and an optional full reconstruction cycle.

3. **End-of-Life Stage**: Final demolition and disposal, recycling cost recovery, and road user costs during demolition.

All future costs are discounted to present value using a real discount rate derived from the user-specified inflation, interest, and discount rates.

1.4.3 PDF Report Generation

After the user completes an analysis, 3psLCCA automatically generates a PDF report. This report is assembled programmatically by a dedicated Python module located at:

`src/three_ps_lcca_gui/report/`

The module constructs a LaTeX document by assembling content from section-specific Python files, which inject user data, computed results, figures, and static LaTeX content into a template. The assembled `.tex` source is then compiled to produce the final PDF output. The report covers:

- Project information and bridge description

- All input data tables (materials, traffic, financial, environmental)

- Life cycle cost results in tabular and chart form
- Summary and conclusions
- Appendix A: Assumptions
- Appendix B: Calculation Methodology

The work undertaken in this internship focused on Appendix A and Appendix B of this report.

Chapter 2

Screening Task

2.1 Problem Statement

The screening task was titled **"PyLaTeX Report Generation"** and was provided as a prerequisite for the semester-long internship with the 3psLCCA team at FOSSEE, IIT Bombay.

The objective was to write a **Python script** that:

1. Reads a **force table from a given Excel file** using Python.
2. Generates a **PDF engineering report** programmatically using the `PyLaTeX` library.
3. The report must contain:
   - An **introduction section** with an embedded beam image.
   - The **force table** from the Excel file, recreated as a properly formatted LaTeX table inside the PDF.

This task was designed to assess two things simultaneously: proficiency with Python for data reading and processing, and the ability to produce structured technical documents programmatically using LaTeX tooling—both of which are directly relevant to the 3psLCCA report generation pipeline.

2.2 Approach and Implementation

2.2.1 Reading the Excel Force Table

The force table was read from the provided `.xlsx` file using the `pandas` library. The relevant columns (node labels, force components, and moment components) were extracted into a DataFrame, which was then iterated to populate the LaTeX table rows dynamically.

Listing 2.1: Reading force data from Excel using pandas

```
import pandas as pd

df = pd.read_excel("force_table.xlsx")
# Extract relevant columns
nodes   = df["Node"].tolist()
fx_vals = df["Fx (kN)"].tolist()
fy_vals = df["Fy (kN)"].tolist()
mz_vals = df["Mz (kNm)"].tolist()
```

2.2.2 Generating the PDF Report with PyLaTeX

The `PyLaTeX` library provides Python classes that map directly to LaTeX constructs — `Document`, `Section`, `Subsection`, `Tabular`, `Figure`, and so on — allowing a PDF report to be assembled and compiled entirely from Python without manually writing `.tex` files.

The report was structured as follows:

1. **Document setup**: Page geometry, title, and required packages were configured on the `Document` object.

2. **Introduction section**: A `Section` with a brief description of the beam problem, followed by a `Figure` block embedding the provided beam image.

3. **Force table section**: A `Tabular` environment was built dynamically by iterating over the DataFrame rows, adding one row per node with proper horizontal lines.

4. **Compilation**: The `doc.generate_pdf()` method was called to trigger `pdflatex` and produce the final output PDF.

Listing 2.2: Core PyLaTeX report structure

```python
from pylatex import Document, Section, Subsection, Figure
from pylatex import Tabular, NoEscape, Command
from pylatex.utils import bold
import pandas as pd

doc = Document(geometry_options={"margin": "2.5cm"})

# Title
doc.preamble.append(Command("title", "Beam Force Analysis Report"))
doc.preamble.append(Command("date", NoEscape(r"\today")))
doc.append(NoEscape(r"\maketitle"))

# Introduction with beam image
with doc.create(Section("Introduction")):
    doc.append("This report summarizes the nodal forces acting on "
               "the simply supported beam under the applied loading.")
    with doc.create(Figure(position="h!")) as fig:
        fig.add_image("beam_image.png", width=NoEscape(r"0.8\linewidth"
            ))
        fig.add_caption("Simply supported beam with applied loading")

# Force table
with doc.create(Section("Nodal Force Table")):
    with doc.create(Tabular("|c|c|c|c|")) as table:
        table.add_hline()
        table.add_row([bold("Node"), bold("Fx (kN)"),
                       bold("Fy (kN)"), bold("Mz (kNm)")])
        table.add_hline()
        for i in range(len(nodes)):
            table.add_row([nodes[i], fx_vals[i],
                           fy_vals[i], mz_vals[i]])
            table.add_hline()

doc.generate_pdf("beam_force_report", clean_tex=False)
```

2.2.3 Output

The script successfully produced a formatted PDF report containing:

- A title page with the report name and date.
- An introduction section with the embedded beam diagram.
- A complete, properly formatted force table populated from the Excel data, with bold headers and horizontal dividers between all rows.

A video demonstration of the solution was submitted as part of the screening, available at: youtube.com/watch?v=GYbrdr9yxbQ.

2.3 Relevance to Internship Work

This task was a direct precursor to the internship work. The 3psLCCA report generation module uses a conceptually identical pattern: Python code that constructs and assembles LaTeX content and triggers compilation to produce the final PDF. The screening task built familiarity with the PyLaTeX API, the `pdflatex` compilation workflow, and the practice of embedding structured data (tables) and images into programmatically generated documents — all of which were applied directly during the internship.

Chapter 3

Internship Task: LaTeX Report Generation Module for 3psLCCA

3.1 Problem Statement

The 3psLCCA software produces an automated PDF report at the end of every analysis. This report is the primary deliverable to the engineer or agency using the software—it documents all input data, computed results, and the underlying methodology in a professionally formatted document.

The Appendix sections of this report serve a critical function: Appendix A communicates all modelling assumptions to the reader, and Appendix B provides the complete calculation methodology so that any computed result can be traced back to its underlying equation. These sections make the report suitable for submission to regulatory bodies, peer review, and institutional records.

At the time of this internship, these two appendices needed to be developed as LaTeX source code that could be integrated into the existing Python-based report pipeline. The task was:

1. Develop **Appendix A – Assumptions**: a clean, itemized list of all modelling assumptions underlying the LCCA framework.

2. Develop **Appendix B – Calculation Methodology**: a complete, accurately typeset reference section covering a 50+ symbol glossary and all cost equations across the three lifecycle stages (Initial, Use Stage, End-of-Life), with accompanying

vehicle-class-specific empirical equation tables.

3. Ensure the LaTeX code renders correctly when assembled and compiled by the existing Python pipeline.

3.2 Understanding the Report Generation Pipeline

Before writing any LaTeX code, it was necessary to understand how the report generation module works. The relevant portion of the project structure is:

```
src/three_ps_lcca_gui/report/
├─ base_report.py
├─ lcca_generate.py
├─ lcca_template.py
├─ plot_exporter.py
├─ utils.py
├─ constants.py
└─ sections/
   ├─ title_page.py
   ├─ introduction.py
   ├─ input_data.py
   ├─ results.py
   ├─ appendix.py
   ├─ appendix_A_content.py
   └─ appendix_B_content.py
```

The key files relevant to this work are:

- `lcca_generate.py`: The main entry point that orchestrates the full report. It calls individual section modules, collects their LaTeX string output, assembles them into a complete `.tex` document, and triggers compilation.

- `appendix.py`: Defines the top-level appendix section structure and calls the content modules.

- `appendix_A_content.py`: Contains the Python function that returns the LaTeX string for Appendix A.
- `appendix_B_content.py`: Contains the Python function that returns the LaTeX string for Appendix B.
- `lcca_template.py`: Holds the document-level LaTeX preamble and structural template into which all section content is inserted.

The pattern is consistent throughout the module: each section is a Python function returning a raw LaTeX string. These strings are concatenated in order and compiled via a system call to `pdflatex`. This architecture means the LaTeX code for each appendix must be self-contained—it cannot rely on packages not declared in the preamble, and wide equations must handle their own overflow management.

3.3 Tasks Done

3.3.1 Appendix A – Assumptions

Appendix A presents all modelling assumptions adopted in the LCCA framework. These assumptions cover six broad areas: the analysis period and scope, financial parameters, traffic and road user behaviour, emission factor conventions, maintenance scheduling, and end-of-life treatment of materials.

The LaTeX coding of this section was straightforward in structure but required careful attention to detail: every assumption had to be worded precisely to match the technical intent of the model. The section uses a standard `itemize` environment with `\textbf{}` for key terms.

A representative excerpt is shown below:

Listing 3.1: Appendix A – Sample assumptions in LaTeX

```
\begin{itemize}
\item The analysis period corresponds to the \textbf{design/service
   life
      of the bridge} as specified in the project data.
\item All costs are expressed in \textbf{present value terms}, using
   the
```

```
      discount rate specified in the financial input data.
\item Inflation and interest rates are assumed constant over the
      analysis period.
\item Only \textbf{GHG emissions} are considered for environmental
      related cost.
\item Emission factors remain constant over the analysis period.
\end{itemize}
```

The complete Appendix A covers 22 itemized assumptions across all modelling domains.

3.3.2 Appendix B – Glossary of Symbols

Appendix B begins with a comprehensive glossary listing every variable used in the cost equations. The glossary contains over 50 entries, each mapping a LaTeX math symbol to its full English description.

The symbols fall into several categories:

- **Traffic and accident variables**: A_{Di}, A_{Ne}, A_{Tn}, V_{CD}, CR, WZM
- **Cost variables**: C_{Di}, C_{Ti}, E_{VD}, E_{Vi}, VOC, VOT, RUC
- **Emission and environmental variables**: EF_m, EF_v, EF_{ma}, SCC, IEC, VEC
- **Lifecycle cost variables**: PWF, r, f, x, n, IC, TC, RCN
- **Vehicle performance variables**: V, RG, RF, RS, FL, W, EOL, G

Each entry is typeset as a math-mode symbol paired with a plain-text definition, using the `itemize` environment. Consistent notation was maintained across the glossary and all equations to ensure the appendix functions as a coherent reference.

Listing 3.2: Sample glossary entries in LaTeX

```
\item $A_{Di}$ = Accident distribution
\item $CF_D$ = Distance related congestion factor
\item $CF_T$ = Time related congestion factor
\item $\mathrm{PWF}$ = Present Worth Factor
\item $\mathrm{SCC}$ = Social cost of carbon
\item $EF_{ma}$ = Emission factor for machinery energy use
```

3.3.3 Appendix B – Cost Equations

The equations section of Appendix B is the most technically involved part of the work. It covers all cost equations across the three lifecycle stages, each subdivided into economic, social, and environmental sub-sections.

Stage B.1 – Initial Cost

The initial stage covers construction cost, time cost of capital, road user cost during construction, and environmental emission costs.

The vehicle operating cost during construction (VOC_{cn}) is a double summation over all vehicle classes, involving time-related costs (C_{Ti}), distance-related costs (C_{Di}), congestion factors (CF_T, CF_D), and WPI escalation indices. This equation is notably wide and required `\resizebox{\linewidth}{!}{}` to prevent overflow in the compiled PDF:

Listing 3.3: Wide equation handling with resizebox

```
\[
\resizebox{\linewidth}{!}{$\displaystyle
VOC_{cn} = D_{wm} \times DC_m \times RD \times
\left[
  \sum_{j=1}^{o} \left\{ (C_{Ti})_{j} \times (V_{CD})_{j}
      \times (CF_{T})_{j} \times (WPI_{Ti})_{j} \right\}
  + \sum_{k=1}^{p} \left\{ (C_{Di})_{k} \times (V_{CD})_{k}
      \times (CF_{D})_{k} \times (WPI_{Di})_{k} \right\}
\right]
$}
\]
```

The environmental costs in this stage—embodied carbon from construction materials, vehicular emission due to rerouting, and carbon from on-site machinery—use the Social Cost of Carbon (SCC) as a monetisation factor:

Listing 3.4: Environmental cost equation

```
\[ IEC = SCC \times \sum_{j=1}^{o}
   \left[ Q_j \times (COF)_j \times (EF_m)_j \right] \]

\[ IAEC = SCC \times \sum_{j=1}^{o}
   \left[ (ECR)_j \times (H)_j \times (D_{ma})_j
```

```
    \times (EF_{ma})_j \right] \]
```

Stage B.2 – Use Stage Cost

The use stage covers all recurring costs across the bridge's service life. These are discounted to present value using the Present Worth Factor (PWF):

$$PWF = \sum_{i=1}^{\lfloor n/x \rfloor - 1} \frac{(1+f)^{ix}}{(1+r)^{ix}}$$

where n is the design life, x is the recurrence interval (in years), f is the inflation rate, and r is the discount rate.

The economic costs in this stage include routine inspection, periodic maintenance, major inspection, major repair, and bearing/expansion joint replacement—each computed as a PWF-scaled percentage of the initial construction or superstructure cost. The social costs (road user costs during each disruption) and environmental costs (emission costs from maintenance activities and vehicle rerouting) follow the same PWF-discounting structure.

Stage B.3 – End-of-Life Cost

The end-of-life stage covers the optional reconstruction cycle (demolition of the existing bridge + new construction) and the final demolition, recycling cost recovery, and road user costs during demolition.

Recycling is treated as a cost recovery and enters as a negative cost:

$$RE_c = PWF \times \sum_{i=0}^{m} (RS)_i \times (Q_{rm})_i$$

The time cost for reconstruction uses an absolute value to ensure it is non-negative regardless of the sign of the underlying PWF:

$$TCR = \left| PWF \times TC \right|$$

3.3.4 Appendix B – Vehicle-Class Empirical Equation Tables

A distinctive feature of Appendix B is a series of ten tables (B-1 through B-10) that present empirical regression equations for vehicle-class-specific performance parameters. These equations are the basis for computing Vehicle Operating Costs across all lifecycle disruptions.

The vehicle classes covered are: Small Car, Big Car, Two-wheeler, Buses (Ordinary and Deluxe), LCV (Light Commercial Vehicle), HCV (Heavy Commercial Vehicle, 2/3 axle), and MCV (Multi-Axle Commercial Vehicle).

The ten tables cover:

Table	**Content**
B-1	Fuel consumption equations (litres/100 km)
B-2	Vehicle speed equations (km/h) as a function of roughness and rise/fall
B-3	Spare part to new vehicle cost ratio
B-4	Tyre life equations (km)
B-5	Engine oil consumption equations
B-6	Other oil consumption equations
B-7	Grease consumption equations
B-8	Utilization per day equations
B-9	Time-related congestion factor equations
B-10	Distance-related congestion factor equations

Table 3.1: Summary of vehicle-class equation tables in Appendix B

Each table is typeset using a two-column format with a narrow vehicle-class column (`p{3cm}`) and a wide equation column (`p{13.5cm}`). Mathematical expressions inside cells use the `\makebox[\linewidth][c]{}` pattern to center equations within the constrained column width, and custom `arraystretch` values are used to provide adequate vertical spacing around equations with fractions or superscripts.

A representative table definition is shown below:

Listing 3.5: Table B-2: Vehicle speed equations (excerpt)

```
\begin{table}[H]
\centering
\caption*{\textit{Table B-2 Speed equations for different vehicles}}
{\fontsize{9pt}{11pt}\selectfont
\setlength{\tabcolsep}{6pt}
```

```
\renewcommand{\arraystretch}{1.9}
\begin{tabular}{|p{3cm}|p{13.5cm}|}
\hline
Small Car & \makebox[13.5cm][c]{$V = 81.19
    - (0.7892 \times RF) - [0.001891 \times (RG - 2000)]$} \\ \hline
Big Car   & \makebox[13.5cm][c]{$V = 81.92
    - (0.7963 \times RF) - [0.001915 \times (RG - 2000)]$} \\ \hline
Two-wheeler & \makebox[13.5cm][c]{$V = 59.71
    - (0.7892 \times RF) - [0.001891 \times (RG - 2000)]$} \\ \hline
...
\end{tabular}
}
\end{table}
```

The use of `\fontsize{9pt}{11pt}\selectfont` and `\everymath` settings ensures that all inline math within the table is automatically rendered at the reduced font size, preventing equation overflow without manually adjusting every individual formula.

3.4 Integration with the Python Pipeline

The LaTeX code developed was not a standalone document. It was designed to be embedded as a string returned by Python functions inside `appendix_A_content.py` and `appendix_B_content.py`. The calling convention follows the pattern used by all other section modules in the report:

Listing 3.6: Pattern for LaTeX content in the Python pipeline

```
def get_appendix_a_content():
    """Returns the LaTeX string for Appendix A (Assumptions)."""
    content = r"""
\section*{\fontsize{14pt}{16pt}\selectfont\bfseries
          Appendix A: Assumptions}
\noindent
This Life Cycle Cost Assessment (LCCA) has been carried out ...
\begin{itemize}
\item The analysis period corresponds to the ...
...
\end{itemize}
"""
```

```python
    return content
```

The use of Python raw strings (`r""" ... """`) ensures that backslashes in the LaTeX code are passed through literally without Python's string escape processing. All section strings are concatenated by `lcca_generate.py` and written to a temporary `.tex` file, which is then compiled to PDF and returned to the user.

3.5 Challenges and Solutions

- **Wide equations**: Several road user cost equations span two summation terms and are wider than the page's text column. These were handled with `\resizebox{\linewidth}{!}{...` applied inside a displayed equation environment, which scales the equation proportionally to fit the column.

- **Math in table cells**: Standard LaTeX equation environments cannot be used inside table cells. The pattern adopted was to use `\makebox[\linewidth][c]{$...$}` with inline math, combined with manual font size reduction via `\fontsize` to ensure expressions with fractions and exponents do not overflow the cell boundary.

- **Consistent notation**: With over 50 variables in the glossary and equations spanning three stages, maintaining consistent symbol notation throughout the document required careful cross-referencing against the 3psLCCA methodology documentation to ensure the compiled appendix accurately represented the implemented model.

- **Preamble constraints**: Since the LaTeX content is injected as a string into a shared document template, no additional package declarations could be made in the appendix content itself. All required packages (`amsmath`, `geometry`, `resizebox`, `float`, etc.) had to be confirmed as already declared in `lcca_template.py`.

Chapter 4

Conclusions

4.1 Tasks Accomplished

The following tasks were completed over the course of this semester-long internship:

- Studied the 3psLCCA Life Cycle Cost Analysis framework in detail, including the three-stage (Initial, Use, End-of-Life) and three-pillar (Economic, Social, Environmental) cost structure.

- Understood the Python-based automated report generation pipeline within the `src/three_ps_lcca_gui/report/` module and the conventions followed by existing section modules.

- Developed the complete LaTeX source for **Appendix A – Assumptions**, covering 22 modelling assumptions across financial, traffic, emission, maintenance, and end-of-life domains.

- Developed the complete LaTeX source for **Appendix B – Calculation Methodology**, comprising:
  - A glossary of over 50 mathematical symbols with proper LaTeX math notation.
  - All cost equations across Stages B.1, B.2, and B.3, covering economic, social, and environmental sub-components.
  - Ten vehicle-class equation tables (B-1 through B-10) with empirical regression equations for fuel consumption, vehicle speed, tyre life, engine oil consumption,

other oil consumption, grease consumption, fixed cost, utilization per day, time-related congestion factors, and distance-related congestion factors.

- Integrated the developed LaTeX code into the Python pipeline, ensuring correct rendering in the compiled PDF output.
- Addressed technical challenges including wide-equation overflow management, math typesetting within constrained table cells, and preamble-level package compatibility.

4.2 Skills Developed

- **Advanced LaTeX typesetting**: Equation environments, multi-column tables with math content, font size control, overflow management with `\resizebox`, and `amsmath` constructs such as `align*` and `displaystyle`.
- **Open-source software contribution**: Working within an established codebase, following existing module conventions, and writing code that is compatible with the rest of the system without modifying shared infrastructure.
- **Engineering domain knowledge**: Understanding the LCCA framework for bridge infrastructure, the three sustainability pillars, present worth factor discounting, vehicle operating cost calculations, and carbon emission monetisation.
- **Python integration**: Embedding static LaTeX content as Python raw strings within a programmatic document assembly pipeline.
- **Technical documentation reading**: Interpreting engineering methodology documents and accurately translating mathematical content into typeset notation.

Chapter A

Appendix

A.1 Work Reports

FOSSEE Semester Internship 2026

Work Report

Intern Name:	Shubh Raj Gupta
Institute:	Rungta College of Engineering & Technology, Bhilai
Department:	Computer Science and Technology
Year of Study:	2nd Year B.Tech
Project Title:	Development of LaTeX-Based Report Generation Module for 3psLCCA (Osdag)
Project Guide:	Prof. Siddhartha Ghosh, Dept. of Civil Engineering, IIT Bombay
Mentors:	Parth Karia, Swastik Gupta
Internship Start:	6 January 2026
Internship End:	9 May 2026

1 Internship Overview

This document is the weekly work report for the FOSSEE Semester Internship (January – May 2026). The project involved developing a LaTeX-based automated report generation module for the **3psLCCA** (Three-Pillar Sustainability Life Cycle Cost Analysis) software, part of the **Osdag** open-source structural engineering toolkit developed at IIT Bombay.

The internship comprised a **pre-internship screening task** (PyLaTeX report generation) followed by the **core internship task**: creating `appendix_A_content.py` and `appendix_B_content.py` — two Python modules that programmatically emit LaTeX source code for the Assumptions and Calculation Methodology appendices of the automated LCCA report.

2 Internship Calendar

Period	**Summary**
Jan 6 – Jan 17	Orientation, codebase study, LCCA framework study
Jan 20 – Jan 31	Screening task: PyLaTeX report generation
Jan 26	*Holiday — Republic Day*
Feb 3 – Feb 14	Appendix A development (22 modelling assumptions)
Feb 17 – Feb 28	Appendix B: Glossary of 50+ mathematical symbols
Mar 3 – Mar 13	Appendix B: Cost equations (Stage B.1 – Initial Cost)
Mar 6	*Holiday — Holi*
Mar 16 – Mar 27	Appendix B: Cost equations (Stage B.2 – Use Stage Cost)
Mar 30 – Apr 10	Appendix B: Cost equations (Stage B.3 – End-of-Life Cost)
Apr 14	*Holiday — Dr. Ambedkar Jayanti*
Apr 3	*Holiday — Good Friday*
Apr 13 – Apr 30	Appendix B: Vehicle-class tables B-1 to B-10
May 1 – May 8	Integration, testing, report writing, and submission

3 Weekly Work Log

Wk	Dates	Work Done	Status
Phase 0: Screening Task & Orientation			
1	06 Jan – 10 Jan 2026	Received internship offer. Studied project description for 3psLCCA. Set up development environment (Python 3.11, VS-Code, MiKTeX). Read through 3psLCCA source tree to understand the report pipeline.	Done
2	13 Jan – 17 Jan 2026	Studied the FOSSEE/Osdag codebase conventions. Read through `base_report.py`, `lcca_generate.py`, and `lcca_template.py` to understand how LaTeX content is assembled and compiled inside the Python pipeline.	Done
3	20 Jan – 24 Jan 2026	**Screening Task:** Wrote Python script using `pandas` to read force table from `force_table.xlsx`. Generated a structured PDF engineering report using the `PyLaTeX` library containing an introduction section, embedded beam image, and dynamically populated force table. Submitted video demonstration.	Done
	26 Jan 2026	*Holiday — Republic Day (Monday).*	—
Phase 1: Appendix A — Assumptions			
4	02 Feb – 06 Feb 2026	Studied 3psLCCA methodology documentation provided by mentors. Identified all modelling assumptions across the six domains: analysis period and scope, financial parameters, traffic behaviour, emission factor conventions, maintenance scheduling, and end-of-life treatment.	Done
5	09 Feb – 13 Feb 2026	Wrote complete LaTeX source for **Appendix A – Assumptions** inside `appendix_A_content.py` as a Python raw string. Coded 22 itemized assumptions using `itemize` with `\textbf{}` highlights. Verified render in compiled test PDF.	Done
6	16 Feb – 20 Feb 2026	Reviewed Appendix A content with mentors (Parth Karia, Swastik Gupta). Incorporated feedback: revised wording on three assumptions related to GHG scope and discount rate treatment. Re-tested compilation.	Done
Phase 2: Appendix B — Glossary of Symbols			
7	23 Feb – 27 Feb 2026	Extracted all variable symbols from the 3psLCCA methodology document. Organised 50+ symbols into categories: traffic/accident variables, cost variables, emission variables, lifecycle cost variables, and vehicle performance variables.	Done
8	02 Mar – 06 Mar 2026	Typeset the full 50+ symbol glossary inside `appendix_B_content.py` using LaTeX `itemize` with math-mode symbols paired with plain-text definitions. Ensured consistent notation (e.g., PWF, A_{Di}, VOC) with the equations to be written in subsequent weeks.	Done
Phase 3: Appendix B — Cost Equations			
9	09 Mar – 13 Mar 2026	Typeset **Stage B.1 – Initial Cost** equations. Handled the wide double-summation VOC_{cn} equation using `\resizebox{\linewidth}{!}{...}` to prevent page overflow. Typeset IEC and IAEC environmental cost equations using SCC monetisation.	Done
	06 Mar 2026	*Holiday — Holi (Friday).*	—
Continued on next page...			

Wk	Dates	Work Done	Status
10	16 Mar – 20 Mar 2026	Typeset **Stage B.2 – Use Stage Cost** equations. Coded the Present Worth Factor (PWF) formula and all recurring cost equations (routine inspection, periodic maintenance, major inspection, major repair, bearing/expansion joint replacement) as PWF-scaled percentages.	Done
11	23 Mar – 27 Mar 2026	Typeset social and environmental sub-components of Stage B.2: road user cost equations during maintenance disruptions and vehicle rerouting emission cost equations, all discounted by PWF.	Done
12	30 Mar – 03 Apr 2026	Typeset **Stage B.3 – End-of-Life Cost** equations: reconstruction cycle, demolition, recycling cost recovery (RE_c), and time cost for reconstruction (TCR) with absolute-value treatment. Reviewed all three equation stages end-to-end for notational consistency.	Done
13	06 Apr – 10 Apr 2026	Mentor review of complete Equations section. Resolved two notation inconsistencies flagged by Swastik Gupta. Recompiled and verified the full appendix render.	Done
Phase 4: Appendix B — Vehicle-Class Empirical Tables (B-1 to B-10)			
	14 Apr 2026	*Holiday — Dr. Ambedkar Jayanti (Tuesday).*	—
14	13 Apr – 17 Apr 2026	Designed two-column table format: narrow vehicle-class column (`p{3cm}`) + wide equation column (`p{13.5cm}`). Used `\makebox[\linewidth][c]{$...$}` to centre inline math. Typeset **Tables B-1** (Fuel consumption) and **B-2** (Vehicle speed equations).	Done
	03 Apr 2026	*Holiday — Good Friday.*	—
15	20 Apr – 24 Apr 2026	Typeset **Tables B-3** (Spare parts ratio), **B-4** (Tyre life), **B-5** (Engine oil consumption), and **B-6** (Other oil consumption). Applied `\fontsize{9pt} {11pt}\selectfont` with `arraystretch` adjustments to handle equations with fractions and exponents inside constrained table cells.	Done
16	27 Apr – 30 Apr 2026	Typeset **Tables B-7** (Grease consumption), **B-8** (Utilization per day), **B-9** (Time-related congestion factor), and **B-10** (Distance-related congestion factor). Verified all 10 tables compile without overflow or alignment errors.	Done
Phase 5: Integration, Testing, and Report Submission			
17	01 May – 05 May 2026	Integrated `appendix_A_content.py` and `appendix_B_content.py` into the live 3psLCCA report pipeline. Confirmed that `lcca_generate.py` correctly calls both functions, concatenates their output, and compiles a full PDF without errors. Verified all required packages (`amsmath`, `geometry`, `float`, etc.) were already declared in `lcca_template.py`.	Done
18	06 May – 08 May 2026	Wrote internship report (LaTeX document covering introduction, screening task, internship task, and conclusions). Final review with mentors. Submitted all deliverables to FOSSEE.	Done

4 Deliverables Submitted

1. **Screening Task Submission:** Python script for PyLaTeX-based force table PDF report generation, with video demonstration (YouTube link).
2. **`appendix_A_content.py`:** Python module returning the LaTeX string for Appendix A (22

modelling assumptions covering financial, traffic, emission, maintenance, and end-of-life domains).

3. `appendix_B_content.py`**:** Python module returning the LaTeX string for Appendix B — comprising a 50+ symbol glossary, all cost equations across Stages B.1, B.2, and B.3, and ten vehicle-class empirical equation tables (B-1 through B-10).
4. **FOSSEE Semester Internship Report:** Full LaTeX report (`fossee_report.pdf`) documenting the entire internship work, submitted via the FOSSEE portal.

5 Holidays / Non-Working Days

Date	Day	Occasion
26 January 2026	Monday	Republic Day
6 March 2026	Friday	Holi
3 April 2026	Friday	Good Friday
14 April 2026	Tuesday	Dr. B.R. Ambedkar Jayanti

Total Internship Duration: 6 January 2026 – 8 May 2026 **Effective Working Weeks:** 18 weeks (4 holidays excluded)

Shubh Raj Gupta
FOSSEE Semester Intern, 2026
Rungta College of Engineering & Technology, Bhilai

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: 2026.

[3] FOSSEE Project. Osdag website. Accessed: 2026.