FOSSEE Autumn Internship Report

On

Development of LCC module for OSDAG

Submitted by

Prerna Praveen Vidyarthi

3rd Year B.Tech Student, Department of Information technology

Banasthali vidyapith

Rajasthan

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 6, 2026

Acknowledgments

Acknowledgments

I would like to express my sincere gratitude to everyone who supported and guided me throughout my internship. This project has been a deeply enriching learning experience, and I am truly thankful for the encouragement, mentorship, and opportunities I received.

I am especially grateful to the Osdag team members – Ajmal Babu M. S., Ajinkya Dahale, and Parth Karia – for their constant guidance, technical insights, and valuable feedback during the course of my work. Their mentorship not only helped me understand the project better but also enhanced my problem-solving and programming skills.

I would also like to express my deep appreciation to Prof. Siddhartha Ghosh, Principal Investigator of Osdag, Department of Civil Engineering, IIT Bombay, for his vision and leadership that made this project possible.

My heartfelt thanks go to Prof. Kannan M. Moudgalya, Principal Investigator of the FOSSEE Project, Department of Chemical Engineering, IIT Bombay, for creating such a platform where students like me can contribute and learn through impactful open-source projects.

I am also thankful to FOSSEE Managers, Ms. Usha Viswanathan and Ms. Vineeta Parmar, and their entire team for their constant support and for ensuring a smooth and well-organized internship experience.

I gratefully acknowledge the support provided by the National Mission on Education through Information and Communication Technology (ICT), Ministry of Education (MoE), Government of India, whose initiative enabled me to work on this project.

I would also like to thank my colleagues and fellow interns at FOSSEE, with whom I had the opportunity to collaborate, exchange ideas, and grow together.

Finally, I am grateful to my college, the Department of Computer Science, and my

faculty members, for their encouragement and support during my academic journey, which helped me take full advantage of this internship.

This internship has been a significant milestone in my academic and professional development, and I feel privileged to have been a part of the FOSSEE Summer Fellowship 2025.

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.

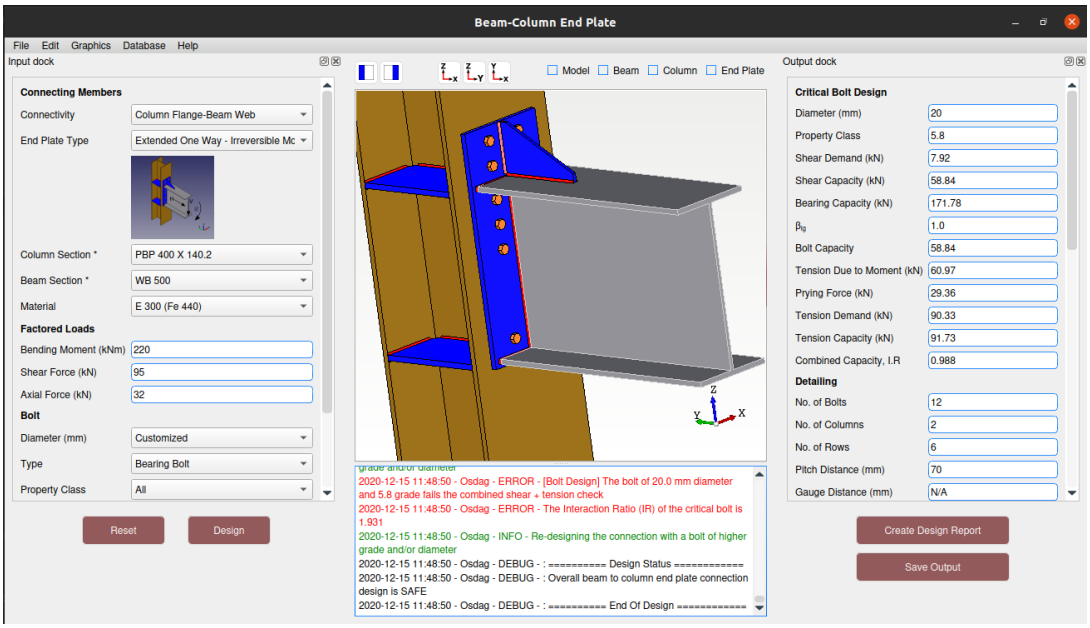

Figure 1.2: Osdag GUI

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

Internship Task 1: Architectural Research & Project Initiation

2.1 Overview

The initial phase of the internship commenced on August 8, 2025, focusing on understanding the existing codebase of the Osdag Life Cycle Costing (LCC) desktop application [**?**]. The primary objective of this phase was to prepare the system for a major upgrade from a monolithic structure to a modular architecture.

2.2 Problem Statement

The existing Osdag LCC desktop application required updates to handle increasing system complexity. Specifically, the application lacked a mechanism to dynamically load widgets and calculation modules, as all components were loaded at application startup. Additionally, existing bugs related to user interface components—particularly dropdown menu handling—required immediate resolution [**?**].

2.3 Technical Implementation

2.3.1 Modular Architecture Research

Between August 12 and August 13, 2025, extensive research was conducted on Python's internal import mechanisms to support a modular architecture [**?**].

- **Library Study:** Studied Python libraries such as `importlib` and `entry_points` to understand dynamic module loading.
- **Prototype Design:** Conceptualized a plugin-based system in which new LCC modules (for example, different bridge types) could be added as independent Python packages without requiring changes to the core application engine.

2.3.2 Initial Bug Fixes

On August 8, 2025, a critical meeting was conducted to discuss required updates to the desktop version of the application. Following this discussion, work was carried out to correct issues related to dropdown menu handling in the user interface [**?**]. These fixes involved improving event-handling logic to ensure that selecting an item from the dropdown correctly triggered the associated backend processes.

2.4 Outcome

This phase successfully established the conceptual foundation for modularizing the Osdag LCC application and improved the stability of core user interface interactions, enabling smoother future development.

Chapter 3

Internship Task 2: UI Modernization & Visualization

3.1 Overview

From August 19 to August 26, 2025, the focus shifted to enhancing the User Experience (UX) and User Interface (UI). The goal was to transform the existing single-window interface into a multi-tabbed environment capable of handling complex engineering workflows [**?**].

3.2 Implementation Details

3.2.1 Multi-Tab Development

On August 19, 2025, development began on the design and implementation of the multi-tab feature [**?**].

- **State Management:** A state management system was designed to ensure that data in one tab (e.g., a steel bridge project) did not interfere with data in another tab (e.g., a concrete road project).
- **Dynamic Instantiation:** Logic was implemented to dynamically instantiate new widget classes whenever a user requested a new tab.

3.2.2 Visualization Components

On August 25, 2025, graphical visualization components were integrated into the application [?]. This involved embedding plotting libraries, such as Matplotlib, within the GUI framework to visually render cost breakdowns.

3.2.3 Version Control Operations

On August 26, 2025, a code rebase was performed to synchronize the implementation with teammate contributions [?]. This ensured smooth integration of the multi-tab functionality with parallel developments.

3.3 Conclusion

The successful integration of the multi-tab feature significantly improved application usability by enabling engineers to compare multiple project scenarios side-by-side. latex

3.4 UI Screenshots

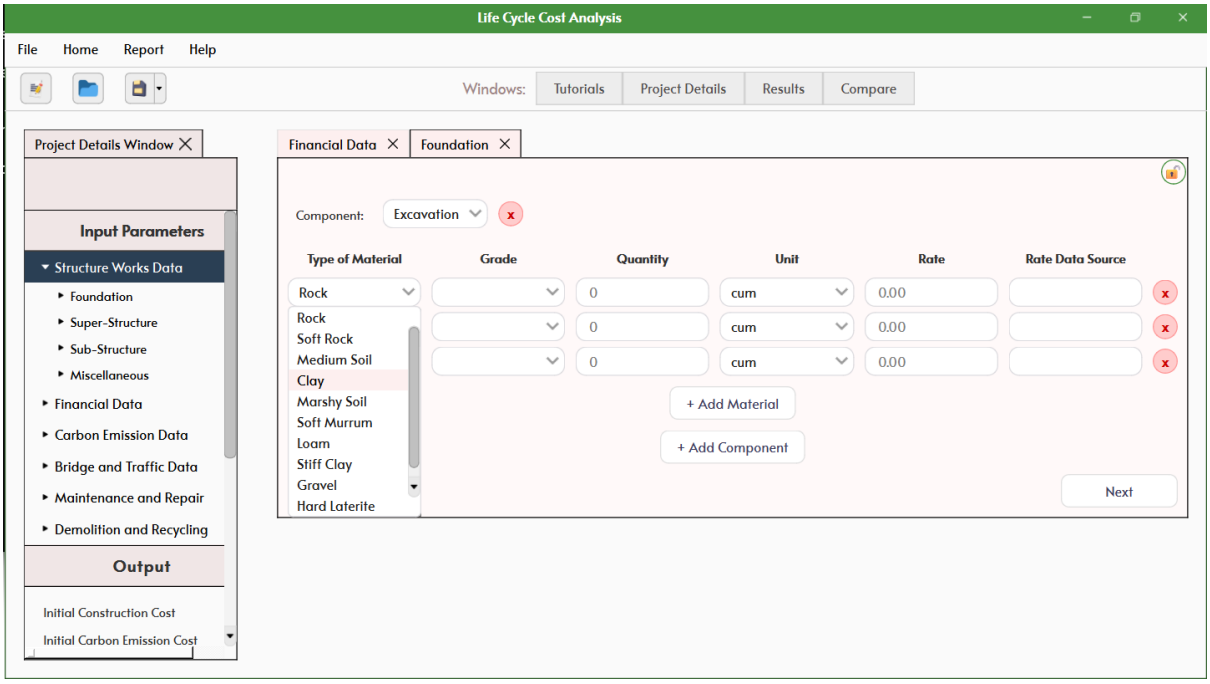

Figure 3.1: Multi-tabbed Foundation and Structural Input Interface

Chapter 4

Internship Task 3: Backend Logic & Initial Costing

4.1 Overview

The third phase, spanning September 3 to September 9, 2025, focused on the core mathematical logic of Life Cycle Costing (LCC), particularly the computation of initial costs. This phase bridged the gap between frontend widgets and backend engineering formulas.

4.2 Technical Implementation

4.2.1 Formula Implementation

On September 3, 2025, an analysis of initial cost formulas was conducted, followed by backend implementation [**?**]. Engineering equations related to material costs, labor, and machinery were translated into Python functions.

4.2.2 Widget Integration

On September 6, 2025, the backend logic was integrated with the Bridge and Traffic widgets [**?**].

- **Data Binding:** Inputs from the Bridge widget, such as span length and width, were accurately passed to the calculation engine.

- **Database Setup:** A database schema was initialized to support data generated by these widgets.
- **Financial Connectivity:** By September 9, 2025, database connectivity for the Financial Input section was established, enabling persistence of discount rates and inflation values [**?**].

4.3 Challenges

The primary challenge was maintaining unit consistency across widgets, which required strict validation of user inputs. latex Copy code

Chapter 5

Internship Task 4: Database Architecture & Data Persistence

5.1 Overview

Between September 12 and September 30, 2025, work focused on developing the data layer of the application. A robust local database was required to store user projects and retrieve standard engineering data.

5.2 Database Design

5.2.1 SQLite Integration

SQLite was selected as the storage engine. On September 12 and September 14, 2025, SQL queries were developed to store user input data securely [**?**].

- **INSERT Operations:** Parameterized queries were created to safely insert project data.
- **Schema Refinement:** Database tables were structured to support hierarchical LCC data (Project → Modules → Components).

5.2.2 Structural Data Logic

On September 18, 2025, data storage logic for structural widgets was updated to ensure proper serialization before storage [?].

5.2.3 UI Enhancements for Data

On September 16, 2025, the Financial widget UI was enhanced to reflect saved data states, providing visual confirmation to users [?].

5.3 Outcome

A fully functional persistence layer was implemented, allowing users to save and reload projects across sessions. latex

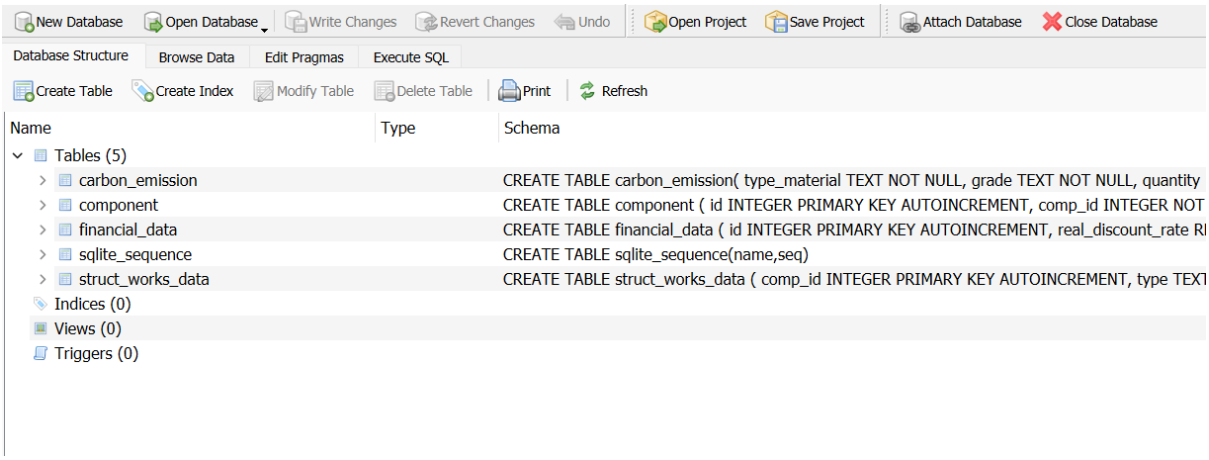

Figure 5.1: database table

Table: component

Filter in any column

	id	comp_id	type_material	grade	quantity	unit	rate	rate_data_source	created_at	updated_at
	Fil...	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	Marshy Soil		12.0	MT	34.0		2026-01-06 18:08:17	2026-01-06 18:08:17
2	2	1	Rock		5.0	cum	23.0		2026-01-06 18:08:17	2026-01-06 18:08:17
3	3	1	Loam		2.0	kg	12.0		2026-01-06 18:08:17	2026-01-06 18:08:17
4	4	2	Steel Rebar	Fe500	4.0	MT	12.0		2026-01-06 18:08:37	2026-01-06 18:08:37
5	5	2	Steel Rebar	Fe415	2.0	kg	32.0		2026-01-06 18:08:37	2026-01-06 18:08:37
6	6	3	Reinforced Cement Concrete	M30	2.0	cum	1.0		2026-01-06 18:09:07	2026-01-06 18:09:07
7	7	3	Steel Rebar	Fe415	1.0	kg	2.0		2026-01-06 18:09:07	2026-01-06 18:09:07
8	8	4	Rubber		0.0	kg	3.0		2026-01-06 18:09:27	2026-01-06 18:09:27
9	9	4	Structural Steel	E250	0.0	kg	2.0		2026-01-06 18:09:27	2026-01-06 18:09:27

Figure 5.2: database table

Database Structure | Browse Data | Edit Pragmas | Execute SQL

Table: struct_works_data

Filter in any column

	comp_id	type	component_type	created_at	updated_at
	Filter	Filter	Filter	Filter	Filter
1	1	Foundation	Excavation	2026-01-06 18:08:17	2026-01-06 18:08:17
2	2	Super-Structure	Girder	2026-01-06 18:08:37	2026-01-06 18:08:37
3	3	Sub-Structure	Pier	2026-01-06 18:09:07	2026-01-06 18:09:07
4	4	Miscellaneous	Bearings	2026-01-06 18:09:27	2026-01-06 18:09:27

Figure 5.3: database table

Table: carbon_emission

Filter in any column

	type_material	grade	quantity	unit	emission_factor	embodied	created_at	updated_at
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	Loam		2.0	kg	1.0	2.0	2026-01-06 18:10:27	2026-01-06 18:10:27
2	Marshy Soil		12.0	MT	2.0	3.0	2026-01-06 18:10:27	2026-01-06 18:10:27
3	Reinforced Cement Concrete	M30	2.0	cum	2.0	4.0	2026-01-06 18:10:27	2026-01-06 18:10:27
4	Rock		5.0	cum	1.0	2.0	2026-01-06 18:10:27	2026-01-06 18:10:27
5	Rubber		0.0	kg	3.0	2.0	2026-01-06 18:10:27	2026-01-06 18:10:27
6	Steel Rebar	Fe415	3.0	kg	2.0	4.0	2026-01-06 18:10:27	2026-01-06 18:10:27
7	Steel Rebar	Fe500	4.0	MT	1.0	2.0	2026-01-06 18:10:27	2026-01-06 18:10:27
8	Structural Steel	E250	0.0	kg	2.0	4.0	2026-01-06 18:10:27	2026-01-06 18:10:27

Figure 5.4: database table

Chapter 6

Internship Task 5: Carbon Emission Modeling

6.1 Overview

This task, conducted from October 3 to October 13, 2025, focused on environmental impact assessment through the Carbon Emission module, a key component of modern LCC analysis.

6.2 Methodology

6.2.1 Formula Correction and Logic

On October 3, 2025, initial carbon cost formulas were corrected after reviewing environmental engineering standards to ensure accurate emission coefficients for steel and concrete [**?**].

6.2.2 Emission Factors UI

On October 7, 2025, the Carbon Emission Factor UI was updated [**?**].

- **Data Fetching:** Database logic was implemented to populate standard emission factors.

- **Calculation Refinement:** On October 10, 2025, emission calculations and SQL queries were refined to improve accuracy [?].

6.2.3 Core Logic Completion

By October 13, 2025, backend debugging was completed, resulting in stable and reliable carbon emission and cost factor calculations [?]. latex

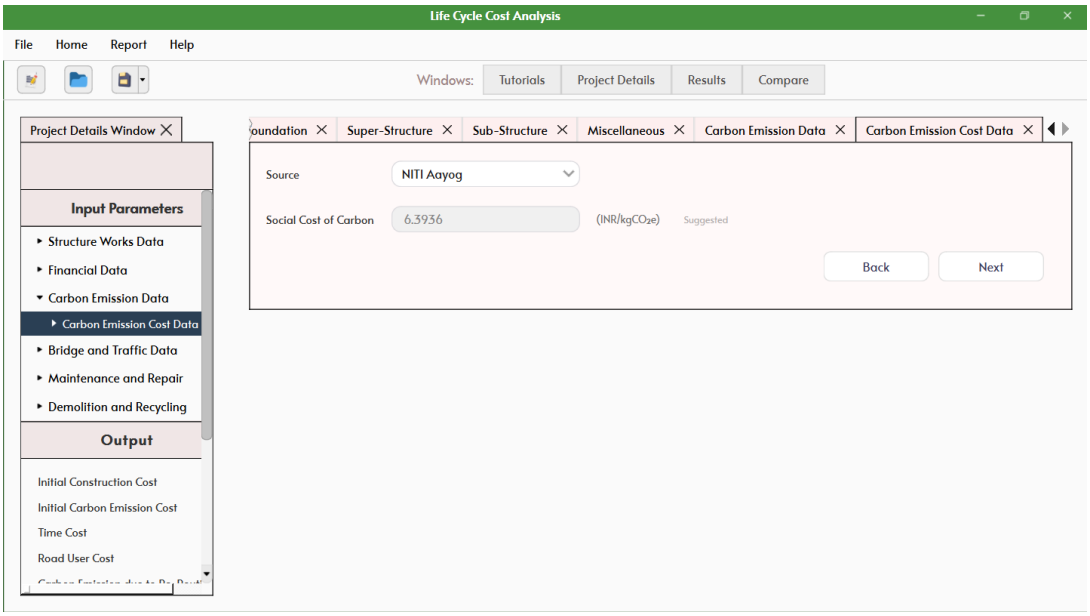

Figure 6.1: Carbon Emission Cost Data Input Module

Chapter 7

Internship Task 6: Structural Module Integration & Debugging

7.1 Overview

From October 17 to October 27, 2025, efforts focused on completing structural logic and resolving interface issues.

7.2 Implementation Details

7.2.1 Core Logic Completion

On October 17, 2025, carbon emission calculation logic was finalized [**?**]. Subsequently, on October 21, the Bridge and Traffic UI was updated to ensure full backend connectivity [**?**].

7.2.2 Optimization and Verification

On October 23, 2025, calculations were verified and SQL queries optimized through stress testing [**?**].

7.2.3 Bug Resolution

On October 27, 2025, a critical UI bug in the Foundation widget was resolved, restoring full usability [**?**].

7.3 Conclusion

This phase unified environmental and structural models into a single, consistent calculation engine. latex Copy code

Chapter 8

Internship Task 7: Advanced Economic Metrics (VOT & VOC)

8.1 Overview

From October 30 to November 13, 2025, advanced economic metrics—Value of Time (VOT) and Vehicle Operating Cost (VOC)—were implemented to calculate user costs in LCC analysis.

8.2 Theoretical Background

8.2.1 Value of Time (VOT)

On November 7, 2025, VOT calculations were implemented, including bidirectional Vehicle Operating Cost considerations [**?**].

8.2.2 Research and Development

On November 15, 2025, research was conducted on VOT factors such as vehicle types and lane configurations [**?**].

- **Scripting:** VOC calculation scripts were written on November 13, 2025 [**?**].
- **Test UI:** A dedicated Test UI was developed to validate algorithms before integration [**?**].

8.2.3 Finalization

By November 11, 2025, the Bridge and Traffic module was finalized with integrated VOT and VOC calculations [?]. latex

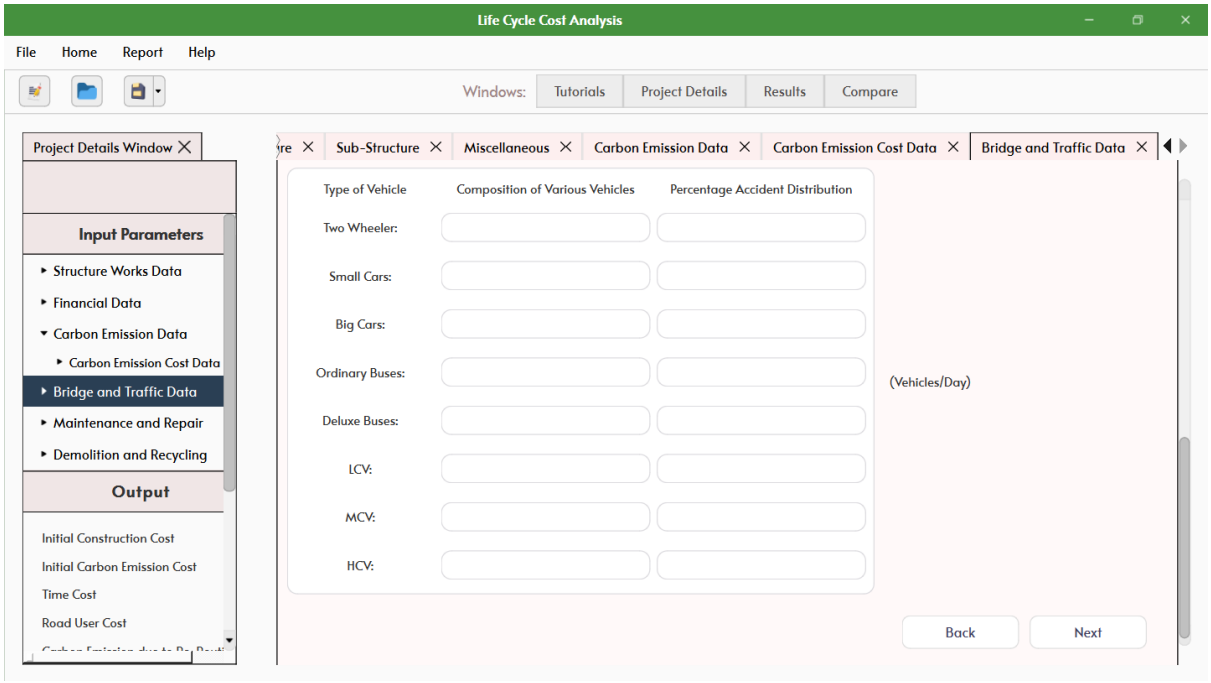

Figure 8.1: Bridge and Traffic Data Input for VOT and VOC Analysis

Chapter 9

Internship Task 8: End-of-Life Module & Final Submission

9.1 Overview

The final phase, concluding on November 30, 2025, addressed End-of-Life (Demolition and Recycling) analysis and final system validation.

9.2 Implementation Details

9.2.1 Demolition and Recycling

On October 30, 2025, backend and UI components for demolition and recycling were updated [**?**].

- **Database Connectivity:** Established on November 4, 2025 [**?**].
- **Calculation Module:** Fully developed by November 7, 2025, enabling salvage value and disposal cost estimation [**?**].

9.2.2 Final Verification

- **Comprehensive Review:** Conducted on November 17, 2025 [**?**].
- **Formula Refinement:** Completed on November 23, 2025 [**?**].

- **Mathematical Verification:** Final validation performed on November 26, 2025 [?].

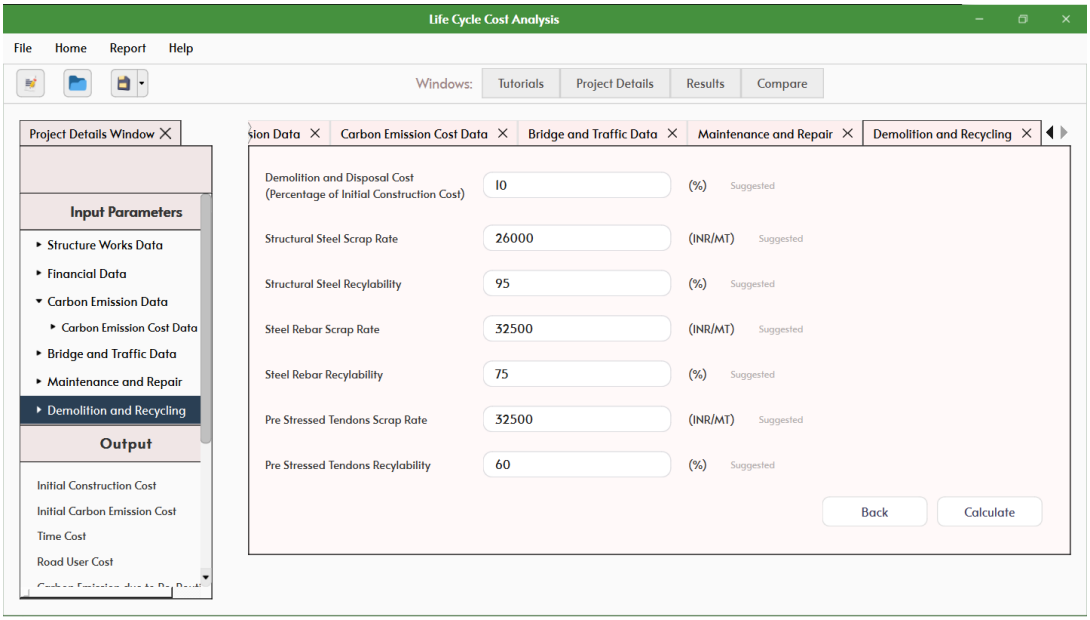

Figure 9.1: Demolition and Recycling Cost Parameters Interface

```
Results:

{'Additional Carbon Emission Cost': None,
 'Carbon Emission due to Rerouting during Demolition and Disposal': 0.0,
 'Carbon Emission due to Rerouting during Initial Construction': None,
 'Carbon Emission due to rerouting during Major Repairs': 331100.0151620212,
 'Carbon Emission due to rerouting during Replacement': 4518.63216134478,
 'Demolition and Disposal Cost': 31.897482533830697,
 'Demolition and Disposal related Carbon Emission': 58583.044396450256,
 'Major Inspection Cost': 0.7297885941136042,
 'Major Repair Cost': 86.4017567673785,
 'Major Repair Related Carbon Emisson Cost': 158685.81312854955,
 'Periodic Maintenance Carbon Emission Cost': 42557.611206592075,
 'Periodic Maintenance Cost': 23.171903647706003,
 'Reconstruction Cost': None,
 'Recycling Cost': 0.0,
 'Repair and Rehabilitation Cost': None,
 'Time Cost': 1284.5625,
 'Total Initial Carbon Emission Cost': 1217668.46,
 'Total Initial Construction Cost': 663.0,
 'Total Road User Cost': None,
 'Total Routine Inspection Cost': 230.19249120260022,
 'Total SuperStructure Cost': 112.0}
```

Figure 9.2: Computed Life Cycle Cost Analysis Results

9.3 Project Conclusion

On November 30, 2025, the project concluded with the successful submission of all calculation modules. The Osdag LCC desktop application was upgraded with modular ar-

chitecture, robust database integration, and comprehensive economic and environmental analysis capabilities [**?**].

9.4 Tasks Accomplished

During the internship period, the following key tasks were successfully completed as part of the development and enhancement of the Osdag Life Cycle Costing (LCC) desktop application:

- Studied and analyzed the existing Osdag LCC desktop application architecture to understand its limitations and prepare it for modularization.
- Researched Python-based modular and plugin architectures to enable dynamic loading of calculation modules and widgets.
- Modernized the User Interface (UI) by transforming the single-window layout into a multi-tabbed environment, allowing multiple projects to be handled simultaneously.
- Implemented robust state management to ensure data isolation between different tabs and project workflows.
- Integrated graphical visualization components to visually represent cost breakdowns and analytical results.
- Developed and implemented backend logic for Initial Cost calculations, translating engineering formulas into executable Python functions.
- Integrated frontend widgets (Bridge, Traffic, Financial inputs) with backend calculation engines through proper data binding.
- Designed and implemented a local SQLite database for persistent storage of user projects and standard engineering data.
- Created parameterized SQL queries and refined database schemas to support hierarchical Life Cycle Costing data.
- Implemented Carbon Emission modeling, including correction of emission formulas and integration of emission factor inputs through the UI.
- Developed and validated structural module logic, ensuring seamless interaction between structural parameters and environmental models.

- Implemented advanced economic metrics such as Value of Time (VOT) and Vehicle Operating Cost (VOC) for accurate user cost estimation.
- Developed testing interfaces to validate complex calculation logic before full integration into the main application.
- Implemented End-of-Life cost analysis modules, including demolition and recycling cost calculations.
- Performed extensive debugging, optimization, and verification of backend calculations and database operations.
- Conducted final system validation, ensuring consistency, accuracy, and stability across all LCC modules prior to project submission.

Chapter A

Appendix

A.1 Work Reports

INTERNSHIP WORK REPORT

Name: Prerna Praveen Vidyarthi

Project: Osdag – LCC (Life Cycle Costing)

Internship: FOSSEE Summer Fellowship 2025

Duration: 08 August 2025 to 30 November 2025

Date	Day	Task Description	Hours
08-Aug-2025	Friday	Meeting regarding Osdag LCC desktop version updates and correction of dropdown menus	4
12-Aug-2025	Tuesday	Discussion on project updates and research on opening the desktop app from any directory	5
13-Aug-2025	Wednesday	Study of Python libraries such as importlib and entry_points for modular handling	4
19-Aug-2025	Tuesday	Development and implementation of multi-tab feature	4
25-Aug-2025	Monday	Integration of graphical visualization components	2
26-Aug-2025	Tuesday	Code rebase and synchronization with teammate contributions	2
03-Sep-2025	Wednesday	Analysis of initial cost formulas and backend implementation	4
06-Sep-2025	Saturday	Integration of Bridge and Traffic widget backend and database setup	4
09-Sep-2025	Tuesday	Database connectivity for Financial Input section	4
12-Sep-2025	Friday	Development of SQL queries for storing user inputs in SQLite	4
14-Sep-2025	Sunday	Correction of Initial Carbon Cost formulas	4
16-Sep-2025	Tuesday	UI enhancements for the Financial widget	2
18-Sep-2025	Thursday	Update of data storage logic for structural widgets	2
23-Sep-2025	Tuesday	Carbon Emission Factor UI update and database data fetching	3

26-Sep-2025	Friday	Refinement of carbon emission calculations and SQL queries	4
30-Sep-2025	Tuesday	Debugging backend variables and resolving calculation inconsistencies	5
03-Oct-2025	Friday	Development of Carbon Emission and Cost factor logic	5
07-Oct-2025	Tuesday	Completion of core carbon emission calculation logic	3
10-Oct-2025	Friday	Update of Bridge and Traffic UI with backend connectivity	3
13-Oct-2025	Monday	Verification of calculations and optimization of SQL queries	3
17-Oct-2025	Friday	Update of Demolition and Recycling backend and UI refinement	3
21-Oct-2025	Tuesday	Database connectivity for Demolition and Recycling inputs	3
23-Oct-2025	Thursday	Development of Demolition and Recycling calculation module	4
27-Oct-2025	Monday	Comprehensive review and correction of all calculation formulas	3
30-Oct-2025	Thursday	Resolution of Foundation widget UI bug	5
04-Nov-2025	Tuesday	System-wide testing and identification of calculation errors	3
07-Nov-2025	Friday	Application of calculation corrections and finalization of Bridge and Traffic module	4
11-Nov-2025	Tuesday	Implementation of Value of Time (VOT) and Vehicle Operating Cost (VOC) calculations	3
13-Nov-2025	Thursday	Writing VOC calculation scripts and development of test UI	4
15-Nov-2025	Saturday	Research on VOT factors including vehicle types and lane configurations	3
17-Nov-2025	Monday	Development of comprehensive VOT calculation file	4
23-Nov-2025	Sunday	Refinement of formulas and implementation of requested changes	3
26-Nov-2025	Wednesday	Final verification of complex mathematical models	3

30-Nov-2025	Sunday	Completion and submission of all final calculation files	—

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.