FOSSEE Winter 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

August 26, 2025

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.

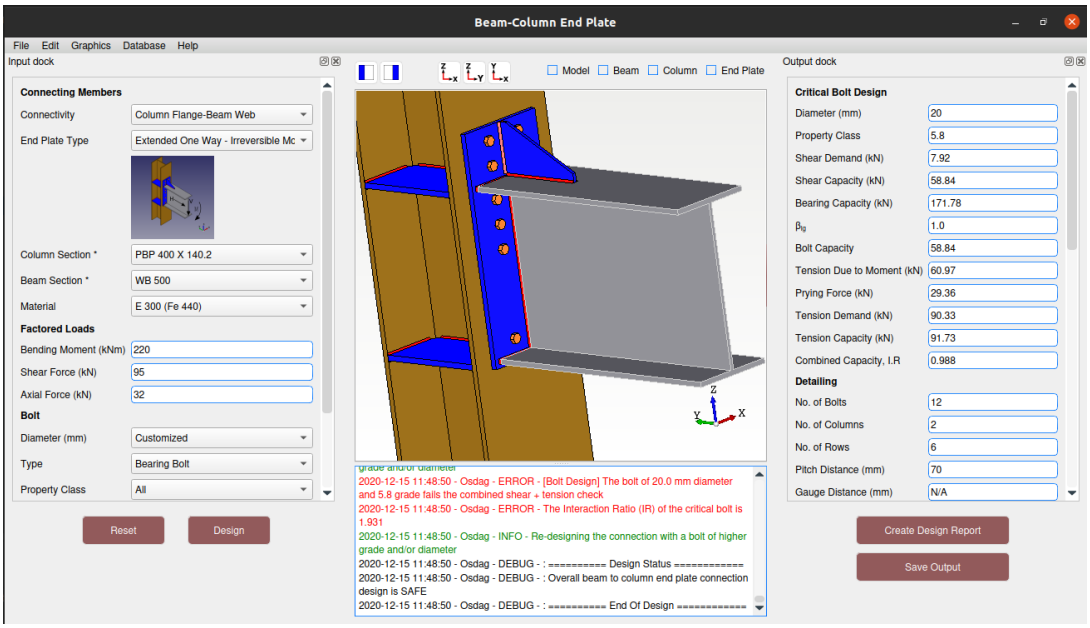

Figure 1.2: Osdag GUI

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

Problem Statement

2.1 Screening Task

The objective of this screening task is to develop a program that calculates the **Shear Force** and **Bending Moment** for a simply supported beam subjected to two moving point loads.

Preference will be given to solutions that make use of the **Influence Line Diagram (ILD)** approach to determine the critical effects at specific points along the beam.

Design Parameters

The problem is defined with the following input parameters:

- **Length of Beam (L)**: The total span of the beam, measured in meters.
- **Moving Load Values (W_1, W_2)**: The magnitudes of the two point loads, measured in kilonewtons (kN).
- **Distance Between Loads (x)**: The spacing between load W_1 and load W_2, in meters.
- **Position from Support A (a)**: The distance from the left support (Point A) to the first load W_1, in meters.

2.2 Tasks Done

As part of the screening task, the following activities were completed:

1. **Analytical Calculations using Python:** Developed a Python-based program to calculate key structural responses for a simply supported beam under two moving point loads. The program computes:

   - Reactions at the supports
   - Shear Force at a given point
   - Bending Moment at a given point
   - Maximum internal forces along the span

   This allows engineers and students to quickly evaluate load impact and identify critical locations.

2. **Visualization using Manim:** Implemented an animated visualization using the Manim library to illustrate the step-by-step solution process. The animation includes:

   - A labeled schematic of the beam with supports and moving loads
   - Step-by-step derivation of support reactions using equilibrium equations
   - Dynamic construction of Shear Force Diagram (SFD) and Bending Moment Diagram (BMD)

   This provides a clear visual understanding of how structural responses evolve as loads move across the beam.

3. **Integration of Learning with Visualization:** By combining numerical calculations with animated demonstrations, the project enhances conceptual clarity and offers an engaging pedagogical tool for students, educators, and practicing engineers.

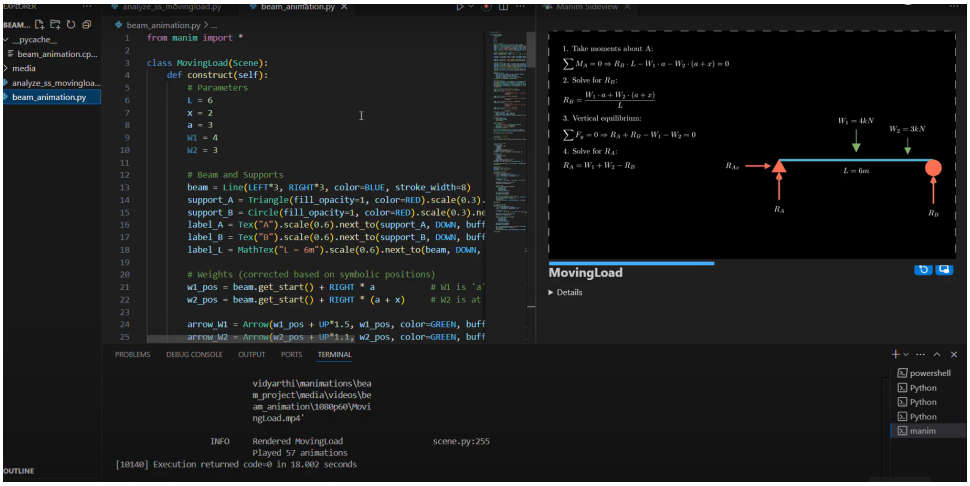

Figure 2.1: Animated visualization of the beam with moving loads and corresponding SFD & BMD.

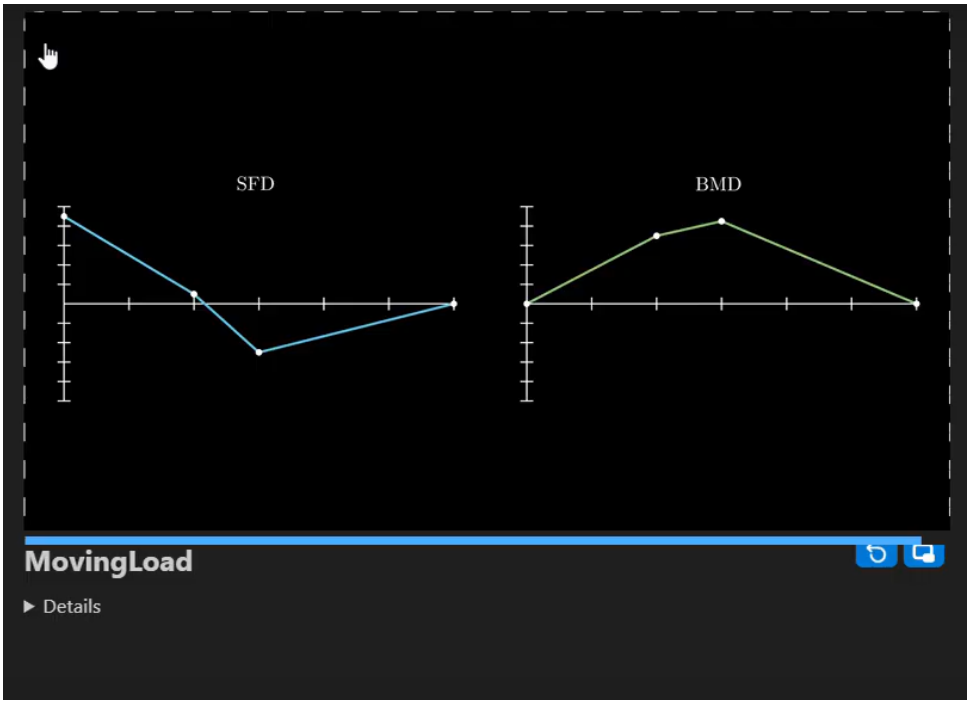

Figure 2.2: Animated visualization of the beam with moving loads and corresponding SFD & BMD.

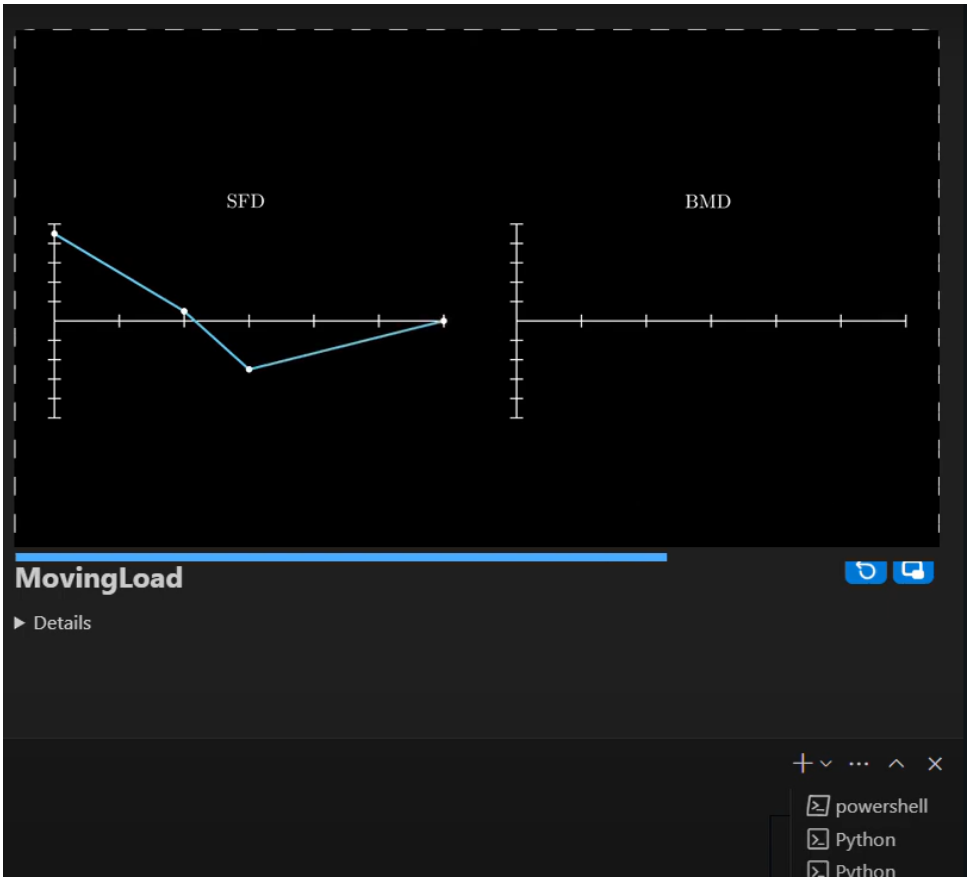

Figure 2.3: Animated visualization of the beam with moving loads and corresponding SFD & BMD.

Chapter 3

Internship Task 1: Reimagining Osdag

3.1 Task 1: Problem Statement

Create a basic UI template of the Home Page and a particular Module Design Page (you may add a few more pages if you want to present something creative or revamp) - You may create these on Figma or similar open-source tools - Create a presentation at the end, which should include: - First Page as the main landing page when Osdag opens (write your name at the bottom of the first page) - Consequently, one slide per page - The last slide should summarise your "reimagining philosophy" in a few bullet points.

3.2 Task 1: Tasks Done

I made a Figma file and redesigned the home page and module page. The redesigned UI is shown below:

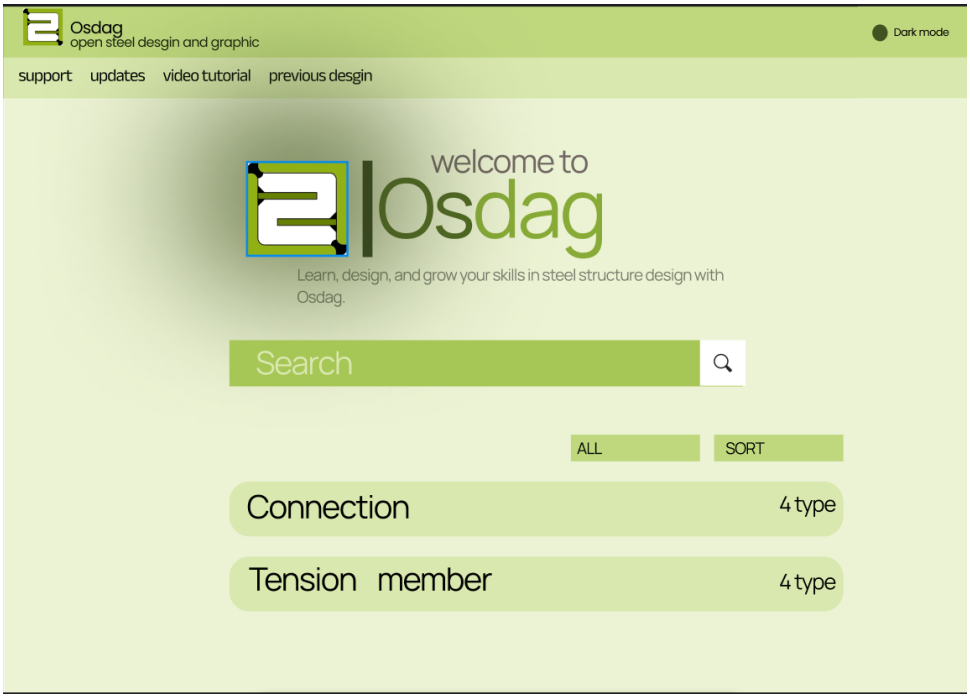

Figure 3.1: Redesigned Osdag Home Page (Figma Design)

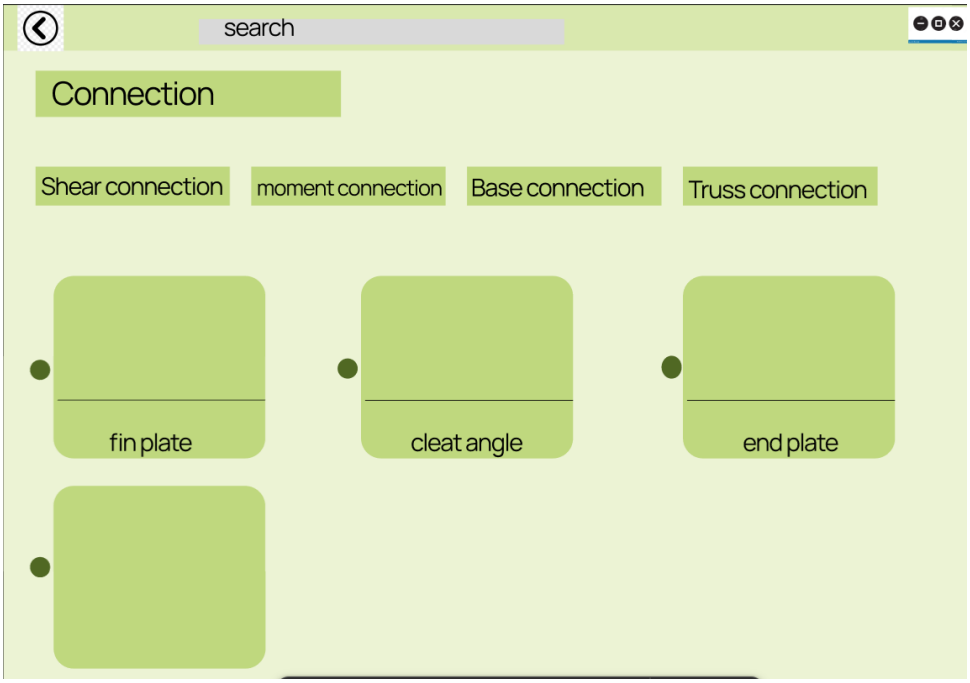

Figure 3.2: Redesigned Osdag Module Page (Figma Design)

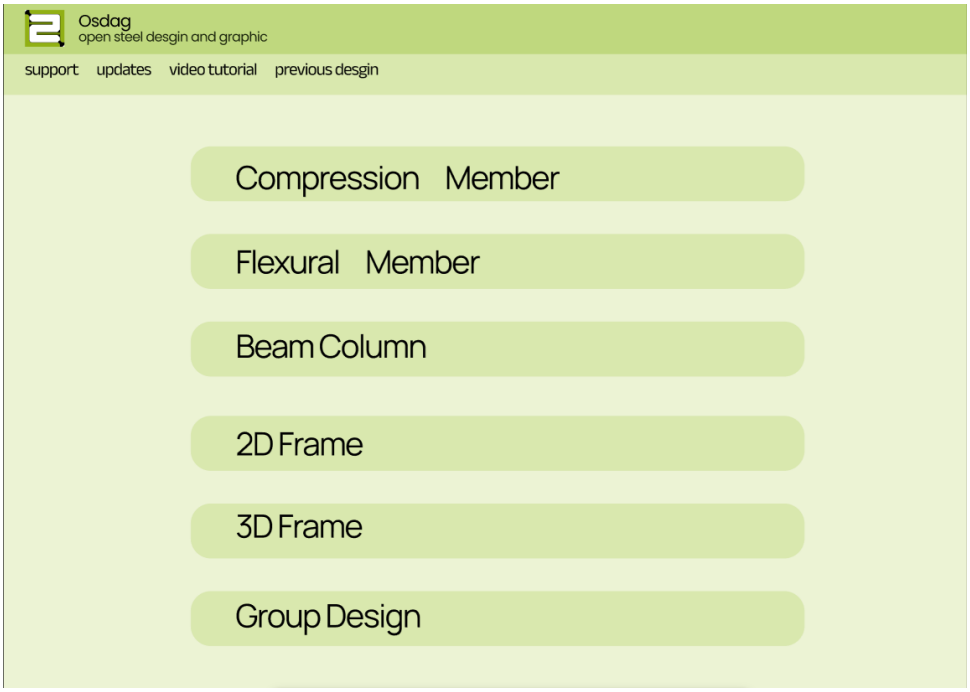

Figure 3.3: Redesigned Osdag Module Page (Figma Design)

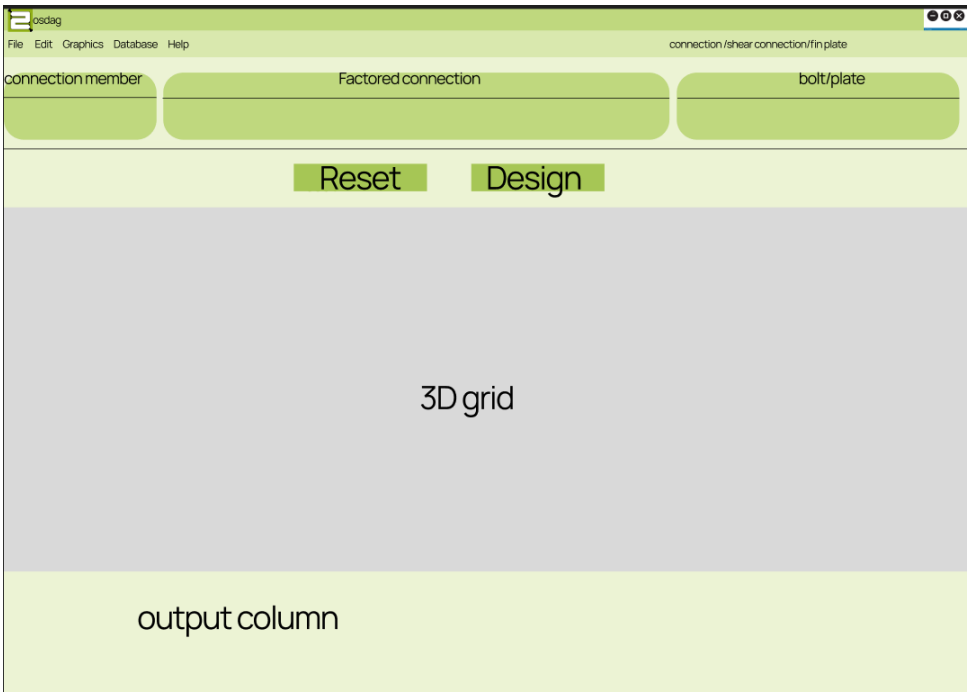

Figure 3.4: Redesigned Osdag Module Page (Figma Design)

Chapter 4

Internship Task 2: LCC Desktop-app - Frontend

4.1 4.1 Task 2: Objective

The task was to develop the desktop application of the LCC (Life Cycle Costing) tool from scratch by using PySide (Python Qt framework). The objective was to create a user-friendly, interactive, and visually appealing interface that enables users to perform Life Cycle Costing analysis efficiently. The desktop app was designed to provide better performance, offline accessibility, and seamless integration with the backend modules.

4.2 4.2 Task 2: Tasks Completed

Setup and Initial Development

- Successfully set up the LCC desktop application environment using Python and PySide6.
- Designed and implemented the main landing window of the application with a clean and intuitive layout.
- Created an interactive **accordion-style widget**, where clicking on "Input Parameters" dynamically expands and collapses the corresponding section.

Form Development and Navigation

- Developed the following primary input forms as individual PySide6 widgets:
  - Structure Works Data (Maintenance and Repair Data)
  - Bridge and Traffic
  - Demolition and Recycling
  - Carbon Emission Data
  - Carbon Emission Cost Data
- Within the **Structure Works Data** module, implemented the following sub-forms as nested widgets:
  - Foundation Form
  - Superstructure Form
  - Substructure Form
  - Miscellaneous Form
- Enabled seamless navigation between forms with **Next** and **Back** button functionality.
- Added a **popup confirmation dialog** for saving form data before proceeding to the next section, ensuring data integrity.

4.3 4.3 Task 2: Documentation

1. Project Structure (Desktop App)

The frontend code for the LCC desktop application is implemented in Python using PySide6. The source code is located at:

`src/osbridgelcca/desktop_app/ui/`

The directory is organized as follows:

- **resources/** – Contains SVG icons for window controls (close, minimize, maximize, restore).

- **themes/** – Includes style sheets for dark mode (`dark_mode.qss`) and light mode (`light_mode.qss`).
- **widgets/** – Contains the main UI components, structured as:
  - **carbon_emission_data/** – Modules for handling carbon-related inputs:
    * `carbon_emission_data.py`
    * `carbon_emission_cost_data.py`
  - **structure_works_data/** – Contains multiple sub-forms for structural details:
    * `foundation_widget.py`
    * `super_structure_widget.py`
    * `sub_structure_widget.py`
    * `auxiliary_works_widget.py`
  - **Other widgets:**
    * `bridge_and_traffic_data.py`
    * `demolition_and_recycling_data.py`
    * `financial_data.py`
    * `maintenance_repair_data.py`
    * `project_details_left_widget.py`
    * `project_details_right_widget.py`

2. Setup and Running the Application

The application can be launched by running the main Python entry script:

```
python app.py
```

Dependencies are managed through `requirements.txt`, ensuring smooth installation and environment setup.

3. Technologies Used

- **Python 3.x** – Core programming language
- **PySide6** – GUI framework for building desktop interfaces

- **Qt Stylesheets (.qss)** – Used for applying dark and light themes
- **SVG Resources** – Icons for window management

4. Component Organization and Functionality

- **Project Details Widgets:** Capture general project metadata (project name, location, etc.) before accessing detailed forms.
- **Form Navigation:** Integrated navigation system with **Next** and **Back** functionality.
- **Popup Confirmation:** Ensures that form data is saved before proceeding.
- **Dynamic Dropdowns:** Dropdown menus adapt dynamically based on user input and material selection.
- **Structure Works Data:** Includes modular forms for foundation, superstructure, substructure, and auxiliary works.
- **Other Forms:**
  - `financial_data.py`
  - `maintenance_repair_data.py`
  - `bridge_and_traffic_data.py`
  - `demolition_and_recycling_data.py`
  - `carbon_emission_data.py`
  - `carbon_emission_cost_data.py`

5. Project Structure Diagram

The following figure illustrates the directory organization of the desktop application:

Figure 4.1: Directory structure of the LCC Desktop Application (PySide6).

6. Application Screenshots

The following figures present some key interfaces of the developed LCC desktop application:

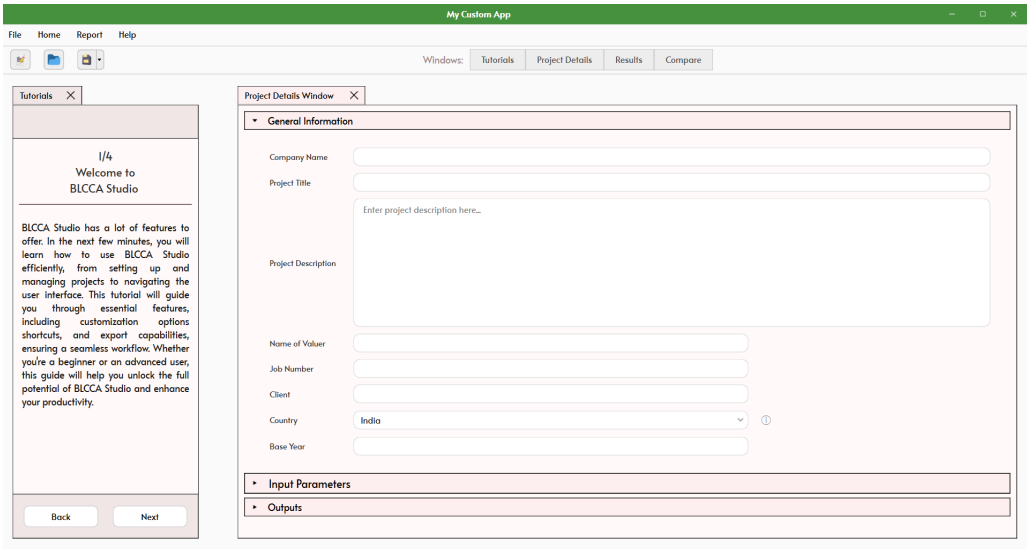

Figure 4.2: Landing page of the LCC Desktop Application.

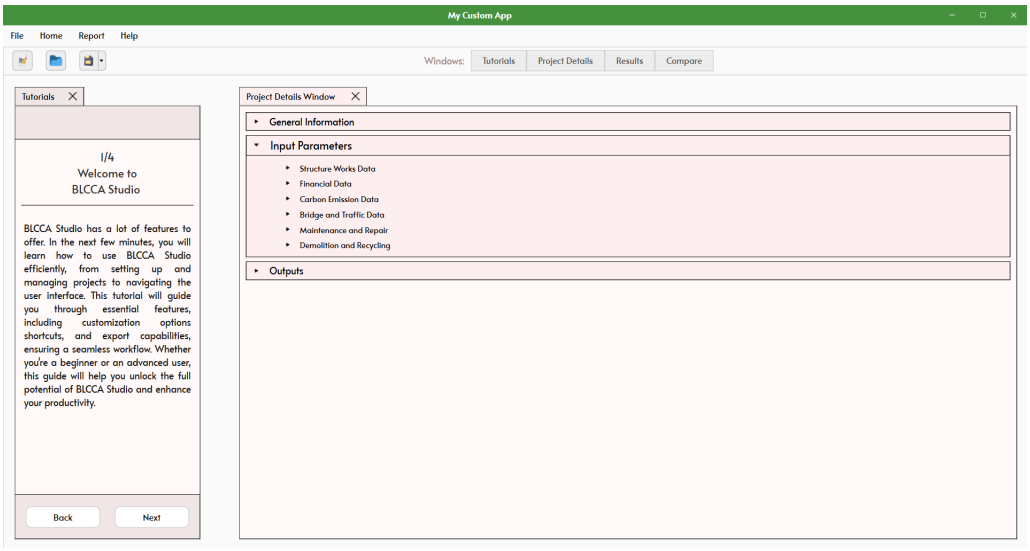

Figure 4.3: Accordion-style Input Parameters section.

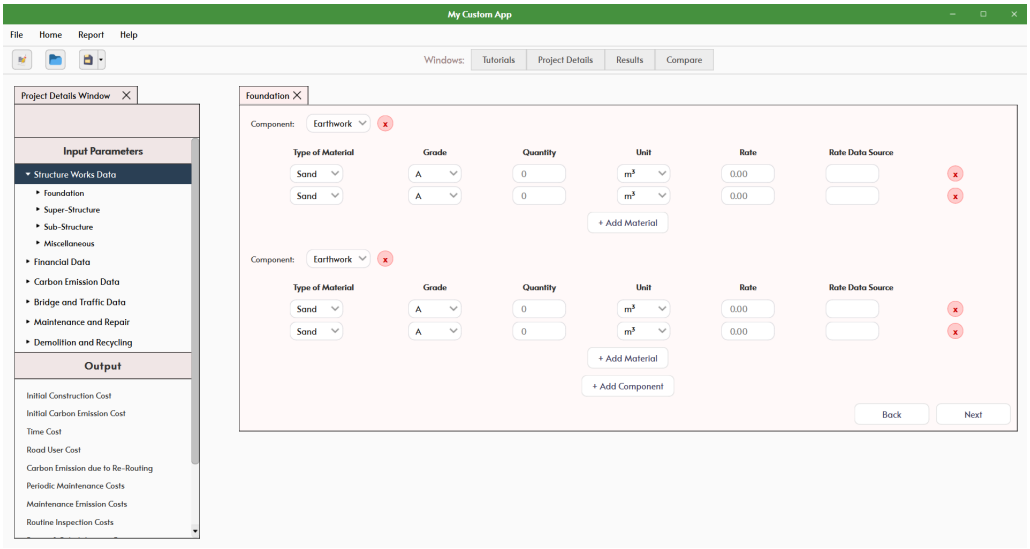

Figure 4.4: Example of a form in the Structure Works Data module.

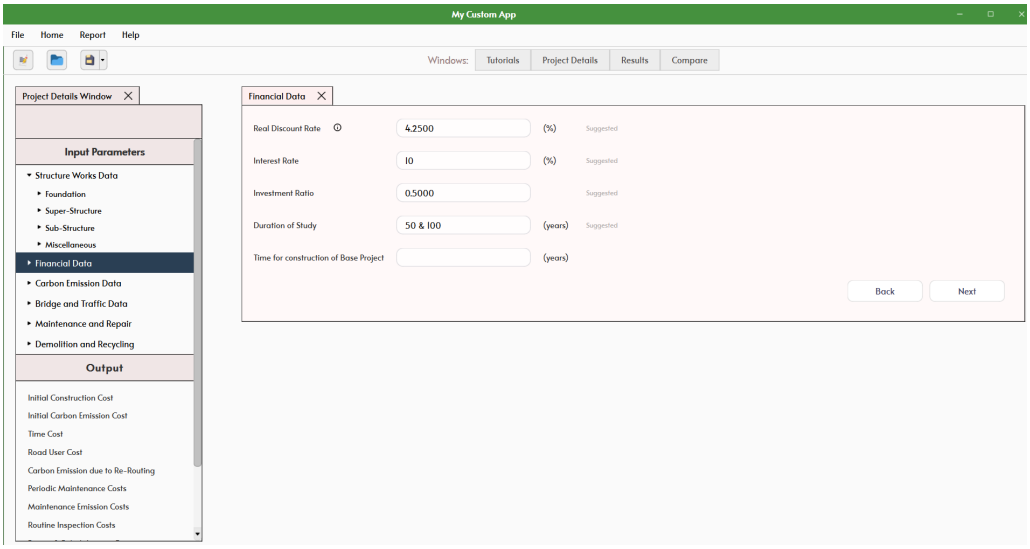

Figure 4.5: Popup confirmation dialog for saving form data.

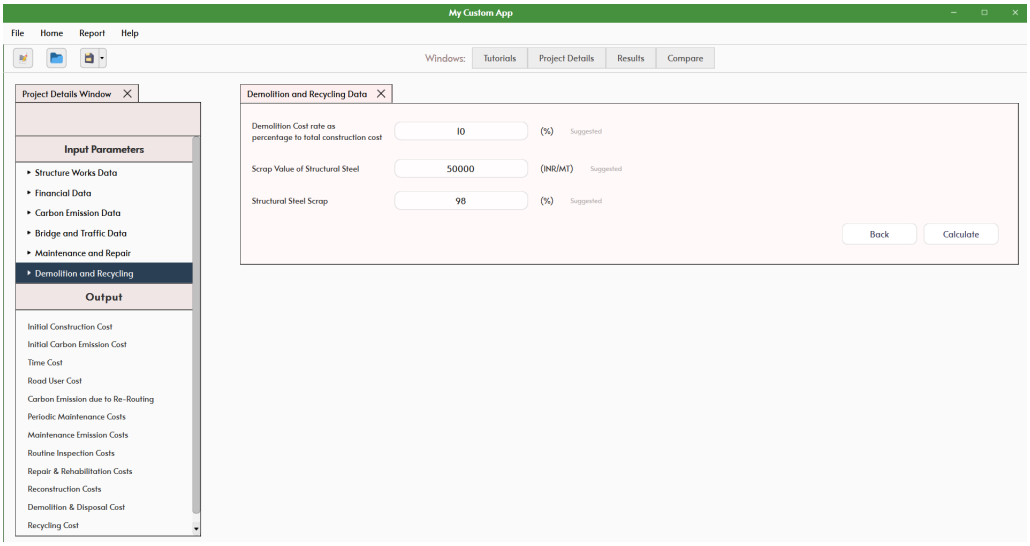

Figure 4.6: Popup confirmation dialog for saving form data.

Chapter 5

Conclusions

5.1 Tasks Accomplished

During the course of this internship, the following tasks were successfully completed:

- Developed the complete frontend for the LCC (Life Cycle Costing) desktop application using Python and PySide6, ensuring an interactive and user-friendly design.
- Designed and implemented multiple input forms as PySide6 widgets, including:
  - StructureWorksData (with sub-forms: *FoundationForm, SuperStructureForm, SubStructureForm, MiscellaneousForm*)
  - FinancialData
  - MaintenanceandRepairData
  - BridgeandTraffic
  - DemolitionandRecycling
  - CarbonEmissionData
  - CarbonEmissionCostData
- Integrated smooth form navigation with **Next/Back** functionality across all forms.
- Implemented popup confirmation dialogs to ensure data is saved before moving to the next section, maintaining data consistency.

- Developed dynamic dropdown menus that adapt based on user inputs and material selections, reducing manual errors.
- Built visualization components within the desktop app, including bar charts, pie charts, and radial charts, to represent both cost and carbon emission analyses.

5.2 Skills Developed

The internship contributed significantly to both technical and professional skill development:

Technical Skills

- Gained hands-on experience with PySide6 (Python Qt framework), focusing on widget-based desktop UI design.
- Improved proficiency in developing complex, modular input forms with interactive navigation for a desktop environment.
- Enhanced understanding of Qt Stylesheets (QSS) for applying light and dark themes in desktop applications.
- Acquired skills in integrating Python-based desktop frontends with backend logic for cost and emission calculations.
- Strengthened knowledge of data visualization in Python, using libraries to generate charts and graphs for cost analysis.

Professional Skills

- Learned to work with structured desktop application architectures while maintaining organized, reusable, and modular code.
- Improved problem-solving skills by handling real-time interactions between multiple forms, dropdowns, and datasets.
- Gained experience in project documentation and collaborative workflows, ensuring clarity in large-scale application development.

- Developed confidence in building offline, high-performance applications suitable for practical industry use.

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: 16-May-2025 to 30-July-2025

Date	Day	Task	Hours Worked
16-May-2025	Friday	Fellowship orientation. Discussion on LCC concepts and backend calculation approach.	5
17-May-2025	Saturday	Set up toolchain (VS Code, Git) and explored the Osdag environment.	5
18-May-2025	Sunday	Set up Osdag desktop app repository for LCC and created the initial database schema.	5
19-May-2025	Monday	Designed tables for carbon emission data and structural works data.	5
20-May-2025	Tuesday	Connected database with Python backend and tested sample queries.	5
21-May-2025	Wednesday	Explored existing Osdag widgets (financial_data, bridge_and_traffic_data, etc.).	5
22-May-2025	Thursday	Created carbon_emission_data.py and carbon_emission_cost_data.py scripts.	5
23-May-2025	Friday	Developed auxiliary_works_widget.py and integrated it with the database.	5
24-May-2025	Saturday	Added foundation_widget.py and tested with dummy inputs.	5
25-May-2025	Sunday	Weekend self-study: reviewed Osdag code structure and backend flow.	5
26-May-2025	Monday	Developed sub_structure_widget.py and super_structure_widget.py.	5
27-May-2025	Tuesday	Debugged integration issues with structure_works_data.	5
28-May-2025	Wednesday	Implemented demolition_and_recycling_data.py for demolition & recycling logic.	5
29-May-2025	Thursday	Designed UI for Project Details (Left & Right widgets).	5
30-May-2025	Friday	Connected financial_data.py with cost calculation backend.	5
31-May-2025	Saturday	Continued development of widgets for LCC input.	5
01-Jun-2025	Sunday	Self-study: worked on cost estimation methods and calculation logic.	5
02-Jun-2025	Monday	Implemented backend logic for life cycle costing in calculation scripts.	5
03-Jun-2025	Tuesday	Integrated calculation backend with frontend (Qt-based UI).	5
04-Jun-2025	Wednesday	Added report generation logic (LaTeX-based draft).	5
05-Jun-2025	Thursday	Created structured database for financial and maintenance data.	5

Date	Day	Task	Hours Worked
06-Jun-2025	Friday	Finalized LCC database schema (Projects, CostItems, EmissionFactors, WorkPackages, Maintenance, DemolitionRecycling). Seeded baseline data.	5
07-Jun-2025	Saturday	Discussed improvements for the desktop-app UI.	5
08-Jun-2025	Sunday	Built Carbon Emission Data widget (add/edit/import).	5
09-Jun-2025	Monday	Built Financial Data and Maintenance/Repair widgets.	5
10-Jun-2025	Tuesday	Implemented Structure Works widgets (Auxiliary, Foundation, Sub-Structure, Super-Structure).	5
11-Jun-2025	Wednesday	Developed Project Details widgets with dynamic dropdowns populated from DB and cascading selection logic.	5
12-Jun-2025	Thursday	Created Demolition & Recycling Data widget with unit normalization, conversion helpers, and data-integrity checks.	5
13-Jun-2025	Friday	Performed end-to-end data-flow testing across all widgets → DB → calculation stubs. Fixed validation/typing issues.	5
14-Jun-2025	Saturday	Debugged backend calculation errors.	5
15-Jun-2025	Sunday	Weekend: self-study on optimization of LCC algorithms.	5
16-Jun-2025	Monday	Documented initial module implementation.	5
17-Jun-2025	Tuesday	Improved frontend UI/UX based on mentor feedback.	5
18-Jun-2025	Wednesday	Debugged backend errors in database connections.	5
19-Jun-2025	Thursday	Mentor review session for the LCC module.	5
20-Jun-2025	Friday	Worked on backend calculation issues and code consistency.	5
21-Jun-2025	Saturday	Debugged errors in calculation formulas.	5
22-Jun-2025	Sunday	Weekend: completed backend integration with database.	5
23-Jun-2025	Monday	Fixed issues in cost calculation logic.	5
24-Jun-2025	Tuesday	Organized new directory structure under osbridgeLCC/src/modules/src/.	5
25-Jun-2025	Wednesday	Refactored input logic into separate modular components.	5
26-Jun-2025	Thursday	Debugged and resolved errors in module integration.	5
27-Jun-2025	Friday	Restructured code and connected all widgets with each other.	5
28-Jun-2025	Saturday	Continued integration of widgets with the database.	5
29-Jun-2025	Sunday	Weekend: validated workflow of refactored LCC module.	5
30-Jun-2025	Monday	Debugged minor issues and performed testing with teammates.	5
01–30 July 2025	July	Final debugging, UI polishing, mentor reviews, and submission of final deliverables.	5/day

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.