FOSSEE Summer Fellowship Report

On

Implementation and Restructuring of Modules in Osdag Web

Submitted by

Amay Dixit

2nd Year B.Tech Student, Department of Computer Science and Engineering

Indian Institute of Technology Bhilai

Bhilai, Chhattisgarh

Under the Guidance of

Prof. Siddhartha Ghosh

Department of Civil Engineering

Indian Institute of Technology Bombay

Mentors:

Parth Karia

Ajmal Babu M. S.

Ajinkya Dahale

July 15, 2025

Acknowledgments

- I would like to express my heartfelt gratitude to everyone who supported and encouraged me throughout the course of this internship. Their guidance and assistance played a vital role in the successful completion of my project.
- I am deeply thankful to the project staff at the Osdag team — Ajmal Babu M. S., Ajinkya Dahale, and Parth Karia — for their continuous support, valuable inputs, and timely feedback throughout the duration of my work.
- I extend my sincere thanks to Prof. Siddhartha Ghosh, Principal Investigator of the Osdag project and faculty member in the Department of Civil Engineering at IIT Bombay, for the opportunity to work under his esteemed guidance.
- I am also grateful to Prof. Kannan M. Moudgalya, Principal Investigator of the FOSSEE project and professor in the Department of Chemical Engineering at IIT Bombay, for enabling such learning experiences through the FOSSEE internship initiative.
- Special thanks to the FOSSEE managers, Ms. Usha Viswanathan and Ms. Vineeta Parmar, along with their entire team, for their constant administrative support and for ensuring the smooth functioning of the internship programme.
- I acknowledge the support provided by the National Mission on Education through Information and Communication Technology (ICT), Ministry of Education (MoE), Government of India, for facilitating this project and creating such opportunities for students across the country.
- Lastly, I am thankful to my college, the Indian Institute of Technology Bhilai, including the faculty, staff, and administration, for their constant encouragement and the academic environment that enabled me to pursue and complete this internship.

Contents

1 Introduction **4**

1.1 National Mission in Education through ICT 4

1.1.1 ICT Initiatives of MoE 5

1.2 FOSSEE Project 6

1.2.1 Projects and Activities 6

1.2.2 Fellowships 6

1.3 Osdag Software 7

1.3.1 Osdag GUI 8

1.3.2 Features 8

2 Screening Task **9**

2.1 Problem Statement 9

2.2 Tasks Done 10

3 Task 1: Reimagining the Osdag Web UI **12**

3.1 Task 1: Problem Statement 12

3.2 Task 1: Tasks Done 13

4 Task 2: Development of Cover Plate Welded to End Gusset Module **16**

4.1 Task 2: Problem Statement 16

4.2 Task 2: Tasks Done 16

4.3 Task 2: Documentation 19

4.3.1 Directory Structure 19

5 Task 3: Frontend Restructuring of Shear Connection Modules **21**

5.1 Task 3: Problem Statement 21

5.2 Task 3: Tasks Done 21

5.3 Task 3: Documentation 25

5.3.1 Directory Structure 25

6 Conclusions **27**

6.1 Tasks Accomplished 27

6.2 Skills Developed 28

A Appendix **30**

A.1 Internship Details 30

A.2 Work Reports 30

Bibliography **35**

Chapter 1

Introduction

1.1 National Mission in Education through ICT

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

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

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

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

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

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

1.1.1 ICT Initiatives of MoE

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

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

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

1.2 FOSSEE Project

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

1.2.1 Projects and Activities

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

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

1.2.2 Fellowships

FOSSEE offers various internship and fellowship opportunities for students:

- Winter Internship
- Summer Fellowship
- Semester-Long Internship

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

For more details, visit the official FOSSEE website.

Figure 1.1: FOSSEE Projects and Activities

1.3 Osdag Software

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

1.3.1 Osdag GUI

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

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

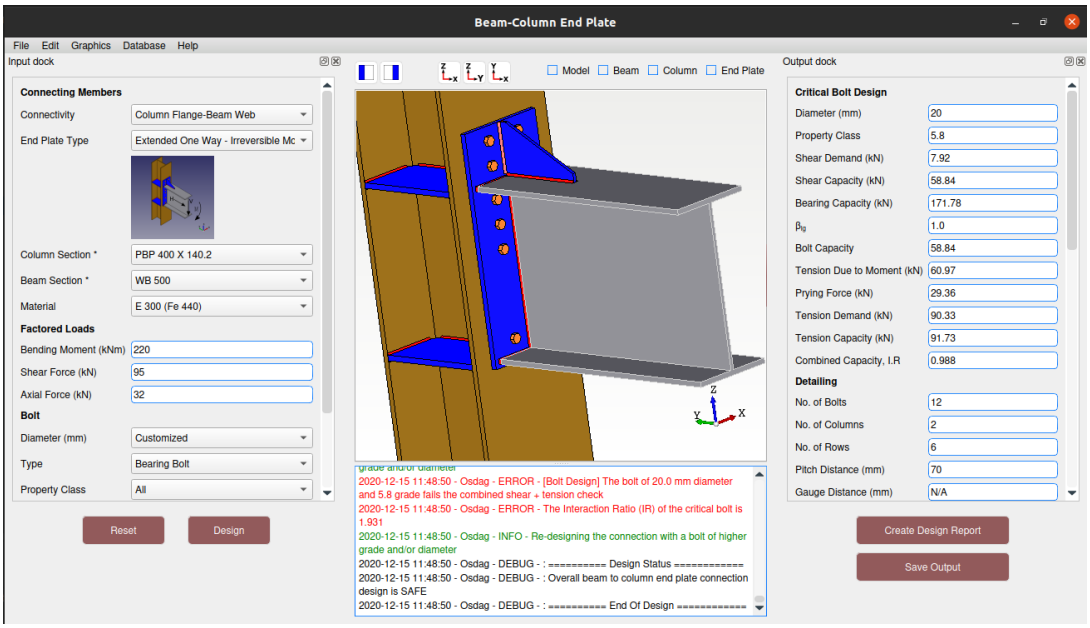

Figure 1.2: Osdag GUI

1.3.2 Features

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

For more details, visit the official Osdag website.

Chapter 2

Screening Task

2.1 Problem Statement

Area of Interest: Web Application Development

The screening task involved contributing to the Osdag web application by developing the UI in React and implementing backend endpoints using Django REST Framework. The goal was to port a structural steel connection module from the desktop version of Osdag to the web-based system. Available options included:

- Beam-to-beam splices – end plate
- Beam-to-column – end plate
- Column-to-column splices – cover plate (bolted/welded)
- Column to column – end plate
- Beam-to-beam – fin plate

I chose to implement the **Beam-to-Beam Fin Plate Connection** module, handling both frontend and backend development.

2.2 Tasks Done

Setup and Deployment

- Cloned and configured the Osdag-on-cloud repo locally with all dependencies (React, Django, PostgreSQL, Docker).
- Used Git for version control and maintained a clean development workflow.

Frontend Development

- Built the frontend using React (Vite) replicating the desktop UI for consistency.
- Core components:
  - **Input Dock:** Inputs for section properties, materials, bolts, and loading.
  - **Visualization Panel:** Placeholder for 3D views (three.js), with toggle support.
  - **Output Dock:** Shows computed design values and connection details.
  - **Control Buttons:** "Design", "Reset", "Save Output", "Generate Report".
- Followed a modular, functional architecture with React hooks.

Backend Development

- Studied the existing Django backend (osdagweb, osdag api).
- Implemented APIs for:
  - **Input Handling:** Validated and processed fin plate input data.
  - **Design Calculations:** Triggered via `designtype.connection.finplateconnection`.
  - **Output API:** Returned computed results to the frontend.
- Used Django REST Framework with serializers, views, and proper RESTful practices.
- Tested endpoints via Postman and ensured frontend compatibility.

Bug Fixes and Enhancements

- Fixed minor UI bugs in other modules (Cleat Angle, End Plate).
- Improved component styling and layout using CSS.

Challenges Tackled

- Understood the structural logic for fin plate design.
- Modularized tightly coupled backend logic into REST APIs.
- Bridged architectural differences between desktop and web versions.
- Resolved integration issues across frontend-backend data flow.

Deliverables

- Functional UI and backend logic for Beam-to-Beam Fin Plate Connection.
- Minor fixes in other modules.
- Documentation outlining architecture, decisions, and integration points.

Chapter 3

Task 1: Reimagining the Osdag Web UI

3.1 Task 1: Problem Statement

The first task of the internship involved reimagining and redesigning the user interface of the Osdag web application. The objective was to rethink the user journey, visual layout, and user interaction patterns of the platform, in line with modern design practices and usability standards.

We were expected to:

- Create a basic UI template of the Home Page and at least one module-specific Design Page.
- Use open-source tools such as Figma to create mockups.
- Ensure that the UI retained the core functionality and structural layout of the desktop version of Osdag, while improving visual clarity and responsiveness.
- Prepare a presentation including:
  - A landing page mockup.
  - Module page mockups.
  - A final slide summarizing the redesign philosophy and thought process.

3.2 Task 1: Tasks Done

UI Redesign on Figma

- Designed a new landing page for Osdag Web that emphasizes modularity, simplicity, and quick navigation to modules.
- Redesigned the Beam-to-Beam Fin Plate connection module page to incorporate a cleaner form layout and component-based display panels.
- Incorporated modern UI elements such as cards, tabs, tooltips, dropdowns, and collapsible panels to improve user experience.
- Created responsive layouts keeping scalability and multi-device usage in mind.
- Added side navigation for module switching and a top navigation bar for branding and user profile.
- Ensured accessibility and visual hierarchy with clear font sizes, consistent color schemes, and spacing.

Design Deliverables

- A Figma file containing:
  - Landing Page
  - Beam-to-Beam Fin Plate Module Page
  - Output Report Page (optional)
- A presentation with:
  - Slide 1: Landing Page
  - Slide 2: Module Page
  - Slide 3: Report Page (optional)
  - Final Slide: Design Philosophy

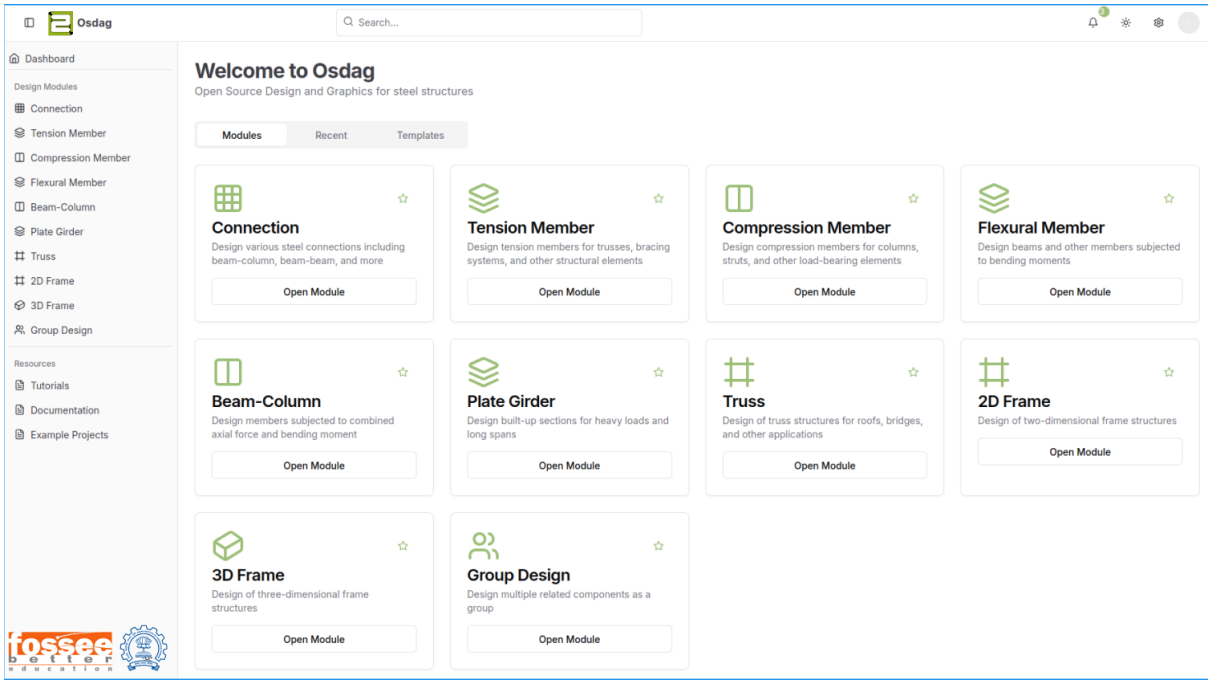

Figure 3.1: Redesigned Osdag Landing Page

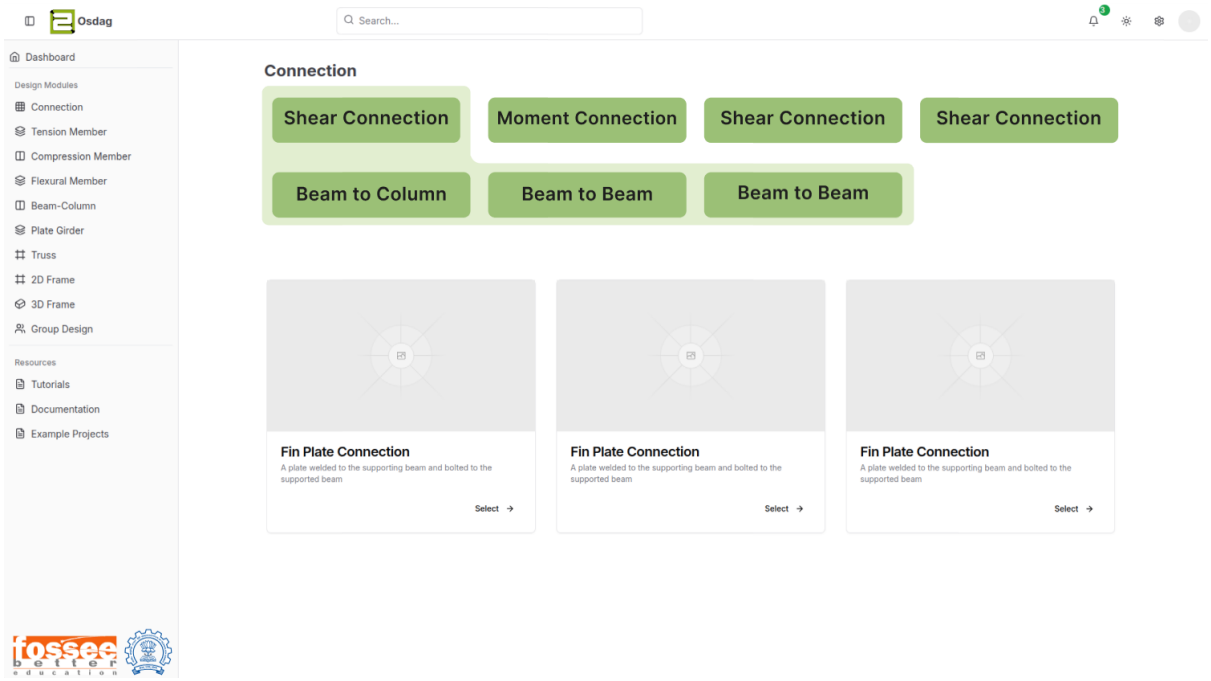

Figure 3.2: Redesigned Module Page

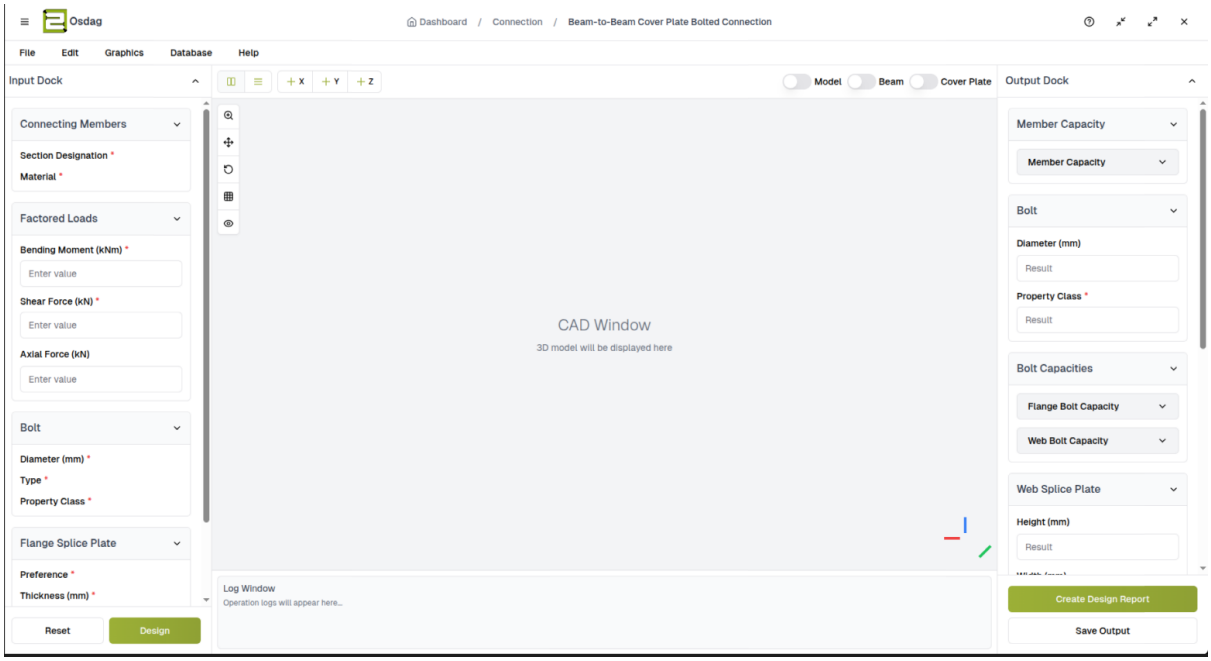

Figure 3.3: Redesigned CAD Page

Design Philosophy

- Emphasize clean and modular layouts for ease of use.
- Simplify user interactions by minimizing the number of clicks required.
- Maintain consistency with the desktop version but improve visual hierarchy.
- Make the UI responsive and accessible across different screen sizes.
- Use open-source and easily maintainable tools and frameworks.

Chapter 4

Task 2: Development of Cover Plate Welded to End Gusset Module

4.1 Task 2: Problem Statement

The second internship task involved the complete implementation of a new connection type — the Cover Plate Welded to End Gusset — in Osdag Web. This connection is used primarily in truss structures for transferring loads from bracing members to gusset plates via welded cover plates.

The objective was to:

- Understand the structural design and behavior of a gusset connection.
- Implement the module from scratch by adhering to Osdag's modular architecture.
- Build both the frontend UI components and the backend calculation logic.
- Generate design reports and CAD models for the module.
- Ensure compatibility with the Osdag web API architecture.

4.2 Task 2: Tasks Done

The entire module was developed from scratch and included full-stack involvement — backend calculations, REST API creation, UI integration, and 2D CAD drawing generation.

Understanding the Architecture and Framework

- Explored the existing Osdag web architecture comprising Django REST backend and React frontend.
- Identified relevant components including:
  - design_type/connection
  - osdag_api/modules/
  - osdag/web_api/inputdata/
  - osdagclient/components/
- Analyzed other connection modules (e.g., Cleat Angle, Fin Plate) to use as references.

Backend Development

- Implemented the connection logic in the Django backend under:
  - design_type/connection/gusset_connection.py
  - osdag_api/modules/gusset_connection.py
- Developed structural checks for weld length, gusset thickness, and load transfer.
- Created views and serializers under:
  - osdag/web_api/inputdata/gusset_connection_input.py
  - osdag/web_api/output_data_api.py
- Used Django REST framework for input validation and output formatting.
- Added automated unit tests for edge cases and validation.

Frontend Development

- Developed React components inside:
  - osdagclient/components/shearConnection/Gusset.jsx

- Built interactive input modals using reusable component structure (e.g., SectionModal, WeldSectionModal).
- Integrated with backend APIs using Axios.
- Rendered output values, warning messages, and result summaries in the OutputDock component.

CAD and Report Generation

- Developed CAD drawing generation logic under:
  - `report/gusset_connection_template.py`
- Leveraged FreeCAD scripting for generating 2D drawings.
- Used `Report_functions.py` to export calculated data to Excel and report PDFs.

Challenges and Solutions

- **Understanding Load Paths**: Resolved confusion by studying references and existing Osdag desktop modules.
- **FreeCAD Integration**: Faced multiple bugs due to object naming; resolved using consistent naming conventions and debugging in standalone FreeCAD.
- **UI Testing**: Ensured responsive inputs and clean separation of validation and display logic.

Deliverables

- Complete backend design and validation logic.
- Fully functioning frontend with input, output, and state management.
- PDF report and CAD drawing generation.
- Integration with Osdag's existing database and module structure.
- Documented Python code and API views.

4.3 Task 2: Documentation

4.3.1 Directory Structure

```
osdag-admin-osdag-web
├── design_type
│   └── connection
│       └── gusset_connection.py
├── osdag_api
│   └── modules
│       └── gusset_connection.py
├── osdag
│   └── web_api
│       ├── inputdata
│       │   └── gusset_connection_input.py
│       └── output_data_api.py
├── osdagclient
│   └── components
│       └── shearConnection
│           └── Gusset.jsx
└── report
    └── gusset_connection_template.py
```

Program Entry Point

The module is registered as a part of the Osdag connection module system. Once the web app is running, the gusset connection appears under the "Shear Connection" category.

Launching the Web App

```
# Backend
$ python manage.py runserver
```

```
# Frontend
$ cd osdagclient
$ npm install && npm run dev
```

Usage Flow

1. Navigate to the “Shear Connection” category.
2. Choose “Cover Plate Welded to End Gusset”.
3. Fill in structural and weld-related parameters.
4. Click “Design” to view calculated results.
5. Optionally generate report and 2D drawing.

Chapter 5

Task 3: Frontend Restructuring of Shear Connection Modules

5.1 Task 3: Problem Statement

The third internship task aimed to **refactor the frontend architecture** of existing shear connection modules — specifically **Fin Plate**, **End Plate**, and **Cleat Angle** — to improve maintainability, reusability, and scalability of the Osdag Web application.

The existing codebase had repetitive logic across modules, which made bug fixes and enhancements tedious. The goal was to:

- Abstract common logic and components into shared files.
- Modularize configuration files for easier testing and extension.
- Standardize state management and rendering flows across modules.
- Simplify the addition of future connection modules through a plug-and-play structure.

5.2 Task 3: Tasks Done

Understanding the Existing Codebase

- Studied the original implementations of Fin Plate, End Plate, and Cleat Angle modules under the ‘osdagclient/components/shearConnection‘ directory.

- Identified common patterns such as:
  - Repeated input rendering logic
  - Hardcoded output formatting
  - Redundant modal implementations
- Analyzed UI state management flow using React hooks and context (‘ModuleState.jsx‘, ‘ModuleReducer.jsx‘).

New Architecture Design

To decouple module-specific logic from reusable logic, a new directory structure was proposed under ‘src/modules/‘, split as follows:

- ‘shared/‘ — for reusable components and hooks
- One directory per module — with its own components and configuration

```
src/modules/
          shared/
                components/
                      EngineeringModule.jsx
                      InputSection.jsx
                      BaseOutputDock.jsx
                      CustomizationModal.jsx
                      DesignReportModal.jsx
                hooks/
                    useEngineeringModule.js
          finPlate/
                FinPlate.jsx
                configs/
                      finPlateConfig.js
                      finPlateOutputConfig.js
                components/
                    FinPlateOutputDock.jsx
          endPlate/
                EndPlate.jsx
                configs/
                      endPlateConfig.js
                      endPlateOutputConfig.js
                components/
                    EndPlateOutputDock.jsx
          cleatAngle/
              CleatAngle.jsx
              configs/
                    cleatAngleConfig.js
                    cleatAngleOutputConfig.js
              components/
                  CleatAngleOutputDock.jsx
```

Figure 5.1: New frontend directory structure after refactor

Shared Component Design

- **EngineeringModule.jsx** — A master wrapper that accepts configuration objects and dynamically renders forms, API calls, and output.
- **InputSection.jsx** — Dynamically renders input fields based on config JSON.
- **BaseOutputDock.jsx** — Used to wrap module-specific output docks with consistent layout and behavior.
- **DesignReportModal.jsx** — Unified modal to preview and download design reports.

- **CustomizationModal.jsx** — Handles weld and bolt detailing for all modules.

Custom Hooks and Config Separation

- **useEngineeringModule.js** — Manages all logic like input state, output fetching, and error handling in a reusable hook.
- **Module Configs** — Each module (e.g., finPlate, endPlate) contains:
  - `<module>Config.js` — Defines input parameters and form sections.
  - `<module>OutputConfig.js` — Defines how to render output keys, formulas, and units.

Individual Module Refactors

- **Fin Plate**
  - Moved logic from ‘FinPlate.jsx‘ to use ‘EngineeringModule‘.
  - Removed hardcoded modals and replaced with shared components.
- **End Plate**
  - Abstracted complex logic for bolt layout display.
  - Integrated shared design modal and output dock.
- **Cleat Angle**
  - Split form rendering into input config files.
  - Unified logic for shear check warnings using ‘BaseOutputDock‘.

Benefits of the New Architecture

- Faster development of new modules (only config + output dock needed).
- Easier bug fixes — shared logic centralized.
- Improved code readability and file organization.
- Standardized output rendering and report generation across modules.

5.3 Task 3: Documentation

5.3.1 Directory Structure

```
src/modules
├── shared
│   ├── components
│   │   ├── EngineeringModule.jsx
│   │   ├── InputSection.jsx
│   │   ├── BaseOutputDock.jsx
│   │   ├── CustomizationModal.jsx
│   │   └── DesignReportModal.jsx
│   └── hooks
│       └── useEngineeringModule.js
├── finPlate
│   ├── FinPlate.jsx
│   ├── configs
│   │   ├── finPlateConfig.js
│   │   └── finPlateOutputConfig.js
│   └── components
│       └── FinPlateOutputDock.jsx
├── endPlate
│   ├── EndPlate.jsx
│   ├── configs
│   │   ├── endPlateConfig.js
│   │   └── endPlateOutputConfig.js
│   └── components
│       └── EndPlateOutputDock.jsx
└── cleatAngle
    ├── CleatAngle.jsx
    ├── configs
    │   ├── cleatAngleConfig.js
    │   └── cleatAngleOutputConfig.js
```

Usage Flow

1. Developer adds a new module directory inside ‘src/modules‘.
2. Defines input/output configuration in JSON-like JS objects.
3. Creates a JSX wrapper that invokes `<EngineeringModule />` with appropriate config.
4. Optionally overrides output dock with module-specific JSX.

Future Scope

- The architecture allows plug-in style module integration.
- Potential to migrate remaining modules (moment, seated angle, etc.) to the same structure.
- Could integrate international design codes in the same framework via config extensions.

Chapter 6

Conclusions

6.1 Tasks Accomplished

Over the course of the FOSSEE Summer Fellowship, I successfully contributed to the development and enhancement of the Osdag Web application by completing the following major tasks:

- **Task 1: Reimagining the Osdag Web UI**
  Conducted a design review of the existing Osdag Web UI and proposed a cleaner, modular, and scalable UI architecture. This formed the foundation for future restructuring and simplified the process of adding new modules.

- **Task 2: Development of Cover Plate Welded to End Gusset Module**
  Designed and implemented a complete structural engineering module from scratch. This included developing backend logic for design calculations and validation, frontend components for user interaction, REST API integration, and CAD/report generation capabilities using FreeCAD and Python.

- **Task 3: Frontend Restructuring of Shear Connection Modules**
  Refactored the Fin Plate, End Plate, and Cleat Angle modules by abstracting shared logic into reusable components and separating configuration from logic. This restructuring improved code maintainability, scalability, and developer productivity.

- **Additional Contributions**
  Addressed multiple bugs across UI components, resolved integration issues between

frontend and backend modules, and contributed to styling improvements and UI consistency across Osdag Web.

These tasks were completed while maintaining compatibility with the existing codebase and following good software engineering practices, including modular design, reusable code, and documentation.

6.2 Skills Developed

This fellowship provided me with a unique opportunity to work on a real-world open-source structural engineering software project. Through this experience, I developed the following technical and professional skills:

Technical Skills

- **Full Stack Web Development:** Gained hands-on experience in working with a full-stack architecture using Django REST Framework (backend) and React.js with Vite (frontend).
- **Frontend Architecture and Refactoring:** Learned how to design scalable and maintainable component hierarchies, use configuration-driven development, and implement reusable logic using custom hooks in React.
- **Structural Engineering Computation:** Understood the engineering logic behind structural connections such as fin plates, end plates, and gusset connections, and translated these into programmatic validations and calculations.
- **CAD Automation and Report Generation:** Used FreeCAD for 2D drawing generation and Python scripts for Excel/PDF report export.
- **Testing and Debugging:** Wrote unit tests for backend logic, debugged FreeCAD integration issues, and ensured robust UI performance through manual and component-level testing.
- **Version Control and Collaboration:** Worked with Git, GitHub, and followed collaborative workflows to manage contributions, resolve merge conflicts, and follow review cycles with mentors.

Professional Skills

- **Project Understanding and Planning:** Learned to analyze large codebases, break down complex tasks, and plan development phases effectively while ensuring compatibility with existing systems.
- **Collaboration and Communication:** Engaged in regular discussions with mentors and project leads, incorporated feedback iteratively, and maintained proper documentation to ease onboarding for future contributors.
- **Problem Solving and Adaptability:** Tackled diverse challenges — from engineering calculations to frontend optimization — while adapting to new technologies and tools quickly.

Overall, this internship was a significant learning experience that bridged my academic foundation with real-world software development, deepened my understanding of open-source practices, and encouraged me to contribute more meaningfully to community-driven engineering tools.

Chapter A

Appendix

A.1 Internship Details

Name	Amay Dixit
Project	Osdag
Fellowship	FOSSEE Summer Fellowship 2025
Duration	May 15, 2025 – July 15, 2025
Type	Full-time internship (6 days/week, Sundays off)

A.2 Work Reports

DATE	**DAY**	**TASK**	**Hours Worked**
15-May-2025	Thursday	Project setup, cloned Osdag Web repository, and explored codebase	7
16-May-2025	Friday	Understood Django-React integration and existing backend structure	8
17-May-2025	Saturday	Studied shear connection modules: Cleat Angle, Fin Plate	7
18-May-2025	Sunday	Understood gusset connection behavior and design logic	8

Continued on next page

Table A.1 – continued from previous page

DATE	DAY	TASK	Hours Worked
19-May-2025	Monday	Drafted initial backend structure and planned module endpoints	7
20-May-2025	Tuesday	Designed input data schema and started writing gusset input API	8
21-May-2025	Wednesday	Created gusset calculation logic and validated with test cases	8
22-May-2025	Thursday	Integrated gusset backend module into API framework	8
23-May-2025	Friday	Implemented gusset calculation validations and edge cases	8
24-May-2025	Saturday	Developed CAD drawing logic using FreeCAD scripting	7
25-May-2025	Sunday	Debugged FreeCAD drawing output and ensured accuracy	8
26-May-2025	Monday	Designed frontend layout and created Gusset.jsx component	7
27-May-2025	Tuesday	Integrated backend with frontend using Axios	8
28-May-2025	Wednesday	Built parameter input modal and report generation UI	8
29-May-2025	Thursday	Connected frontend inputs to API and tested interaction flow	7
30-May-2025	Friday	Validated end-to-end design flow and made UI enhancements	7
31-May-2025	Saturday	Added error handling, warning messages, and result summaries	8
01-Jun-2025	Sunday	Conducted unit tests and fixed backend validation issues	8

Continued on next page

Table A.1 – continued from previous page

DATE	DAY	TASK	Hours Worked
02-Jun-2025	Monday	Added export-to-report and export-to-CAD functionalities	8
03-Jun-2025	Tuesday	Finalized module structure and ensured API consistency	7
04-Jun-2025	Wednesday	Wrote README and documentation for the gusset module	8
05-Jun-2025	Thursday	Peer-reviewed and refactored gusset connection codebase	8
06-Jun-2025	Friday	Mentor feedback received and integrated into the module	7
07-Jun-2025	Saturday	Conducted test cases for large/small load edge cases	8
08-Jun-2025	Sunday	Created final report and CAD templates for gusset	8
09-Jun-2025	Monday	Cleaned up code and ensured documentation completeness	7
10-Jun-2025	Tuesday	Final end-to-end testing of the gusset module	8
11-Jun-2025	Wednesday	Initiated restructuring of Fin Plate module	7
12-Jun-2025	Thursday	Refactored FinPlate.jsx and moved config to separate file	8
13-Jun-2025	Friday	Created reusable OutputDock for Fin Plate module	8
14-Jun-2025	Saturday	Used shared hooks for state and validation logic	7
15-Jun-2025	Sunday	Improved modularity of input form components	8
16-Jun-2025	Monday	Updated Fin Plate styling and completed re-structuring	7
17-Jun-2025	Tuesday	Bug testing and component cleanup for Fin Plate module	7

Continued on next page

Table A.1 – continued from previous page

DATE	DAY	TASK	Hours Worked
18-Jun-2025	Wednesday	Peer review and feedback incorporation	7
19-Jun-2025	Thursday	Reviewed integration with OutputDock and DesignModals	7
20-Jun-2025	Friday	Started restructuring End Plate module	8
21-Jun-2025	Saturday	Refactored EndPlate.jsx and config separation	8
22-Jun-2025	Sunday	Created OutputDock for End Plate, integrated shared components	8
23-Jun-2025	Monday	Updated modals and tested data consistency	7
24-Jun-2025	Tuesday	Completed refactor and cleanup of End Plate components	7
25-Jun-2025	Wednesday	Began Cleat Angle module restructuring	8
26-Jun-2025	Thursday	Separated config, reused InputSection and OutputDock	8
27-Jun-2025	Friday	Ensured validation and UI state were decoupled	8
28-Jun-2025	Saturday	Finalized Cleat Angle restructuring and tested integration	8
29-Jun-2025	Sunday	Cross-checked backend compatibility and handled UI bugs	8
30-Jun-2025	Monday	Wrote documentation for all restructured modules	7
01-Jul-2025	Tuesday	Revisited Gusset module and refactored for uniformity	8
02-Jul-2025	Wednesday	Used shared components and hooks in Gusset module	7
03-Jul-2025	Thursday	Created config structure for Gusset module	7
04-Jul-2025	Friday	Updated OutputDock and modal structure	8
05-Jul-2025	Saturday	UI cleanup and responsiveness improvements	8

Continued on next page

Table A.1 – continued from previous page

DATE	DAY	TASK	Hours Worked
06-Jul-2025	Sunday	Finalized Gusset restructuring and tested edge cases	7
07-Jul-2025	Monday	Fixed styling and validation bugs across modules	7
08-Jul-2025	Tuesday	Wrote final documentation and updated README	7
09-Jul-2025	Wednesday	Peer review and bug fixes in multiple modules	7
10-Jul-2025	Thursday	Final polishing of UI and report generation bugs	7
11-Jul-2025	Friday	Feedback meeting with mentors and changes based on suggestions	4
12-Jul-2025	Saturday	Minor patch updates to CAD generation logic	4
13-Jul-2025	Sunday	Testing of complete end-to-end design flow	7
14-Jul-2025	Monday	Prepared project summary and final report	7
15-Jul-2025	Tuesday	Final mentor review, submitted internship report	4

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.