FOSSEE Winter Internship Report

On

Development of LCC module for OSDAG

Submitted by

Sakshi Shamrao Jadhav

3rd Year B.Tech Student, Department of Electrical Engineer

IIT Bhilai

Mumbai

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

July 15, 2025

Acknowledgments

I would like to express my sincere gratitude to everyone who supported and guided me during the course of this internship.

I am deeply thankful to the project staff at the Osdag team, particularly Mr. Parth Karia, for his constant guidance, timely feedback, and invaluable technical insights, which greatly helped me in understanding the workflow and successfully completing the assigned tasks. I also extend my thanks to Mr. Ajmal Babu M. S. and Mr. Ajinkya Dahale for their assistance and support throughout the project.

I would like to express my sincere appreciation to Prof. Siddhartha Ghosh, Principal Investigator (PI) of the Osdag Project, Department of Civil Engineering, IIT Bombay, for providing me with this excellent opportunity to work on the Life Cycle Costing (LCC) project.

I also wish to acknowledge Prof. Kannan M. Moudgalya, Principal Investigator of the FOSSEE Project, Department of Chemical Engineering, IIT Bombay, for initiating and facilitating this program. My heartfelt thanks to FOSSEE Managers, Ms. Usha Viswanathan and Ms. Vineeta Parmar, and their entire team for their support and smooth coordination during the internship.

I gratefully acknowledge the National Mission on Education through Information and Communication Technology (ICT), Ministry of Education (MoE), Government of India, for supporting this project under the FOSSEE initiative.

Lastly, I extend my thanks to my colleagues and fellow interns for their cooperation and discussions, and to my college, IIT Bhilai, for providing the necessary academic foundation and encouragement to participate in 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 9

3 Internship Task 1: Reimagining Osdag **11**

3.1 Task 1: Problem Statement 11

3.2 Task 1: Tasks Done 11

4 Internship Task 2: LCC Website - Frontend **13**

4.1 Task 2: Objective 13

4.2 Task 2: Tasks Completed 13

4.3 Task 2: Documentation 14

5 Internship Task 3: LCC Website - Frontend **19**

5.1 Task 3: Objective 19

5.2 Task 3: Tasks Completed 19

5.3 Task 3: Output Visuals 22

6 Internship Task 4: LCC Website - Backend-Frontend Integration **24**

6.1 Task 4: Objective 24

6.2 Task 4: Tasks Completed 24

6.3 Task 4: Documentation 26

7 Conclusions **28**

7.1 Tasks Accomplished 28

7.2 Skills Developed 29

A Daily Work Log **31**

A.1 Internship Details 31

A.2 Daily Work Activities 31

Bibliography **36**

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.

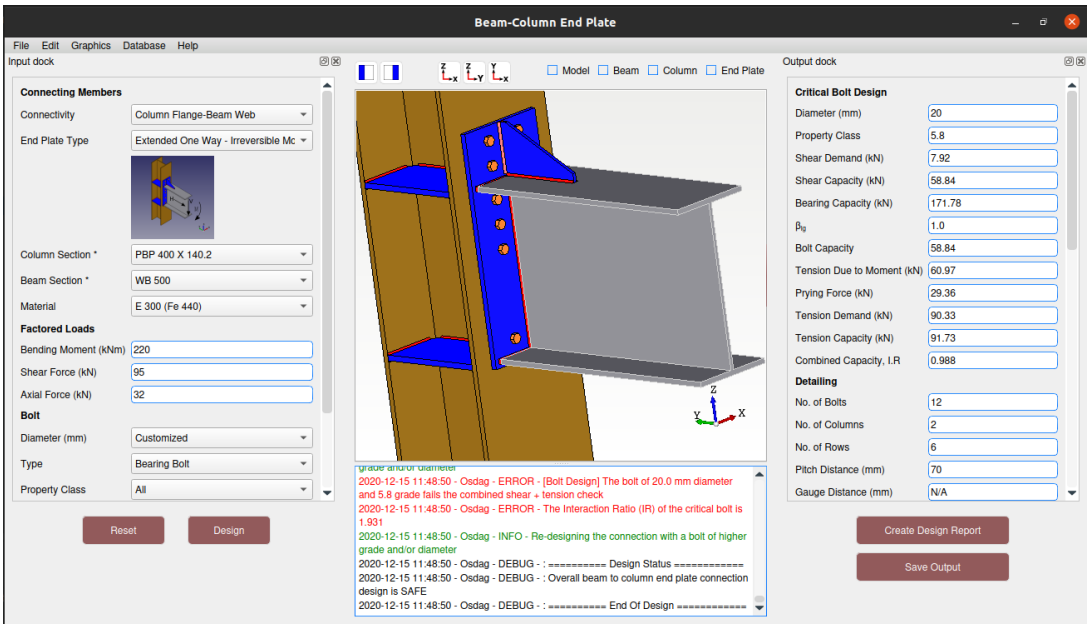

Figure 1.2: Osdag GUI

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

Create the UI of the Osdag web app using React and develop endpoints using Django/REST. Do any of the following

Beam-to-beam splices - end plate Beam to column - end plate Column-to-column splices - cover plate bolted Column-to-column splices - cover plate welded Column to column - end plate

2.2 Tasks Done

Setup and Deployment

- Set up the `Osdag-on-cloud` project in a local development environment.
- Installed and configured necessary dependencies, especially React and associated libraries.
- Used Git for version control to manage the codebase.

Frontend Development for Beam to Beam Splice: End Plate Connection

- Fully implemented the frontend of this module using React.
- Created a UI closely replicating the desktop version of Osdag.

- Developed key components:
  - **Input Dock:** Structural form inputs (section, material, loads, etc.)
  - **Visualization Panel:** View the 3D model with toggleable display options.
  - **Output Dock:** Displays design values and details.
  - **Action Buttons:** Includes "Design", "Reset", "Save Output", and "Create Design Report".

Backend Understanding (Not Fully Integrated)

- Studied the existing backend architecture and API documentation.
- Identified necessary endpoints required for integration.
- Ensured frontend compatibility with future backend integration.

Bug Fixes in Another Module

- Fixed issues in the *End Plate of Shear Connection* module.
- Improved UI/UX and maintained application consistency.

Challenges Tackled

- Replicated complex desktop UI in a web environment.
- Understood structural engineering concepts for input handling.
- Ensured cross-browser consistency and form validation.
- Began integration of 3D visualization features.

Deliverables Produced

- Complete frontend for the Beam to Beam Splice module.
- Fixes and enhancements in the End Plate of Shear Connection module.
- A comprehensive report outlining architecture, implementation, and challenges.

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.

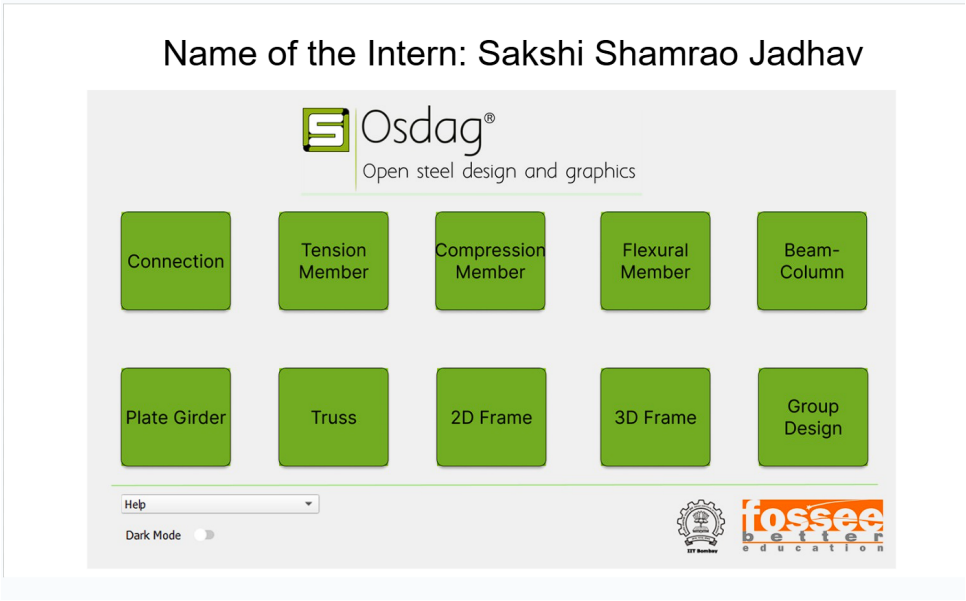

Figure 3.1: Homepage

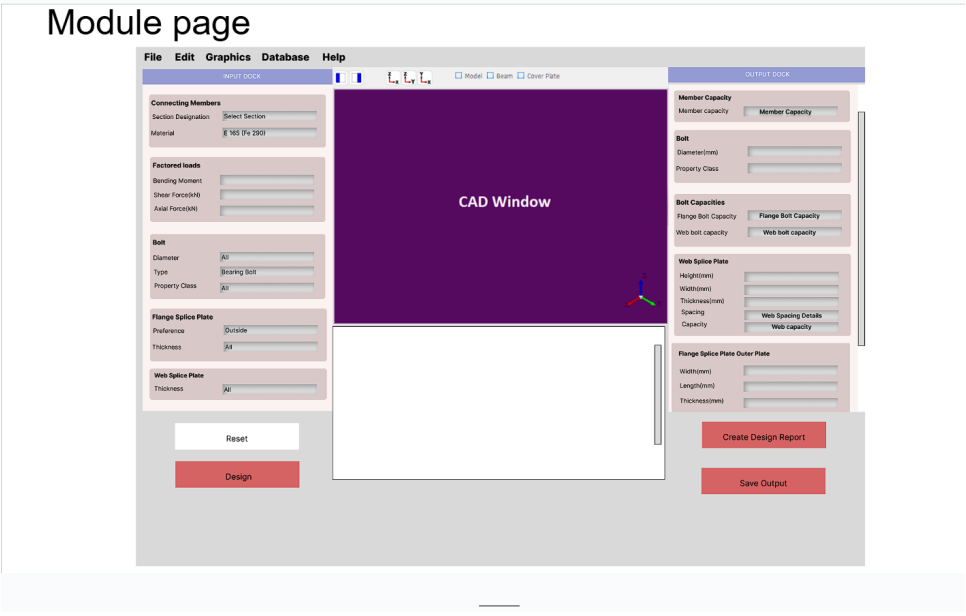

Figure 3.2: Module Page

Chapter 4

Internship Task 2: LCC Website - Frontend

4.1 Task 2: Objective

The task was to develop the frontend of the LCC (Life Cycle Costing) website from scratch using React.

4.2 Task 2: Tasks Completed

Setup and Initial Development

- Successfully set up the LCC web application on the local development environment.
- Developed the landing page of the website.
- Implemented an interactive accordion feature, where clicking on "Input Parameters" expands the corresponding section.

Form Development and Navigation

- Created the following primary forms:
  - `StructureWorksData`
  - `FinancialData`

  - `MaintenanceandRepairData`
  - `BridgeandTraffic`
  - `DemolitionandRecycling`
  - `CarbonEmissionData`
  - `CarbonEmissionCostData`
- Within the `StructureWorksData` module, implemented the following sub-forms:
  - `FoundationForm`
  - `SuperStructureForm`
  - `SubStructureForm`
  - `MiscellaneousForm`
- Implemented seamless form navigation with **Next** and **Back** functionality.
- Added a **popup confirmation** feature for saving form data.

Dynamic Dropdowns and Backend Integration

- Integrated dropdowns with backend APIs to fetch real-time data.
- Implemented logic to dynamically change dropdown values based on the selected material grade.
- In the `CarbonEmissionData` form, ensured that all materials filled in the `StructureWorksData` form are displayed automatically.

4.3 Task 2: Documentation

1. Project Structure (Frontend)

The frontend code is located in:

`src/osbridgelcca/web_app/`

The relevant directory structure is as follows:

```
web_app
|-- App.js                    -> Root React component controlling
   overall structure
|-- Tutorial.js               -> Interactive tutorial section for
   user guidance
|-- src/
    |-- components/
        |-- ProjectDetails/  -> Main container managing project-
           specific forms
        |   |-- ProjectDetails.js
        |   |-- StructureWorksData/
        |   |   |-- FoundationForm.js
        |   |   |-- SuperStructureForm.js
        |   |   |-- SubStructureForm.js
        |   |   |-- MiscellaneousForm.js
        |   |-- FinancialData.js
        |   |-- MaintenanceandRepairData.js
        |   |-- BridgeandTraffic.js
        |   |-- DemolitionandRecycling.js
        |   |-- CarbonEmissionData.js
        |   |-- CarbonEmissionCostData.js
        |   |-- useFormNavigation.js -> Custom hook for Next/Back
            navigation
        |-- InputForm/
            |-- GeneralInfoForm.js
```

2. Setup and Running the Frontend

1. Navigate to the frontend directory:

```
cd src
cd osbridgelcca
cd web_app
```

2. Start the development server using Vite:

   ```
   npm run dev
   ```

3. Access the application in the browser (default: `http://localhost:5173`).

3. Technologies Used

- **React (Vite-based setup):** for building the component-based UI.
- **React Hooks:** (`useState`, `useEffect`) for state management.
- **Lucide React:** for icons and interactive elements.

4. Component Organization and Functionality

- **Tutorial.js:** Provides an introductory guide to help users understand how to fill forms and use the application.
- **GeneralInfoForm.js:** Located under `InputForm/`, it captures basic project metadata (project name, location, etc.) before the detailed forms are accessed.
- **ProjectDetails.js:** Acts as the main container for all project-specific forms. Includes:
  - Integrated navigation using the custom **useFormNavigation** hook for seamless Next and Back functionality.
  - Popup confirmation to save data before moving to the next form.
  - Dynamic dropdowns that change based on selected material grades.
- **StructureWorksData:** Contains multiple sub-forms for structural details:
  - `FoundationForm.js`
  - `SuperStructureForm.js`
  - `SubStructureForm.js`
  - `MiscellaneousForm.js`
- **Other Forms:**

- `FinancialData.js`
- `MaintenanceandRepairData.js`
- `BridgeandTraffic.js`
- `DemolitionandRecycling.js`
- `CarbonEmissionData.js` (auto-fills material data from StructureWorksData)
- `CarbonEmissionCostData.js`

5. Screenshots

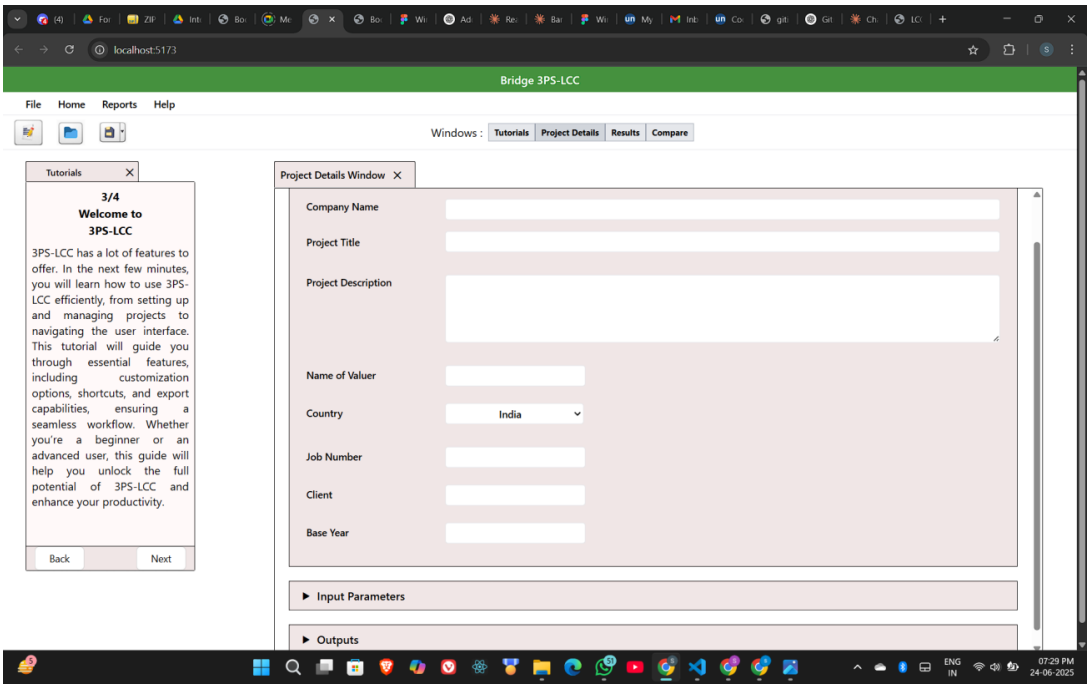

Figure 4.1: Landing page of the LCC website

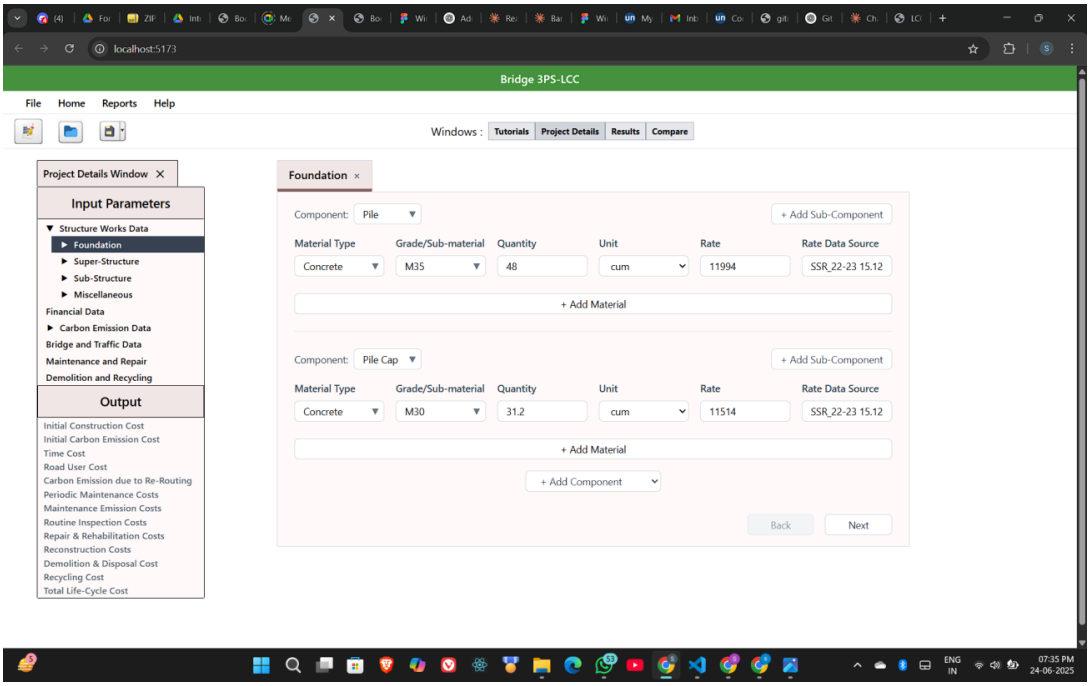

Figure 4.2: Foundation form page

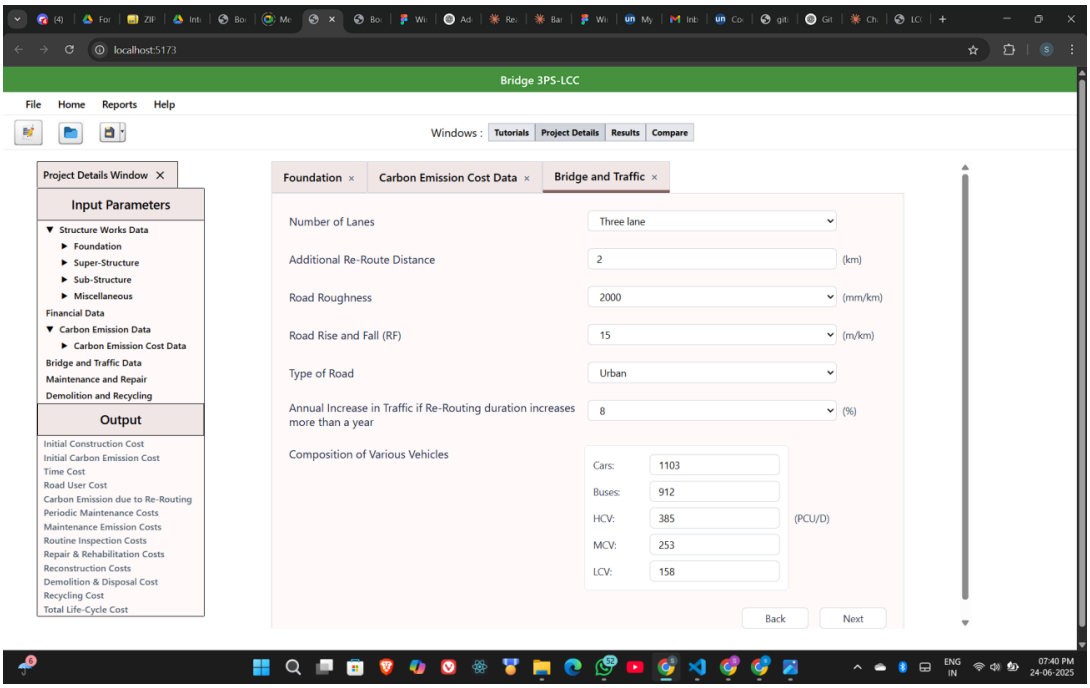

Figure 4.3: Multiple project forms displayed sequentially

Chapter 5

Internship Task 3: LCC Website - Frontend

5.1 Task 3: Objective

The objective of this task was to develop multiple result-oriented graphs for the LCC (Life Cycle Costing) website. These graphs visually represent various cost components calculated from the input forms. Dummy data was used for demonstration, and backend integration was not performed at this stage.

5.2 Task 3: Tasks Completed

Graph Components Developed

The following graph components were created as reusable React components:

- `RadialChartsComponent`
- `SocialCostRadialChart`
- `EnvironmentalCostRadialChart`
- `EconomicCost100YearsRadialChart`
- `SocialCost100YearsRadialChart`
- `EnvironmentalCost100YearsRadialChart`

- `PieChart`
- `BarChart`
- `BubbleChart`
- `HorizontalBarChart`

Each graph uses color-coded cost components, such as:

```
const outputItems = [
  { text: "Initial Construction Cost", color: "#CC9933" },
  { text: "Initial Carbon Emission Cost", color: "#808000" },
  { text: "Time Cost", color: "#CC9933" },
  { text: "Road User Cost", color: "#8B4513" },
  { text: "Carbon Emission due to Re-Routing", color: "#808000" },
  { text: "Periodic Maintenance Costs", color: "#CC9933" },
  { text: "Maintenance Emission Costs", color: "#808000" },
  { text: "Routine Inspection Costs", color: "#CC9933" },
  { text: "Repair & Rehabilitation Costs", color: "#CC9933" },
  { text: "Reconstruction Costs", color: "#CC9933" },
  { text: "Recycling Cost", color: "#CC9933" },
  { text: "Total Life-Cycle Cost", color: "#CC9933" }
];
```

Interactivity and Navigation

- Integrated navigation arrows using **Lucide React icons** (`ChevronLeft`, `ChevronRight`) for switching between graphs.
- Dynamic switching between different cost views (economic, social, environmental) was implemented.
- Responsive design ensures proper scaling of graphs across screen sizes.

Directory Structure (Frontend Relevant)

The graphs are organized inside the React frontend located at:

```
web_app/src/components/
|-- InputForm.jsx
|-- results/
    |-- RadialChartsComponent.js
    |-- SocialCostRadialChart.js
    |-- EnvironmentalCostRadialChart.js
    |-- EconomicCost100YearsRadialChart.js
    |-- SocialCost100YearsRadialChart.js
    |-- EnvironmentalCost100YearsRadialChart.js
    |-- PieChart.js
    |-- BarChart.js
    |-- BubbleChart.js
    |-- HorizontalBarChart.js
```

Technologies Used

- **React (Vite-based setup):** Component-driven graph development.
- **Lucide React:** Icons for navigation between graphs.
- **React Hooks:** (`useState`, `useEffect`) for managing chart data and interactivity.

5.3 Task 3: Output Visuals

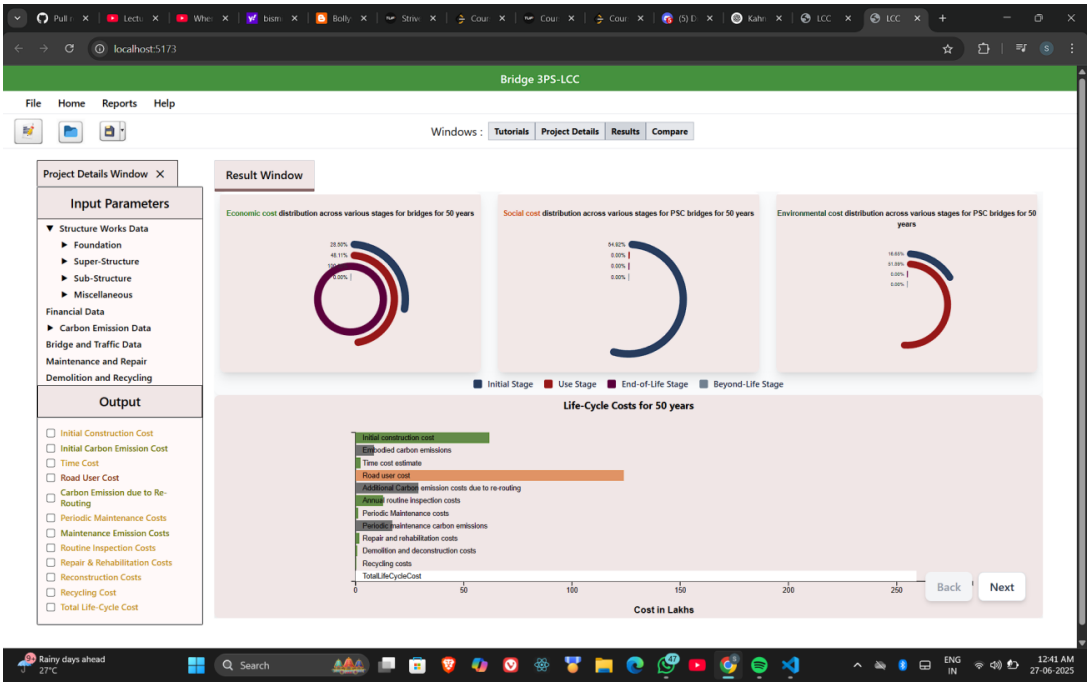

Figure 5.1: Economic, Social, Environmental cost and life cycle cost Visualization for 50 years

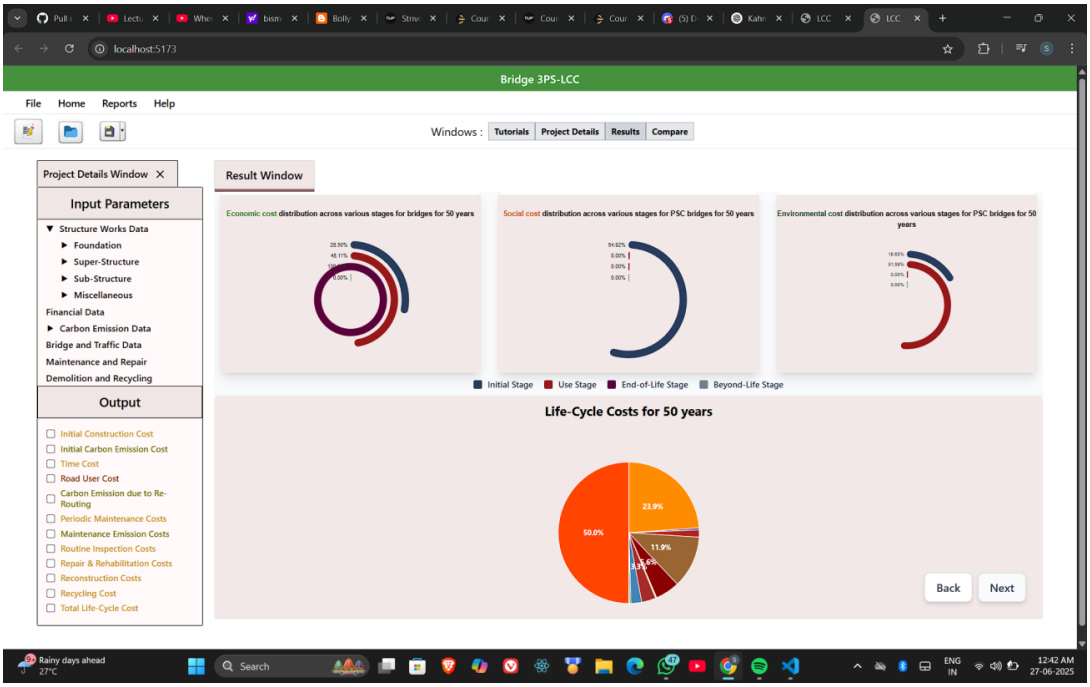

Figure 5.2: Pie chart visualization for 50 years

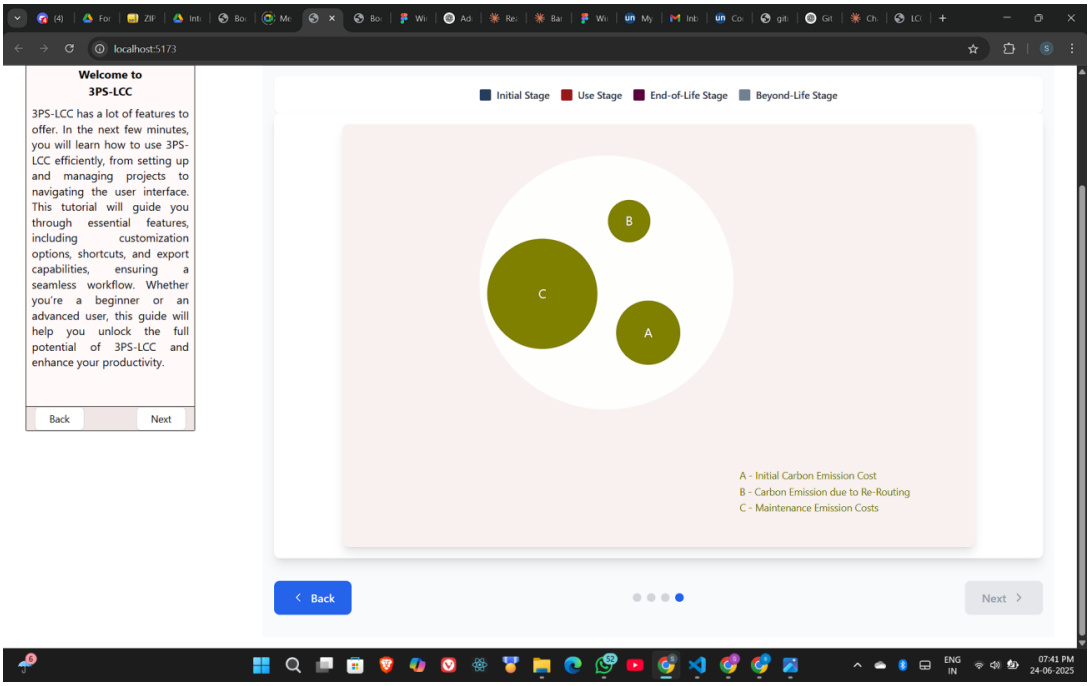

Figure 5.3: Bubble chart Visualization

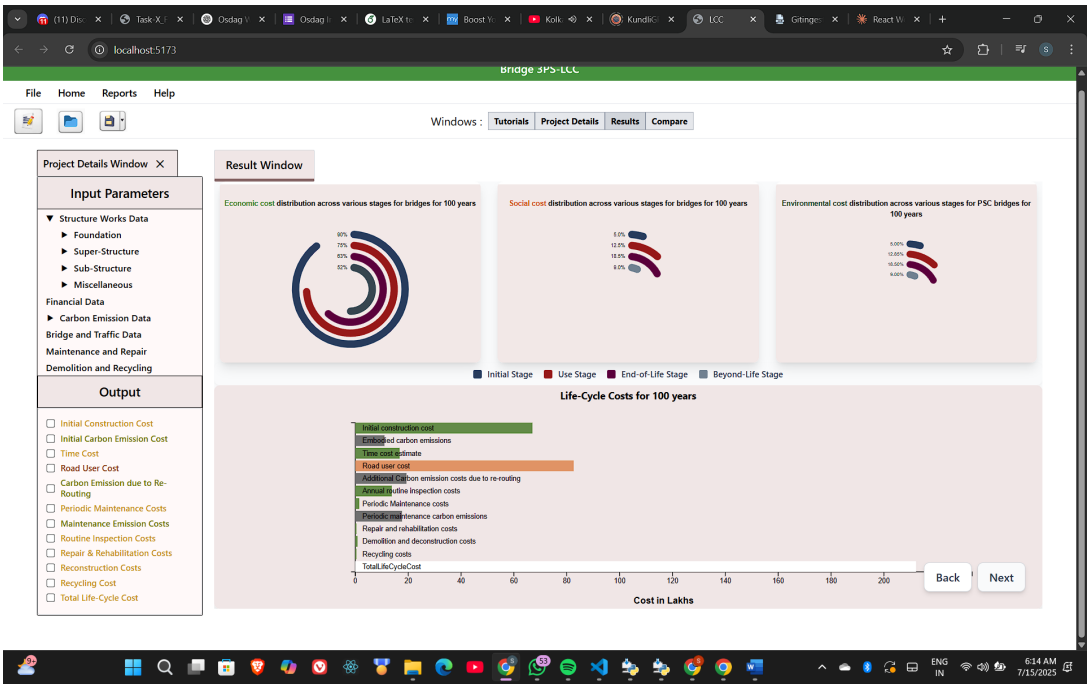

Figure 5.4: Economic, Social, Environmental cost and life cycle cost Visualization for 50 years

Chapter 6

Internship Task 4: LCC Website - Backend-Frontend Integration

6.1 Task 4: Objective

The objective of this task was to achieve full integration between the frontend and backend of the Life Cycle Costing (LCC) website. This involved configuring all API routes, ensuring proper communication between the React-based frontend and the Flask backend, and enabling the forms to send inputs and receive calculated outputs for visualization.

6.2 Task 4: Tasks Completed

1. Backend Route Configuration and Development

All relevant backend routes were registered and tested to ensure smooth data exchange. The routes are organized into the following categories:

- **Form and Data Management Routes:**
  - `/api/forms` – Retrieves all available forms.
  - `/api/components/<form_name>` – Returns components associated with a specific form.
  - `/api/form-data/<form_name>` – Fetches saved form data.
  - `/api/save-form-data/<form_name>` – Saves submitted form data.

- `/api/get-carbon-materials` – Provides the list of materials for carbon emission calculations.

- **Material and Unit Management Routes:**
  - `/api/materials` – Lists all available materials.
  - `/api/material-data/<material_name>` – Retrieves properties for a specific material.
  - `/api/sub-materials/<material_name>` – Returns sub-materials for a selected material.
  - `/api/units/<material_name>` – Lists valid units for the selected material.
  - `/api/materials/<form_name>/<component_name>` – Legacy endpoint for component-wise materials.
  - `/api/sub-materials/<form_name>/<component_name>/<material_name>` – Legacy sub-material endpoint.
  - `/api/units/<form_name>/<component_name>/<material_name>` – Legacy unit endpoint.

- **Cost Calculation Routes:**
  - `/api/calculate-initial-cost` – Computes the initial construction cost.
  - `/api/calculate-initial-construction-cost` – Alternate endpoint for initial construction cost.
  - `/api/calculate-carbon-emission-cost` – Calculates carbon emission-related cost.
  - `/api/calculate-time-cost` – Calculates time-related costs.
  - `/api/calculate-road-user-cost` – Estimates road user cost.
  - `/api/calculate-maintenance-costs` – Processes periodic and routine maintenance costs.
  - `/api/calculate-demolition-recycling-costs` – Estimates demolition and recycling costs.

- `/api/calculate-all-costs` – Aggregates all above costs to produce the total Life-Cycle Cost (LCC).

- **Stored Result Management:**
  - `/api/get-calculation-results/<project_id>` – Retrieves cost calculation results for a specific project.
  - `/api/get-all-stored-results` – Returns all stored calculation results.
  - `/api/clear-calculation-storage` – Clears stored cost calculation data.

- **Debug and Health Check Routes:**
  - `/health` – Verifies that the API is running.
  - `/api/debug/<form_name>` – Debugs form-related issues.
  - `/api/debug/material/<material_name>` – Debugs material-related issues.

4. Frontend Integration

- Connected all forms to their respective endpoints, ensuring real-time cost calculation and material fetching.
- Configured dynamic dropdowns, making form values dependent on material and unit data from the backend.
- Enabled automatic population of fields such as carbon emission materials in `CarbonEmissionData` based on previously filled structural data.
- Ensured smooth CORS configuration, allowing React (Vite server on `localhost:5173`) to interact with Flask (`localhost:5000`).

6.3 Task 4: Documentation

1. Relevant Backend Directory Structure

The backend files directly relevant to this task are:

```
src/osbridgelcca/backend/
|-- config.py                        # Configuration settings
|-- main.py                          # Registers blueprints and
   runs Flask server
|-- api/
    |-- models/                      # Data models
    |   |-- database.py             # Database connection and setup
    |   |-- project.py              # Project data model
    |-- routes/                      # Route definitions
    |   |-- cost_calculations.py    # Handles all cost calculation
       logic
    |   |-- emissions.py            # Fetches carbon emission-
       related data
    |   |-- traffic_analysis.py     # Road user and time cost
       endpoints
    |   |-- user_data.py            # Saves and retrieves form data
    |-- services/                    # Business logic services
        |-- calculation_service.py # Core calculation algorithms
        |-- visualization_service.py # Data visualization
           processing
```

2. Running the Backend

The backend was executed locally as follows:

```
cd C:\Users\asus\Desktop\OsBridgeLCCA\OsBridgeLCCA
$env:PYTHONPATH = "src"
python src\osbridgelcca\backend\main.py
```

The server runs at `http://127.0.0.1:5000`, exposing all the above endpoints.

Chapter 7

Conclusions

7.1 Tasks Accomplished

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

- Developed the complete frontend for the LCC (Life Cycle Costing) website using React, implementing multiple input forms such as:
  - `StructureWorksData` (with sub-forms: FoundationForm, SuperStructureForm, SubStructureForm, MiscellaneousForm)
  - `FinancialData`, `MaintenanceandRepairData`, `BridgeandTraffic`
  - `DemolitionandRecycling`, `CarbonEmissionData`, `CarbonEmissionCostData`
- Integrated seamless form navigation with Next/Back functionality and popup confirmations for saving data.
- Implemented dynamic dropdowns and automatic population of carbon emission-related materials based on earlier form inputs.
- Created multiple cost and emission visualization components, including radial charts, bar charts, pie charts, and bubble charts, to represent both short-term and long-term LCC components.
- Developed and integrated backend routes to process form inputs and calculate life-cycle costs, including endpoints for material data, form data storage, and cost calculations.

- Successfully established frontend-backend integration, enabling real-time data exchange between forms and cost calculation APIs.

7.2 Skills Developed

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

Technical Skills

- Gained hands-on experience with React, including Hooks (`useState`, `useEffect`) and component-based architecture.
- Improved proficiency in designing complex, dynamic forms and implementing user-friendly navigation within a multi-step web application.
- Acquired knowledge of integrating frontend applications with backend APIs using RESTful principles.
- Enhanced understanding of data visualization techniques by creating diverse graphs and charts for cost analysis.
- Strengthened backend development skills in Flask, including defining routes, handling CORS, and managing form data for LCC cost computation.

Professional Skills

- Learned to work on structured, large-scale project repositories while adhering to organized coding practices.
- Improved problem-solving skills by handling real-time data flow between forms, dropdowns, and cost calculations.
- Developed familiarity with collaborative development workflows and documentation best practices.

Overall, the internship provided a comprehensive learning experience by combining frontend development, backend integration, and data visualization within a single project environment.

Chapter A

Daily Work Log

A.1 Internship Details

Name	Sakshi Shamrao Jadhav
Project	OSDAG & LCC Website Development
Fellowship	FOSSEE Summer Fellowship 2025
Duration	May 15, 2025 - July 15, 2025
Type	Full-time internship (6 days/week, Sundays off)

A.2 Daily Work Activities

Date	Day	Task	Hours
15-May-2025	Thursday	Initial setup and installation of OSDAG on development machine	8
16-May-2025	Friday	Continued OSDAG installation and configuration, initial testing	8
17-May-2025	Saturday	Set up LCC website development environment on local system	8
19-May-2025	Monday	Explored OSDAG codebase and began familiarization with project structure	8
20-May-2025	Tuesday	Started working on LCC website frontend development setup	8
21-May-2025	Wednesday	Configured React development environment for LCC website	8
22-May-2025	Thursday	Completed entire project details documentation and structure works components	8
23-May-2025	Friday	Worked on basic project structure and component organization	8
24-May-2025	Saturday	Developed initial UI components for LCC website	8
26-May-2025	Monday	Implemented basic navigation system for the application	8
27-May-2025	Tuesday	Enhanced tab navigation functionality and user interface	8
28-May-2025	Wednesday	Refined navigation system and improved user experience	8
29-May-2025	Thursday	Made significant UI improvements and enhanced visual design	8
30-May-2025	Friday	Continued UI enhancement work and component styling	8
31-May-2025	Saturday	Finalized UI improvements and conducted testing	8

Date	Day	Task	Hours
02-Jun-2025	Monday	Focused on frontend development and component integration	8
03-Jun-2025	Tuesday	Implemented major UI changes and improved user interface	8
04-Jun-2025	Wednesday	Continued frontend development and component refinement	8
05-Jun-2025	Thursday	Advanced frontend development and feature implementation	8
06-Jun-2025	Friday	Completed backend integration setup and initial API connections	8
07-Jun-2025	Saturday	Worked on backend API development and database integration	8
09-Jun-2025	Monday	Continued backend development and API endpoint creation	8
10-Jun-2025	Tuesday	Advanced backend development and service implementation	8
11-Jun-2025	Wednesday	Refined backend services and improved API functionality	8
12-Jun-2025	Thursday	Completed backend integration and tested API endpoints	8
13-Jun-2025	Friday	Finalized backend development and conducted thorough testing	8
14-Jun-2025	Saturday	Implemented initial cost backend calculations and data processing	8
16-Jun-2025	Monday	Developed financial data backend systems and database connections	8
17-Jun-2025	Tuesday	Continued financial backend development and testing	8
18-Jun-2025	Wednesday	Enhanced financial data processing and validation	8

Date	Day	Task	Hours
19-Jun-2025	Thursday	Completed financial backend integration and testing	8
20-Jun-2025	Friday	Worked on advanced backend features and optimization	8
21-Jun-2025	Saturday	Finalized backend development and conducted comprehensive testing	8
23-Jun-2025	Monday	Started working on graph visualization components	8
24-Jun-2025	Tuesday	Implemented various graph changes and visual improvements	8
25-Jun-2025	Wednesday	Continued graph development and data visualization	8
26-Jun-2025	Thursday	Fixed dropdown backend issues and enhanced graphs UI	8
27-Jun-2025	Friday	Finalized graph improvements and user interface enhancements	8
28-Jun-2025	Saturday	Completed graph visualization system and conducted testing	8
30-Jun-2025	Monday	Worked on advanced features and system optimization	8
01-Jul-2025	Tuesday	Continued system optimization and performance improvements	8
02-Jul-2025	Wednesday	Enhanced system performance and conducted thorough testing	8
03-Jul-2025	Thursday	Worked on system refinements and bug fixes	8
04-Jul-2025	Friday	Completed system optimization and performance tuning	8
05-Jul-2025	Saturday	Finalized system improvements and conducted final testing	8

Date	Day	Task	Hours
07-Jul-2025	Monday	Started backend integrations with frontend for all forms	8
08-Jul-2025	Tuesday	Continued backend-frontend integration work	8
09-Jul-2025	Wednesday	Advanced backend integration and form connectivity	8
10-Jul-2025	Thursday	Completed backend integrations for all forms except carbon emission	8
11-Jul-2025	Friday	Integrated carbon emission functionality with form.js	8
12-Jul-2025	Saturday	Finalized all backend integrations and conducted comprehensive testing	8
14-Jul-2025	Monday	Final system testing, bug fixes, and documentation	8
15-Jul-2025	Tuesday	Final project review, documentation completion, and internship wrap-up	8

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.