VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI - 590 018

DBMS MINI PROJECT REPORT

on

“CARBON COUNT”

Submitted by

Aditi S Naik	**4SF22CS009**
Evita Sharon Barboza	**4SF22CS063**

In partial fulfillment of the requirements for the V semester

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Mrs. Vidya V V

Mr. Harisha

Department of CS&E

at

SAHYADRI

College of Engineering & Management

(An Autonomous Institution)

MANGALURU

2024 - 25

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
MANGALURU

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the DBMS mini project work entitled **"Carbon Count"** has been carried out by **Aditi S Naik (4SF22CS009), Evita Sharon Barboza (4SF22CS063),** the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment of the requirements for the V semester of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi during the year 2024 - 25. It is certified that all suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Lab Incharge **Mrs. Vidya V V** Assistant Professor Dept. of CSE	**Lab Incharge** **Mr. Harisha** Assistant Professor Dept. of CSE	**HOD** **Dr. Mustafa Basthikodi** Professor & Head Dept. of CSE

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
MANGALURU

Department of Computer Science & Engineering

DECLARATION

We hereby declare that the entire work embodied in this DBMS mini Project Report titled **“Carbon Count”** has been carried out by us at Sahyadri College of Engineering & Management, Mangaluru under the supervision of **Mrs. Vidya V V** and **Mr. Harisha,** in partial fulfillment of the requirements for the V semester of **Bachelor of Engineering** in **Computer Science and Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Aditi S Naik (4SF22CS009)
Evita Barboza (4SF22CS063)
Dept. of CSE, SCEM, Mangaluru

Abstract

This project aims to develop an innovative website or application that measures the carbon footprints of buildings and other structures with precision. By leveraging a dynamic and comprehensive database, the platform assesses the environmental impact of various construction projects. It incorporates custom values specific to building types, materials, and features—such as flooring, fittings, and more—ensuring accurate emissions assessments tailored to diverse scenarios. This approach allows for a highly detailed understanding of the carbon footprint across various building designs.

The initiative emphasizes ecological conservation by promoting sustainable practices in the construction and design industries. The platform seeks to raise awareness about the environmental impact of building practices, encouraging informed decision-making among architects, engineers, developers, and individuals. By bridging the gap between data-driven insights and actionable solutions, the platform empowers stakeholders to adopt eco-friendly strategies. Ultimately, this project aims to contribute to environmental protection, positioning itself as a pioneering tool in sustainability and aligning construction with global ecological goals for a greener future.

Table of Contents

CHAPTER 1
INTRODUCTION

1.1 Database Management System

Database Management Systems (DBMS) are software applications designed to efficiently store, organize, retrieve, and manipulate structured data. They serve as a fundamental tool for managing large volumes of data across various industries, ensuring data integrity, security, and accessibility. Key features of DBMS include data modeling, query languages like SQL, transaction management, and concurrency control. By centralizing data storage and eliminating redundancy, DBMS offer benefits such as improved data security, scalability, and performance. In summary, DBMS play a vital role in modern information management by providing a reliable framework for handling data,facilitating data-driven decision-making, and enabling innovative applications and services

1.2 MySQL

MySQL is widely used in web development, powering applications such as content management systems, e-commerce platforms, and social networking sites. Its support for ACID (Atomicity, Consistency, Isolation, Durability) properties ensures reliable transaction processing, making it suitable for both small-scale and enterprise-level applications. MySQL's scalability allows it to handle large volumes of data, while its strong community support provides extensive resources and plugins. Additionally, MySQL integrates easily with various programming languages like PHP, Python, and Java, enhancing its versatility and making it a popular choice among developers worldwide.

1.3 Project Overview

Our project, **Carbon Count**, also known as the **Carbon Emission Calculator**, aims to develop a website or application that accurately measures the carbon footprints of various structures, including buildings, houses, and other construction types. By leveraging a comprehensive database, the platform will evaluate the environmental impact of different types of structures based on customizable parameters such as materials, flooring, fittings, and other features.

This initiative is focused on promoting ecological conservation and sustainability within the construction and design industries. The Carbon Count platform will provide detailed emissions assessments, raising awareness about the environmental consequences of various construction choices. By encouraging efforts to reduce carbon footprints, it aims to foster sustainable practices in the construction industry. Ultimately, the project seeks to contribute to environmental protection by empowering individuals and organizations with the tools and knowledge needed to make more eco-friendly decisions. In doing so, it positions itself as a leading solution for sustainability across all types of infrastructure, supporting a greener and more responsible future for the built environment.

CHAPTER 2
LITERATURE SURVEY

This thesis focuses on migrating an Excel-based application for analyzing energy consumption and CO2 emissions to a web-based platform using the spiral lifecycle model and a LAMP stack (Linux, Apache, PHP, MySQL). This approach addresses deployment and maintenance challenges while offering a more user-friendly interface tailored to company needs [1]. The paper emphasizes MySQL's role in database design, comparing it to proprietary systems like Oracle and MS SQL Server, and presents two cases: forward-engineering and reverse-engineering databases with MySQL Workbench and Community Server [2].

The study also revises the Ecological Footprint methodology by updating the Average Forest Carbon Sequestration (AFCS) factor. This updated AFCS calculation includes sequestration rates for various forest types and emissions from wildfires, soil, and harvested wood products, highlighting the need for sustainable resource management [3]. Another study evaluates online carbon footprint calculators for individuals, creating a Feature Index based on common features and applying it to 31 calculators. The study suggests design improvements based on user feedback and performance evaluations to enhance user engagement and climate awareness [4].

A study on China's carbon emissions trading market proposes a data management system using multi-source data fusion. The system shows significant improvements in data transmission capacity, achieving a UOS value of 91.23%, which enhances market efficiency and supports low-carbon transformation [5]. By proposing a carbon emissions data management system based on multi-source data fusion, the research demonstrates significant improvements in data transmission capacity, with the system achieving a UOS value of 91.23% in experiments, highlighting its potential to enhance market operations and support low-carbon transformation.[6]

Finally, a BIM-LCA-based system is introduced for calculating and analyzing bridge carbon emissions. It simplifies data collection across the bridge's life cycle, providing actionable insights for carbon reduction and supporting sustainable bridge engineering [7].

CHAPTER 3
PROBLEM FORMULATION

3.1 Problem Statement

The construction and design industries significantly contribute to global carbon emissions, yet there is a lack of accessible, precise, and actionable tools to measure and manage the carbon footprint of buildings and structures. Current approaches to emissions assessment are often generic, failing to account for the diverse materials, building types, and design features involved in construction projects. This gap limits the ability of architects, engineers, and developers to make informed decisions and implement sustainable practices, thereby hindering efforts to minimize environmental impact and promote ecological conservation.

3.2 Objective

To develop an innovative website or application that precisely measures the carbon footprint of buildings and structures by leveraging a comprehensive database of custom values for materials, building types, and design features. The platform aims to:

- Offer detailed and scenario-specific emissions data for a wide range of construction projects.
- Enhance understanding of the environmental impact of building practices among stakeholders in the construction and design industries.
- Facilitate data-driven decision-making to promote the adoption of eco-friendly materials and methods.
- Align construction practices with global ecological objectives to support a greener and more sustainable future.

CHAPTER 4

METHODOLOGY

4.1 Design

4.1.1 ER Diagram

An Entity-Relationship model describes interrelated things of interest in a specific domain of knowledge. The ER Diagram of our project is shown in the figure given below:

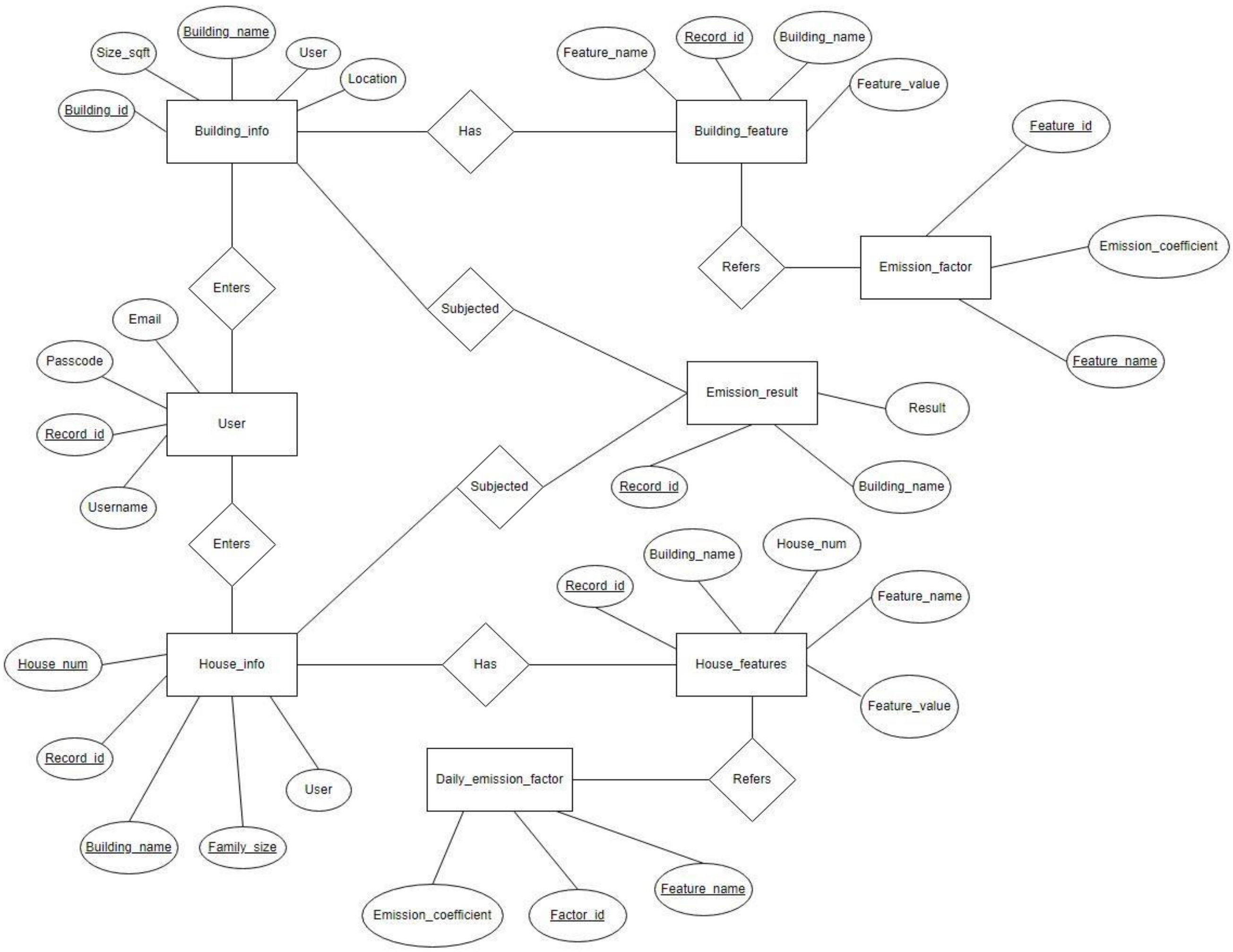

Figure 4.1: ER Diagram of Carbon Count

\

4.1.2 Schema Diagram

A schema diagram is a visual representation of the structure of a database, showing how the data is organized and related within the database. It defines the relationships between different entities, tables, and fields in a database, including constraints and primary/foreign keys. The Schema Diagram of our project is shown in the figure given below:

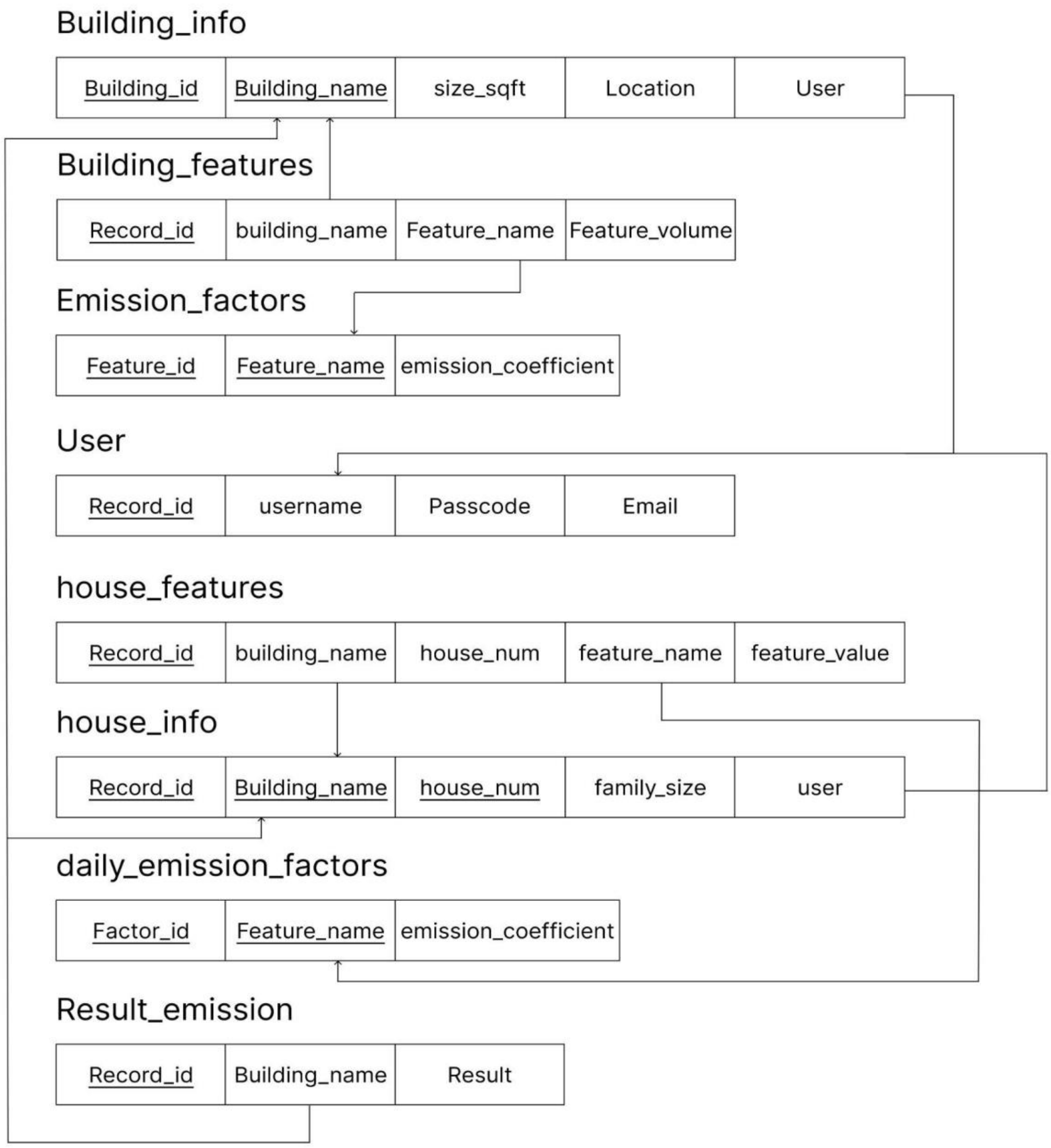

Figure 4.2: Schema Diagram of Carbon Count

4.2 Relational Schema (ER to relational schema)

4.2.1 Mapping of Regular Entity Types

For every entity in our relationship diagram, we have created a separate relation. These created relations contain the respected attributes and respected primary key.

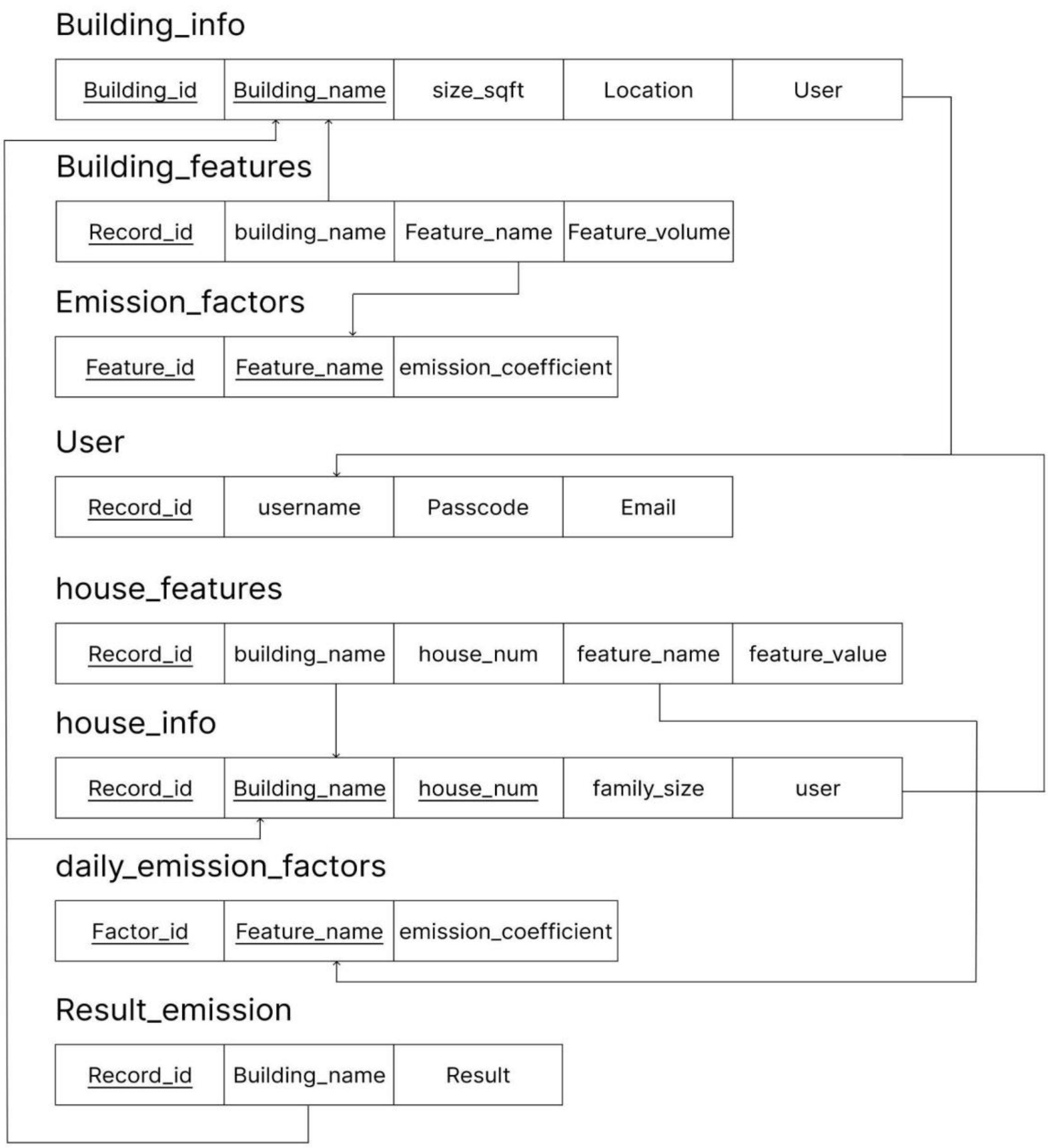

Figure 4.3: Mapping of Regular Entity Type in Carbon Count

4.2.2 Mapping of Weak Entity Types

In the provided database schema, there are no weak entity types present Weak entity types are entities that do not have a primary key attribute of their own and rely on a related strong entity (owner entity) for their existence. They are typically identified by their relationship with the strong entity and the weak entity's discriminator attribute.

4.2.3 Mapping of Binary 1:1 Relationship Types

1. Signup_log - User

Each user can sign up only once and hence there shall be only one log of record_id for each user triggered during signup.The record_id in the user table is referenced as a foreign key in the signup_log entity to establish a 1:1 relation type.

2. House_features - Result_Emission

Summation of House_features of each building from the user can be referenced to only one emission value in result emission for each building.Each Building can have only one emission_value and each record id in Result_Emission can have only one building name.

4.2.4 Mapping of Binary 1:N Relationship Types

1. User - Building_info

An user can have multiple Buildings registered but each building can have only one user recorded per id. User Id is used as a foreign key to reference to user table from building_info

2. User - House_info

An user can have multiple houses registered but each house can have only one user recorded per id. User Id is used as a foreign key to reference to the User table from House_info.

3. Building_Features - Emission_factors

Each building_feature can have only one emission value referenced via emission_id from Emission_factors but each emission_factors can be referenced by multiple buildings from building_features.

4. **House_Features - Daily_Emission_factors**

Each house_feature can have only one emission value referenced via emission_id from house_features but each daily_emission_factors can be referenced by multiple buildings from house_features.

4.2.5 Mapping of Binary M:N Relationship Types

1. **Building_name - Feature_name**

A building can have multiple features, and each feature can belong to multiple buildings.Feature value is an additional attribute that qualifies the relationship between Building_name and Feature_name

4.2.6 Mapping of N-ary Relationship

In the provided database schema, there are no explicit N-ary relationships.

4.2.7 Mapping of Multivalued Attributes

In the provided database schema, there are no explicit Multivalued attributes.

4.3 Normalization

4.3.1 Normalization Definitions

First Normal Form (1NF): In 1NF, each table must have a primary key, and the values in each column must be atomic (indivisible).

Second Normal Form (2NF): In 2NF, the table must be in lNF, and all attributes must be fully functional dependent on the primary key. This means that every non-prime attribute (attributes that are not part of any candidate key) must be dependent on the whole primary key.

Third Normal Form (3NF): In 3NF, the table must be in 2NF, and it must not have any transitive dependencies. This means that no non prime attribute should depend on another non-prime attribute

4.3.2 Normalization Tables

1. **Building_info:**

Building_info

Building_id	Building_name	size_sqft	Location	User

Figure 4.4: Building_info Normalization

1NF: Atomicity: Each column contains indivisible values (no repeating groups or arrays).Each row is uniquely identified (e.g., a primary key exists). Table is 1NF

2NF: The table is in 1NF.There are no **partial dependencies**, meaning no non-prime attribute depends only on part of a composite primary key. The primary key is record_id, which is a single-column key. All other attributes (username, passcode, email) depend solely on record_id. The Table is 2NF.

3NF: The table is in 2NF. There are no **transitive dependencies**, meaning no non-prime attribute depends on another non-prime attribute.

2. Building_Features

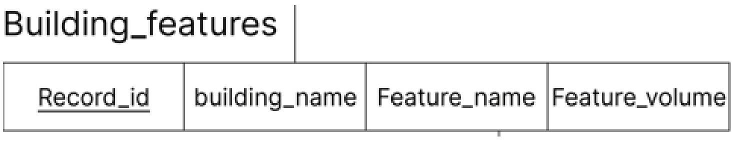

Figure 4.5: Building_features Normalization

1NF: The current table already satisfies 1NF because all values are atomic (no arrays or lists).

2NF: No partial dependencies on Primary Key

3NF: No transitive dependencies on Primary Keys.

3. Emission_factors

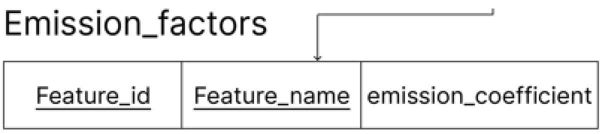

Figure 4.6: Emission_factors Normalization

1NF: The current table already satisfies 1NF because all values are atomic (no arrays or lists).

2NF: No partial dependencies on Primary Key

3NF: No transitive dependencies on Primary Keys.

4. User

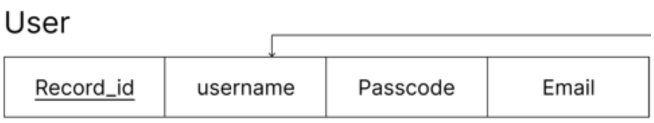

Figure 4.7: User Normalization

1NF: The current table already satisfies 1NF because all values are atomic (no arrays or lists).

2NF: No partial dependencies on Primary Key

3NF: No transitive dependencies on Primary Keys.

5. House_Features

house_features

Record_id	building_name	house_num	feature_name	feature_value

Figure 4.8: House_feature Normalization

1NF: The current table already satisfies 1NF because all values are atomic (no arrays or lists).

2NF: No partial dependencies on Primary Key

3NF: No transitive dependencies on Primary Keys.

6. House_Info

Record_id	Building_name	house_num	family_size	user

Figure 4.9: House_info Normalization

1NF: The current table already satisfies 1NF because all values are atomic (no arrays or lists).

2NF: No partial dependencies on Primary Key

3NF: No transitive dependencies on Primary Keys.

7. Daily_House_Emission

Factor_id	Feature_name	emission_coefficient

Figure 4.10: Daily_house_emission Normalization

1NF: The current table already satisfies 1NF because all values are atomic (no arrays or lists).

2NF: No partial dependencies on Primary Key

3NF: No transitive dependencies on Primary Keys.

8. Result_Emission

Record_id	Building_name	Result

Figure 4.11: Result_emission Normalization

1NF: The current table already satisfies 1NF because all values are atomic (no arrays or lists).

2NF: No partial dependencies on Primary Key

3NF: No transitive dependencies on Primary Keys.

4.4 Table Structure

4.4.1 Building_info

CREATE TABLE building_info (

building_id INT AUTO_INCREMENT PRIMARY KEY,

building_name VARCHAR(30) PRIMARY KEY,

size_sqft INT,

location VARCHAR(25),

user VARCHAR(25) NOT NULL

);

```
mysql> desc building_info;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| building_id   | int          | NO   | PRI | NULL    | auto_increment |
| building_name | varchar(50)  | NO   | PRI | NULL    |                |
| size_sqft     | int          | YES  |     | NULL    |                |
| location      | varchar(100) | YES  |     | NULL    |                |
| user          | varchar(20)  | YES  | MUL | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Figure 4.12: Building_info

4.4.2 Building_features

CREATE TABLE building_features (

feature_id INT AUTO_INCREMENT PRIMARY KEY,

building_name VARCHAR(255) NOT NULL,

feature_name VARCHAR(255) NOT NULL,

feature_value VARCHAR(255),

FOREIGN KEY (building_name) REFERENCES building_info(building_name)

);

```
mysql> desc building_features;;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| record_id     | int         | NO   | PRI | NULL    | auto_increment |
| building_name | varchar(50) | YES  | MUL | NULL    |                |
| feature_name  | varchar(50) | YES  | MUL | NULL    |                |
| feature_value | int         | NO   |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Figure 4.13: Building_features

4.4.3 Emission_factors

CREATE TABLE emission_factors (

feature_id INT AUTO_INCREMENT PRIMARY KEY,

feature_name VARCHAR(255) PRIMARY KEY,

emission_coefficient FLOAT

);

```
mysql> desc emission_factors;
+----------------------+-------------+------+-----+---------+----------------+
| Field                | Type        | Null | Key | Default | Extra          |
+----------------------+-------------+------+-----+---------+----------------+
| feature_id           | int         | NO   | PRI | NULL    | auto_increment |
| feature_name         | varchar(50) | NO   | PRI | NULL    |                |
| emission_coefficient | float       | YES  |     | NULL    |                |
+----------------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Figure 4.14: Emission_factors

4.4.4 Users

CREATE TABLE user (

username VARCHAR(255) PRIMARY KEY,

passcode VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL

);

```
mysql> desc user;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| record_id | int          | NO   | PRI | NULL    | auto_increment |
| username  | varchar(20)  | NO   | UNI | NULL    |                |
| passcode  | varchar(255) | NO   |     | NULL    |                |
| email     | varchar(15)  | YES  |     | NULL    |                |
+-----------+--------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Figure 4.15: Users

4.4.5 House_info

CREATE TABLE house_info (

house_id INT AUTO_INCREMENT PRIMARY KEY,

building_name VARCHAR(25) NOT NULL,

house_num VARCHAR(50),

family_size INT,

user VARCHAR(255) NOT NULL,

FOREIGN KEY (building_name) REFERENCES building_info(building_name)

);

```
mysql> desc house_info;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| record_id     | int         | NO   | PRI | NULL    | auto_increment |
| building_name | varchar(20) | NO   | PRI | NULL    |                |
| house_num     | varchar(5)  | NO   | PRI | NULL    |                |
| family_size   | int         | YES  |     | NULL    |                |
| user          | varchar(20) | YES  | MUL | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Figure 4.16: House_info

);

4.4.6 House_features

CREATE TABLE house_features (

feature_id INT AUTO_INCREMENT PRIMARY KEY,

building_name VARCHAR(255) NOT NULL,

house_num VARCHAR(50) NOT NULL,

feature_name VARCHAR(255) NOT NULL,

feature_value VARCHAR(255),

FOREIGN KEY (building_name) REFERENCES building_info(building_name),

FOREIGN KEY (house_num) REFERENCES house_info(house_num)

);

mysql> desc house_features;

Field	Type	Null	Key	Default	Extra
record_id	int	NO	PRI	NULL	auto_increment
building_name	varchar(20)	YES	MUL	NULL	
house_num	varchar(5)	YES		NULL	
feature_name	varchar(20)	YES	MUL	NULL	
feature_value	float	YES		NULL	

5 rows in set (0.00 sec)

Figure 4.17: House_features

4.4.7 Result_emission

CREATE TABLE result_emission (

result_id INT AUTO_INCREMENT PRIMARY KEY,

building_name VARCHAR(255) NOT NULL,

result FLOAT,

FOREIGN KEY (building_name) REFERENCES building_info(building_name)

);

```
mysql> desc result_emission;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| record_id     | int         | NO   | PRI | NULL    | auto_increment |
| building_name | varchar(15) | YES  | MUL | NULL    |                |
| result        | int         | YES  |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Figure 4.18: Result_emission

4.4.8 Daily_.emission_factors

```
CREATE TABLE daily_emission_factors (
    result_id INT AUTO_INCREMENT PRIMARY KEY,
    feature_name VARCHAR(255) NOT NULL,
    emission_coefficient INT
);
```

```
mysql> desc daily_emission_factors;
+----------------------+--------------+------+-----+---------+----------------+
| Field                | Type         | Null | Key | Default | Extra          |
+----------------------+--------------+------+-----+---------+----------------+
| factor_id            | int          | NO   | PRI | NULL    | auto_increment |
| feature_name         | varchar(100) | NO   | PRI | NULL    |                |
| emission_coefficient | float        | NO   |     | NULL    |                |
+----------------------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

Figure 4.19: Daily_emission_factora

4.4.9 Trigger

```
DELIMITER //
CREATE TRIGGER after_user_signup  AFTER INSERT ON user  FOR EACH ROW
BEGIN
   INSERT INTO signup_log (record_id, username, email, signed_up_at)
   VALUES (NEW.record_id, NEW.username, NEW.email, NOW());
END; //   DELIMITER
```

CHAPTER 5

RESULT AND DISCUSSION

5.1 Login and Sign up Page

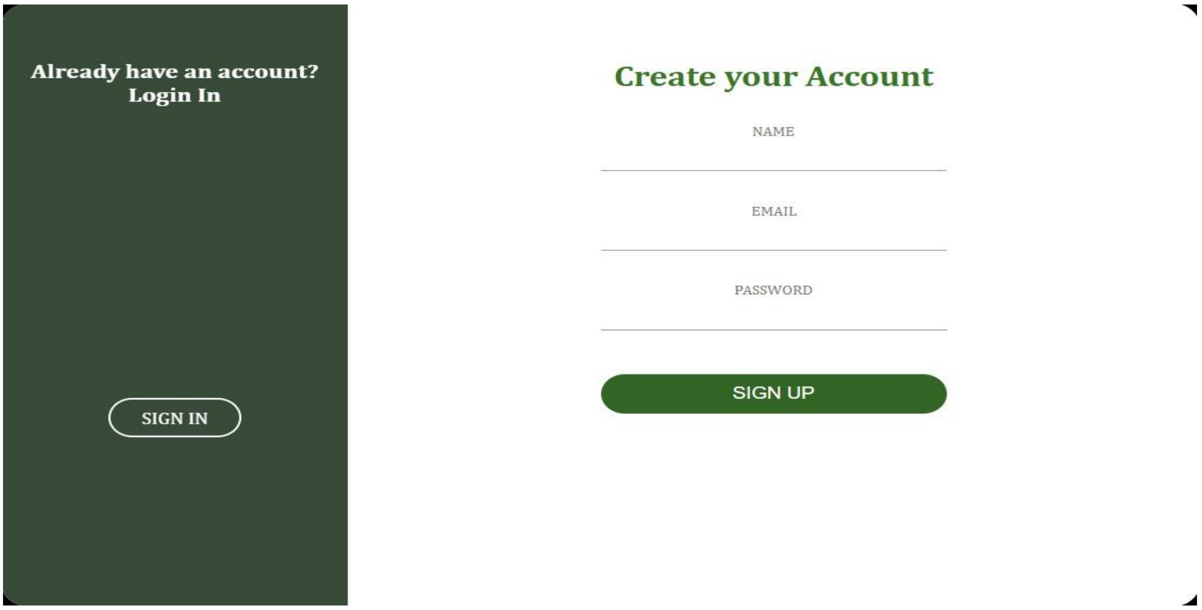

Figure 5.1: Login/Sign Up interface

5.2 Home Page

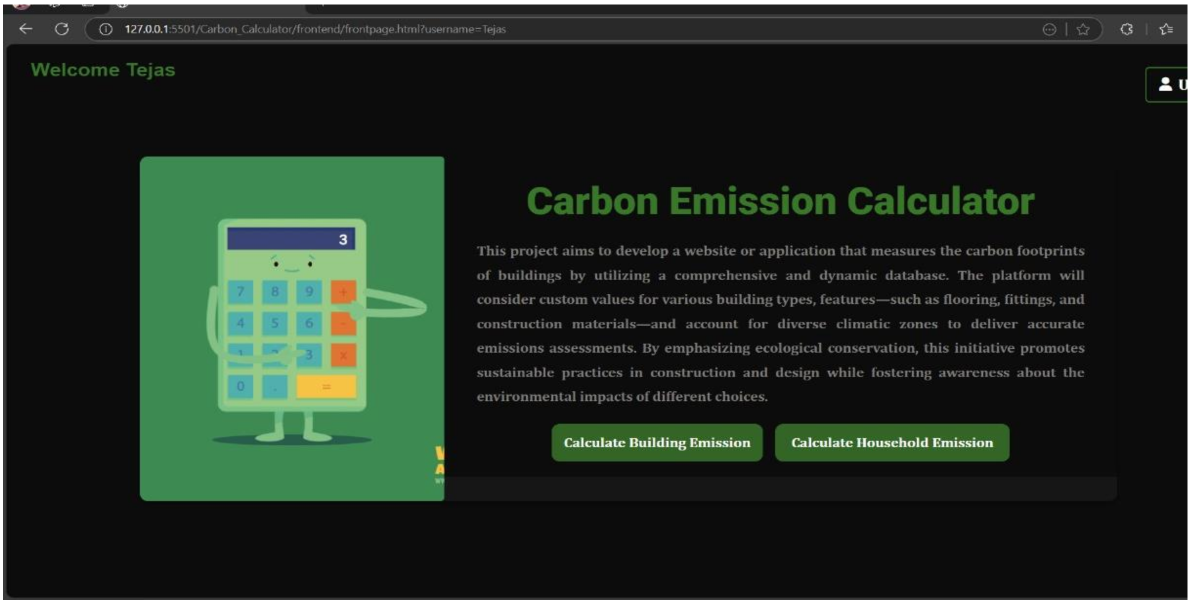

Figure 5.2: Home Page Interface

5.3 User Page

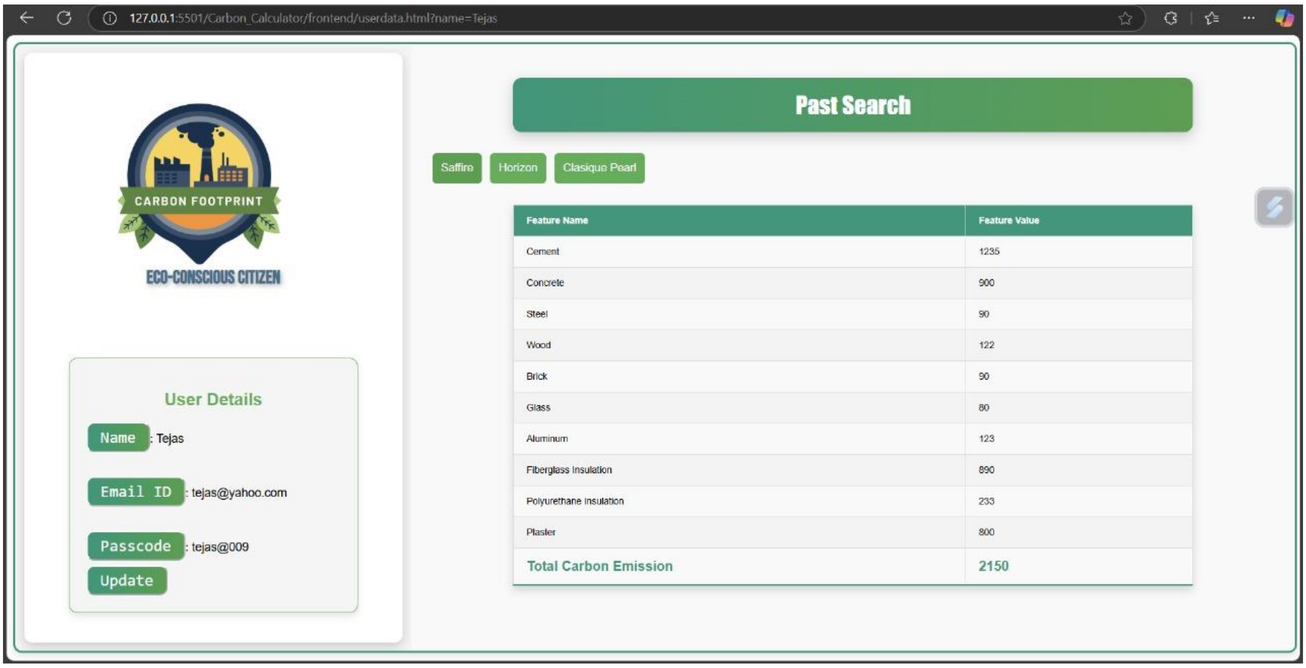

Figure 5.3: User Detail Interface

5.4 Emission Calculation and Update

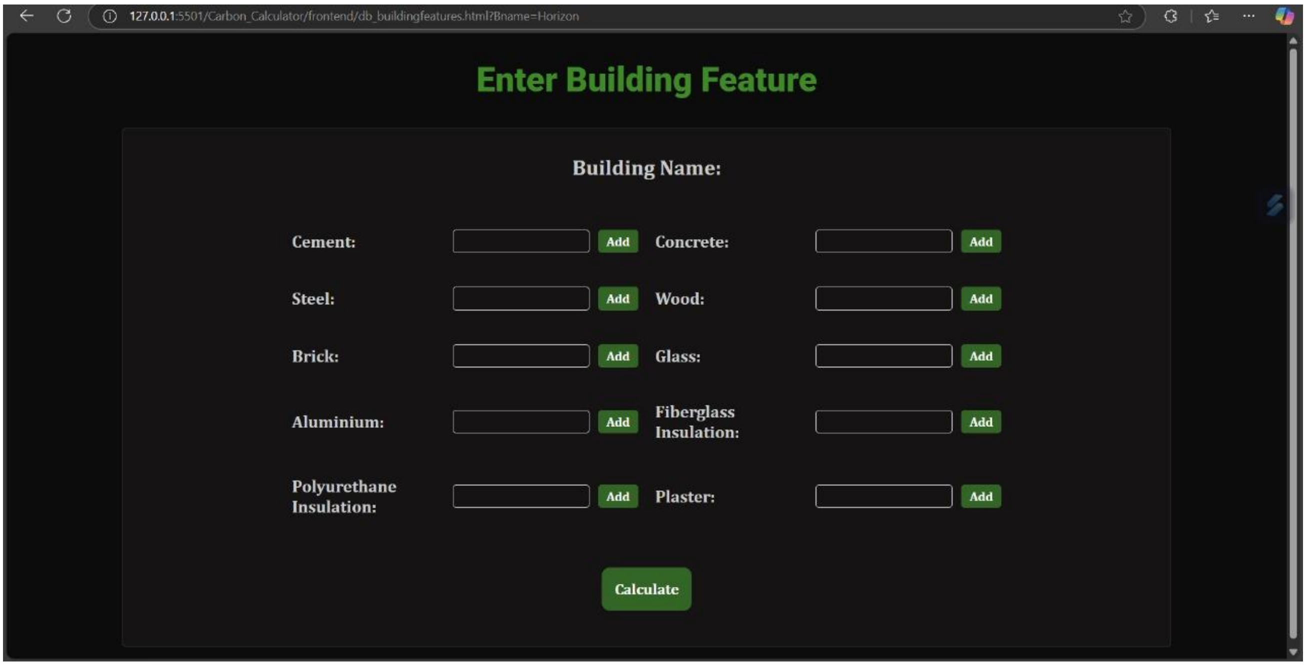

Figure 5.4: Calculation Interface

5.7 Discussion

The platform leverages a dynamic database and customizable inputs for materials, building types, and features to deliver precise and versatile emissions assessments. This approach ensures its applicability to a wide range of construction scenarios, making it a crucial tool in promoting sustainable practices. By providing actionable insights, the platform empowers architects, engineers, and developers to adopt eco-friendly strategies that align with global ecological goals. The integration of accurate data fosters informed decision-making, bridging the gap between awareness and implementation. Future enhancements, such as expanding the database and incorporating real-time data analysis, could further strengthen its impact. This innovation significantly contributes to environmental conservation, paving the way for more sustainable construction practices.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

The proposed platform addresses a critical need within the construction and design industries by offering a precise and dynamic solution for measuring the carbon footprint of buildings and structures. By integrating a comprehensive database tailored to various building types, materials, and features, it ensures accuracy and applicability across diverse scenarios. This innovation not only raises awareness about the environmental impact of construction practices but also empowers stakeholders to make data-driven decisions, fostering a culture of sustainability. The initiative contributes meaningfully to global ecological goals, paving the way for greener and more responsible construction practices.

6.2 Future Scope

- Incorporate AI and machine learning algorithms for predictive analytics, enabling users to simulate and optimize carbon footprint outcomes during the design phase.
- Continuously update and expand the database to include new materials, regional building codes, and innovative sustainable practices.
- Develop mobile and IoT-compatible versions for real-time monitoring and on-site decision-making.
- Create forums and educational resources to encourage knowledge sharing, inspire innovation, and promote sustainability awareness among users.
- Expand the platform’s capabilities to assess the lifecycle carbon footprint of buildings, including maintenance and demolition phases.

By evolving alongside advancements in technology and industry practices, the platform has the potential to become a global benchmark for sustainability in construction.

References

[1]Turan Jr, B. (2009). A Web-Based Database Application as an Analysis Tool for Energy Use and Carbon Dioxide Emission.

[2] Letkowski, J. (2015). Doing database design with MySQL. Journal of Technology Research, 6, 1.

[3] Mancini, M. S., Galli, A., Niccolucci, V., Lin, D., Bastianoni, S., Wackernagel, M., & Marchettini, N. (2016). Ecological footprint: refining the carbon footprint calculation. Ecological indicators, 61, 390-403.

[4] Mulrow, J., Machaj, K., Deanes, J., & Derrible, S. (2019). The state of carbon footprint calculators: An evaluation of calculator design and user interaction features. Sustainable Production and Consumption, 18, 33-40.

[5] Ding, Z., Liu, S., Luo, L., & Liao, L. (2020). A building information modeling-based carbon emission measurement system for prefabricated residential buildings during the materialization phase. Journal of Cleaner Production, 264, 121728.

[6] Lao, W. (2024). Carbon Emission Data Management System Fusion of Multi-Source Data. Procedia Computer Science, 243, 163-171.

[7] Yang, Y., Yue, X., Luo, Y., Jin, L., & Jia, B. (2024). Building Information Modeling–Life Cycle Assessment: A Novel Technology for Rapid Calculation and Analysis System for Life Cycle Carbon Emissions of Bridges. Sustainability (2071-1050), 16(23).