Contents

1. EXECUTIVE SUMMARY

Global warming is the gradual increase in the earth’s temperature due to the greenhouse effect caused by increased levels of carbon dioxide, CFCs, and other pollutants. The pace of warming has significantly increased in the last hundred years due to the burning of fossil fuels such as coal, oil, and natural gas. Global warming has resulted in another issue called climate change which refers to change in weather patterns, sea level rise, more frequent and intense heatwaves etc. Human activities are primarily responsible for global warming, and the serious threat global warming poses to life on Earth calls for environmentally responsible organizations.

As major economic players, corporations have an important role to play, both as developers of new solutions and as users of them. As the effects of climate change have come into focus in recent years, there has been an increasing pressure on organizations to reduce their carbon footprints and address climate change. A growing number of corporates are vocal about their commitment to cutting their carbon footprints. Reporting around sustainability and other environmental issues are increasingly mandated.

The carbon footprint web application aims to help organizations calculate their carbon footprint in various categories and be aware of their actions and helps them take measures to reduce the negative impact.

For the development of this web application, an iterative and incremental process model was followed. As the requirements evolve, this approach allowed for the continuous improvements in order to meet the needs of the users. The front-end technologies such as HTML5, CSS3 and ReactJS were utilized to create an engaging and intuitive user interface and NodeJS, Express.js and MySQL database were used on the back end.

Users of the system are:

- Registered users which are organizations committed to sustainability and responsible corporate citizenship.
- Web app Administrator

2. BACKGROUND

2.1. EXISTING SYSTEM

The approaches to manage their carbon footprint vary depending on the organization and its level of commitment to measuring and reducing carbon emissions. Organizations may rely on manual data collection and calculation with the help of spreadsheets. There are software tools like carbon footprint calculators that give the total footprint based on the inputs provided by the organizations. Other organizations with limited resources or expertise in-house may outsource their carbon footprint assessment to specialized consulting firms.

2.2. DEFINITION OF PROBLEMS

The manual method of carbon footprint calculation is susceptible to human error, such as using incorrect formulas or applying conversion factors inconsistently. Also, it is difficult to identify comprehensive footprint reduction opportunities. It may also be difficult for organizations to measure progress in carbon footprint reduction over time. Without proper tracking, organizations may struggle to evaluate the effectiveness of their sustainability initiatives or identify areas for further improvement. Reporting capabilities in manual processes or standalone tools may be limited, making it challenging to present carbon footprint data in a clear and visually appealing format, which may hinder informed decision-making.

2.3. PROPOSED SYSTEM

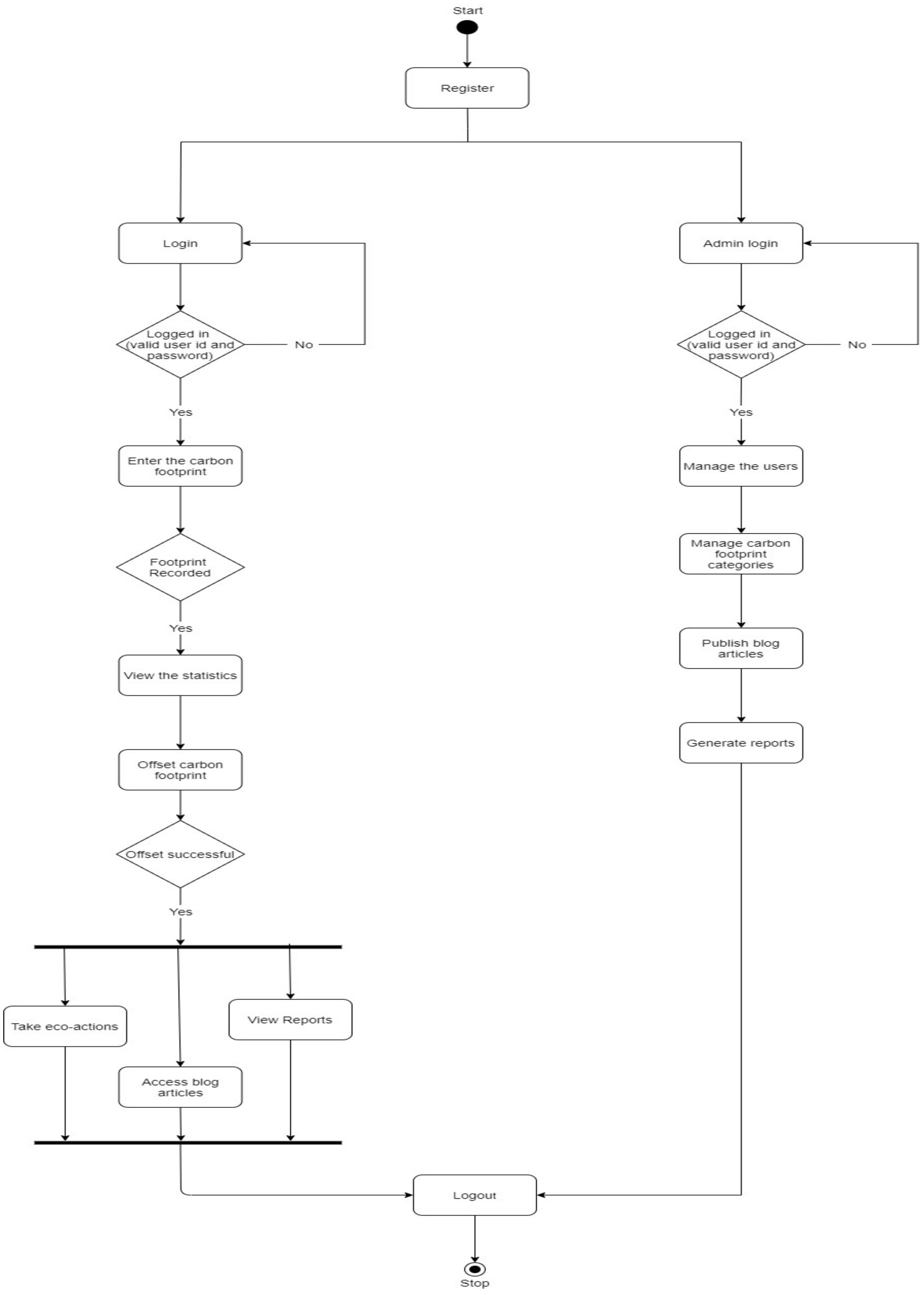
The proposed Carbon Footprint Web Application addresses the inadequacies of current methods of carbon footprint calculation by offering specific solutions:

- It automates calculations, significantly reducing manual effort and increasing operational efficiency.
- The web application is designed to handle large datasets and accommodate growing data needs, ensuring scalability as the organization expands.
- The web application helps track carbon footprint data, allowing organizations to measure progress over time. It offers visualization tools and dashboards to track key performance indicators, identify trends, and evaluate the effectiveness of sustainability initiatives.

- The web application offers robust reporting capabilities, providing customizable reports and visualizations to communicate carbon footprint data effectively, facilitating communication with stakeholders and supporting informed decision-making.
- The web application follows standardized and validated calculation methodologies, ensuring accuracy and consistency in carbon footprint calculations. It utilizes predefined formulas, conversion factors, and emissions factors, reducing the risk of calculation errors and ensuring reliable results.
- The web app can integrate with carbon offset programs or platforms, allowing organizations to participate in carbon offsetting initiatives.
- It also helps organizations identify areas where carbon emissions can be reduced and other sustainable practices.

These solutions enable organizations to make informed decisions, track progress towards carbon reduction goals, and drive sustainability initiatives effectively.

ACTIVITY DIAGRAM



3. PROJECT OVERVIEW

3.1. OBJECTIVE OF THE PROJECT

The Carbon Footprint Web Application is expected to deliver several business benefits to organizations that implement it. Some key business benefits include:

- Enhanced Sustainability Performance:

  The web application helps organizations improve their sustainability performance by accurately measuring, managing, and reducing their carbon footprint. By identifying emission hotspots and implementing targeted reduction strategies, organizations can enhance their environmental credentials and demonstrate their commitment to sustainability.

- Cost Savings:

  The web application enables organizations to identify energy inefficiencies, optimize resource usage, and implement sustainable practices. By reducing energy consumption, waste generation, and carbon emissions, organizations can achieve cost savings through improved operational efficiency and resource management.

- Regulatory Compliance:

  The web application assists organizations in meeting regulatory requirements related to carbon emissions reporting and reduction. By providing accurate and comprehensive carbon footprint calculations and reports, organizations can ensure compliance with environmental regulations and avoid potential penalties.

- Competitive Advantage:

  Implementing a robust carbon footprint management system demonstrates an organization's commitment to environmental stewardship. The web application helps organizations differentiate themselves in the market by showcasing their sustainability initiatives, attracting environmentally conscious customers, and gaining a competitive edge.

- Risk Mitigation:

  By proactively managing carbon footprint and implementing sustainable practices, organizations can reduce their exposure to environmental risks and ensure long-term resilience.

- Innovation and Market Opportunities:

The web application encourages organizations to explore innovative solutions for carbon reduction, leading to the development of new products, services, and business models aligned with sustainability goals. By embracing sustainability as a core value, organizations can tap into emerging market opportunities driven by growing demand for eco-friendly products and services.

The carbon footprint web application brings tangible business benefits by improving sustainability performance, driving cost savings, ensuring regulatory compliance, enhancing competitiveness, strengthening stakeholder relationships, mitigating risks, and unlocking innovation and market opportunities. It empowers organizations to embrace sustainability as a strategic advantage and achieve long-term business success in a carbon-conscious world.

3.2. STAKEHOLDERS

The various stakeholders of this application include:

1. Organizations/Companies:

   Goal: To measure, manage, and reduce their carbon footprint in order to enhance sustainability performance and comply with environmental regulations.

   Benefits: Accurate carbon footprint calculations, identification of emission hotspots, reduction opportunity identification, cost savings through resource optimization, improved operational efficiency, enhanced reputation, stakeholder engagement, and compliance with regulatory requirements.

2. Customers/Consumers:

   Goal: To support environmentally responsible organizations and make informed choices based on sustainability considerations.

   Benefits: Access to transparent information about an organization's carbon footprint, commitment to carbon reduction, and participation in carbon offset initiatives, allowing customers to align their values with their purchasing decisions.

3. Investors:

   Goal: To invest in organizations that demonstrate strong environmental performance and sustainability practices.

   Benefits: Improved transparency and reporting of an organization's carbon footprint, reduction strategies, and participation in carbon offset programs, enabling investors to make informed decisions and support environmentally responsible companies.

4. Regulatory Bodies:

   Goal: To ensure organizations comply with environmental regulations and reduce their carbon emissions.

   Benefits: Accurate and comprehensive reporting of carbon footprint data, compliance with emission reduction targets, and transparency in carbon offset activities, facilitating regulatory oversight and assessment.

5. Environmental Organizations/NGOs:

   Goal: To promote environmental conservation and advocate for sustainable practices.

   Benefits: Collaboration with organizations implementing the Carbon Footprint Web Application to support their sustainability goals, transparency in carbon reduction efforts, and the opportunity to participate in carbon offset initiatives.

6. Supply Chain Partners:

   Goal: To collaborate with environmentally responsible organizations and contribute to sustainable supply chain practices.

   Benefits: Access to carbon footprint data for supply chain analysis, identification of carbon reduction opportunities, collaboration on sustainable initiatives, and alignment with customer expectations for sustainable supply chains.

7. Employees:

   Goal: To work for organizations that prioritize sustainability and contribute to environmental protection.

   Benefits: Increased job satisfaction, engagement, and pride in working for an organization committed to carbon reduction, environmental responsibility, and sustainable practices.

The system empowers stakeholders to actively contribute to environmental protection and sustainable development while driving business success.

3.3. SCOPE OF THE PROJECT

The proposed system is a Carbon Footprint Web Application designed to help organizations measure, manage, and reduce their carbon footprint. The system aims to provide accurate carbon footprint calculations, facilitate carbon reduction strategies, and support carbon offset initiatives. It will be a user-friendly and interactive platform accessible via web browsers.

Key features of the proposed system include:

- User Registration and Authentication:
  Organizations can create accounts and securely access the system using unique credentials. User roles and permissions can be assigned to manage access and data privacy.

- Carbon Footprint Calculation:
  The system will allow input of data related to energy consumption, transportation, waste generation, and other factors contributing to carbon emissions. It will employ algorithms and emission factors to calculate the organization's carbon footprint based on the collected data.

- Data Management and Reporting:
  The system will store and manage the organization's carbon footprint data securely. It will generate comprehensive reports, visualizations, and dashboards to present carbon footprint metrics, trends, and areas for improvement. Customizable reporting options will allow organizations to meet regulatory requirements and demonstrate their sustainability performance.

- Carbon Reduction Strategies:
  The system will provide recommendations and best practices for carbon reduction based on industry standards and benchmarks.

- Carbon Offset Integration:
  The system will offer integration with certified carbon offset providers and projects. Organizations can explore carbon offset options, calculate the offset requirements

based on their carbon footprint, and initiate offset transactions through the system.

Scope Clarifications:

- The installation and setup of the system on servers or cloud infrastructure will not be part of our scope. It will be the responsibility of the organization or their designated IT team.
- Data migration from existing systems, if applicable, will be the responsibility of the organization. We will provide data import/export functionalities to facilitate the process.
- Integration with other systems, such as energy management systems or supply chain management tools, will be considered as separate integration efforts and will be subject to additional analysis and development.

3.4. FEASIBILITY ANALYSIS

3.4.1. TECHNICAL FEASIBILITY

The technical feasibility analysis aims to assess the viability and practicality of developing a carbon footprint web app. The various technical aspects to determine the feasibility of implementing the proposed web app are examined.

- Technology Stack:

  The chosen technology stack for the web app consists of HTML5, CSS3, and React.js (JavaScript library) for the front-end development, and Node.js and Express.js for the back-end. The database management system will be MySQL, ensuring secure and efficient data storage. The selected technology stack is well-established and widely supported, providing a solid foundation for development. It ensures compatibility, scalability, and the ability to incorporate future enhancements.

- Data Management:

  The web app will handle user data, carbon footprint calculations, and user profiles. The data management strategy will involve a well-designed database schema, efficient data querying, and appropriate data encryption techniques to ensure data security and privacy. The selected database management system, MySQL, offers robust data management capabilities and security features. Adequate measures will be taken to ensure data privacy and comply with data protection regulations.

- Integration and APIs:

The web app may integrate with external systems, such as third-party APIs, to provide users with accurate and up-to-date information. The feasibility of integration will depend on the availability and compatibility of the required APIs and the complexity of data synchronization and processing. The availability of APIs and their compatibility with the web app's requirements will be evaluated during the development phase. Alternative integration options or data import mechanisms will be considered if necessary.

- User Experience and Interface:

The web app will prioritize a user-friendly interface and an intuitive user experience. Design considerations will include interactive data input forms, visual representations of carbon footprints, and personalized user dashboards. The chosen technology stack offers extensive capabilities for creating rich and interactive user interfaces. User experience best practices will be followed to ensure a smooth and intuitive app interface.

- Performance and Scalability:

The web app should be capable of handling user loads, concurrent usage, and data processing requirements without compromising performance. The scalability of the chosen technology stack, combined with proper implementation of performance optimization techniques, will ensure the web app's ability to handle increasing user loads and maintain responsiveness.

Based on the assessment of various technical aspects, including technology stack, data management, integration capabilities, user experience, and performance considerations, the proposed carbon footprint web app demonstrates strong technical feasibility.

3.4.2. OPERATIONAL FEASIBILITY

This operational feasibility analysis assesses the practicality and viability of implementing a carbon footprint web app. The focus is on evaluating the operational aspects of the app, including its usability, compatibility with existing systems, resource requirements, and overall impact on the organization.

- User-Friendly Interface:

  The carbon footprint web app will have an intuitive and user-friendly interface, allowing users to easily navigate, input data, and access relevant features.

- Compatibility with Existing Systems:

  The web app will be designed with an open architecture, allowing for integration with existing systems through well-defined APIs or data import/export mechanisms.

- Resource Requirements:

  The operational feasibility analysis will assess the necessary resources, including human resources and infrastructure, for the successful deployment and ongoing maintenance of the web app. Considerations will include the availability of skilled personnel, training requirements, and hardware/software infrastructure needed to support the app's operations.

- Scalability and Performance:

  The web app's scalability and performance will be evaluated to ensure that it can handle increasing user loads and data processing requirements. Performance testing and optimization measures will be implemented to guarantee efficient response times and a seamless user experience, even during peak usage periods.

- Training and Support:

  The operational feasibility analysis will consider the training and support requirements for users of the web app. User documentation will be provided to assist users in navigating and utilizing the app effectively. Adequate training materials and user support mechanisms will be developed to ensure that users can maximize the benefits of the web app.

Based on the operational feasibility assessment, the proposed carbon footprint web app demonstrates strong feasibility in terms of its user-friendly interface, compatibility with existing systems, resource requirements, scalability, performance, and user training and support. These considerations support the successful integration of the web app into the operational processes of the organization, ultimately facilitating effective carbon footprint management and sustainability efforts.

3.4.3. SCHEDULE FEASIBILITY

The schedule feasibility assessment evaluates the timeline and milestones associated with the development and implementation of the carbon footprint web app. It assesses the feasibility of completing the project within the desired timeframe, considering factors such as resource availability, development complexity, and potential risks.

The development process will be divided into key phases, such as requirements gathering, design, development, testing, and deployment. Milestones will be set for each phase to ensure progress tracking and timely completion of deliverables.

Schedule feasibility study for the design is shown below:

Problem identification	5
Requirement analysis	10
Overall design	20
Construction	22
Testing	15

3.4.4. ECONOMIC FEASIBILITY

The economic feasibility report analyzes the financial aspects of developing, implementing, and operating the carbon footprint web app. It assesses the project's profitability and cost-effectiveness, considering factors such as development costs, ongoing maintenance expenses, potential revenue streams, and return on investment.

- Development Costs:

  A comprehensive analysis of the development costs will be conducted to ensure that the investment is financially viable and aligns with the organization's budget and financial goals.

- Operational Costs:

  The ongoing operational costs of maintaining and hosting the web app will be evaluated. This includes expenses related to infrastructure maintenance, software updates, security measures, customer support, and any necessary licensing or subscription fees for third-party services. The operational costs will be carefully assessed to determine the long-term sustainability of the web app and its alignment with the organization's financial resources.

- Revenue Streams:

Potential revenue streams associated with the carbon footprint web app will be explored. This could include options such as subscription fees for premium features or additional services, such as consulting or data analysis, that can be offered to users or partner organizations. Market research and analysis will be conducted to identify potential revenue generation opportunities.

The identification and analysis of revenue streams will determine the economic viability of the web app. This assessment will help determine whether the projected revenue can cover the development and operational costs while generating a desirable return on investment.

- Return on Investment (ROI):

The economic feasibility report will evaluate the projected return on investment for the carbon footprint web app. This analysis will consider the estimated costs, revenue streams, and the expected timeline for recovering the initial investment. It will also account for factors such as market demand, competition, and the potential for growth and expansion.

A thorough ROI analysis will provide insights into the financial viability of the project and determine if the anticipated returns justify the investment of resources.

- Cost-Benefit Analysis:

A cost-benefit analysis will be conducted to weigh the financial benefits against the costs associated with the web app. This analysis will consider both tangible and intangible factors, such as improved sustainability practices, enhanced brand reputation, and potential cost savings resulting from more efficient resource management. The cost-benefit analysis will provide a comprehensive assessment of the economic feasibility by considering the overall benefits derived from the web app compared to the costs incurred.

Based on the economic feasibility assessment, considering development costs, operational expenses, revenue streams, return on investment, and cost-benefit analysis, the carbon footprint web app demonstrates promising economic viability. The revenue generation potential, combined with prudent cost management, suggests that the project can generate a positive return on investment and contribute to the organization's financial objectives.

4. OVERALL PROJECT PLANNING

4.1. DEVELOPMENT ENVIRONMENT

4.1.2. Hardware Specifications

- Processor: Pentium IV or above
- Memory: 4 GB

4.1.3. Software Specifications

- Front End: HTML5, CSS3, ReactJS
- Back End: JavaScript, MySQL (database)
- Web Server: Node.js, Express.js
- IDE: Visual Studio Code
- Operating System: Windows

4.1.3. Others

- UML tool: Draw.io
- Configuration Management tool: Git

4.2. CONSTRAINTS

Constraints in a project refer to the limitations or restrictions that can impact the project's execution and outcomes. The constraints applicable to this project include:

- Time Constraints:
  The project is to be completed within a specific timeframe.

- Data Availability:
  The availability and accessibility of reliable data on carbon emissions from various sources, such as energy consumption, transportation, waste management, or supply chain data, can be a constraint. Limited data availability may affect the accuracy and completeness of the carbon footprint calculations.

- Calculation Methodologies:

Adhering to recognized carbon footprint calculation methodologies and standards, such as the Greenhouse Gas Protocol or ISO 14064, can be a constraint. Ensuring that the app follows the recommended methodologies and accurately calculates emissions can be crucial for credibility and comparability.

- Industry-Specific Factors:

Different industries may have specific factors or emission sources that need to be considered. Adapting the web app to address industry-specific requirements and considering sector-specific emissions factors can be a constraint.

- Localization:

Adapting the web app to different regions, languages, and carbon accounting frameworks may pose constraints. Customizing the app to comply with regional carbon reporting regulations and accounting practices can be necessary.

4.3. DELIVERABLES

The deliverables for the project are listed as follows:

- Requirements Specification
- System Design documents
- Software application
- Complete source code
- DDL script
- Project Management Documentation which includes project schedules

4.4. ASSUMPTIONS AND DEPENDENCIES

Assumptions:

- Data Availability:

It is assumed that the necessary data related to carbon emissions, such as energy consumption, transportation, waste management, etc., is available for the organizations using the web app.

- Accuracy of Data:

The assumption is that the provided data is accurate and reliable for calculating carbon emissions.

- Calculation Methodology:

The web app assumes the use of recognized and accepted carbon footprint calculation methodologies, such as the Greenhouse Gas Protocol, to ensure consistency and credibility in the calculations.

- Compliance:

The web app assumes that the organizations using it are committed to compliance with environmental regulations and reporting requirements.

- User Engagement:

The assumption is that the organizations will actively engage with the web app, input relevant data, and utilize the insights provided to make informed decisions regarding their carbon footprint reduction efforts.

- Scalability:

It is assumed that the web app can handle a growing number of organizations, users, and data volumes as it scales up.

Dependencies:

- Data Sources:

The web app depends on accessing data from various sources, such as energy consumption records, transportation data, waste management data, etc.

- Data Accuracy and Availability:

The availability and accuracy of data depend on the cooperation and information provided by the organizations using the web app.

- Integration with External Systems:

The web app may need to integrate with external systems or databases to fetch relevant data, utility providers, or data providers.

- Internet Connectivity:

The web app relies on a stable internet connection to enable user access and data synchronization.

- Environmental Regulations and Standards:

The web app depends on the adherence to environmental regulations, reporting standards, and methodologies set by relevant authorities or industry bodies.

4.5. RISKS

The list of potential risks that could arise during the execution of the project include:

- Data Availability and Quality:

Risk of insufficient or inaccurate data from organizations for calculating carbon footprints.

Mitigation plan: Establish clear data requirements, provide guidelines for data collection, and conduct data validation checks. Offer support and assistance to organizations in data gathering and verification.

- Technical Challenges

Risks associated with technology, such as compatibility issues, performance bottlenecks, or scalability limitations.

Mitigation plan: Conduct thorough technical feasibility assessments, choose robust and scalable technologies, perform regular system testing, and have contingency plans for addressing technical issues promptly.

- Regulatory Compliance:

Risks related to changes in environmental regulations or reporting standards that impact the functionality or calculations of the web application.

Mitigation plan: Stay updated on relevant regulations, maintain flexibility in the application's design, and have a process in place to adapt to regulatory changes in a timely manner.

- User Adoption and Engagement:

Risk of low user engagement or resistance to using the web application, leading to limited data input and reduced effectiveness.

Mitigation plan: Design a user-friendly interface, provide clear instructions and training materials, offer incentives or recognition for active user participation, and gather feedback to continuously improve the user experience.

- Security and Privacy:

Risks associated with data security breaches or unauthorized access to sensitive information.

Mitigation plan: Implement robust security measures, such as encryption, access controls, and regular security audits. Comply with data privacy regulations, and educate users on best practices for data protection.

- Resource Constraints:

Risks related to limited time, available for the project.

Mitigation plan: Conduct thorough project planning, prioritize critical tasks and regularly monitor and adjust the project timeline and scope as needed.

4.6. PROCESS MODEL

The iterative and incremental process model was chosen to develop this project. This approach involves breaking down the development into smaller iterations or increments, where each iteration builds upon the previous one to deliver a working software increment.

The justifications for its usage in the development of the project is as follows:

- Mitigation of Risk:

The iterative and incremental model helps in identifying and mitigating risks early in the development process. By delivering working increments and gathering feedback, potential issues, such as inaccurate requirements or design flaws, can be identified and addressed promptly, reducing the risk of major setbacks in later stages.

- Flexibility and Adaptability:

This model provides flexibility to accommodate changing requirements and evolving user needs. Each iteration allows for adjustments and refinements based on feedback, ensuring that the final product aligns more closely with stakeholder expectations.

- Early Delivery of Value:

The incremental nature of the model enables the delivery of working software increments early in the development process. This allows users and stakeholders to start benefiting from the software sooner and provides an opportunity for early validation and feedback.

- Continuous Improvement:

With each iteration, lessons learned from previous iterations are incorporated, leading to continuous improvement in the software and the development process itself. The iterative approach encourages learning, adaptation, and the application of best practices throughout the project.

- Better Quality Assurance:

The iterative nature of the model facilitates thorough testing and quality assurance activities. Each iteration includes testing and evaluation of the delivered increment, allowing for the early detection and resolution of defects, leading to improved software quality.

The iterative and incremental process model is particularly suitable when there is a need to adapt to changes and feedback is crucial. It promotes continuous improvement, ultimately leading to a more successful software development outcome.

4.7. TEST STRATEGY

- Unit Testing:

This strategy involves testing individual units or components of the software in isolation to verify their functionality. It focuses on testing small, independent parts of the codebase such as functions.

- Integration Testing:

Integration testing verifies the interactions and integration between different modules or components of the application. It ensures that the individual units work together as expected and that data flows correctly between them. Integration testing can be conducted at different levels, such as module-level, API-level, or database-level, depending on the application architecture.

- System Testing:

System testing evaluates the complete system or application to ensure that it meets the specified requirements. It involves testing the software as a whole, including all integrated modules, components, and external dependencies. System testing verifies the functional and non-functional aspects of the application, such as user interface, performance, security, and reliability.

- User Acceptance Testing (UAT):

UAT involves testing the application from the end-user's perspective to determine if it meets their requirements and expectations. Users or representatives from the target audience perform UAT to validate that the software fulfills its intended purpose and is user-friendly. UAT is typically conducted in a realistic environment that closely resembles the production environment.

5. <u>ITERATION PLANNING</u>

5.1. <u>SCHEDULE</u>

S. NO.	TASK	DURATION
1	Problem identification	5 days
2	Requirement Specification	10 days
3	Database Design and Analysis	12 days
4	Design Analysis	9 days
5	Coding	24 days
6	Testing	10 days
	Total	70 days

5.2. RISKS

The list of potential risks that could arise during the execution of an iteration in the project along with the mitigation strategies are as follows:

- Unclear Requirements:

  Risk of unclear or ambiguous requirements, leading to misunderstandings and potential rework.

  Mitigation plan: Conduct thorough requirements gathering and analysis.

- Scope Creep:

  Risk of uncontrolled expansion of project scope, resulting in increased effort and potential delays.

  Mitigation plan: Clearly communicate project boundaries and manage stakeholders' expectations.

- Resource Constraints:

  Risk of inadequate resources, leading to potential delays or compromised quality.

  Mitigation plan: Conduct resource planning and allocation upfront.

- Technical Challenges:

  Risks associated with complex technical tasks, such as integrating third-party systems or implementing new technologies.

  Mitigation plan: Conduct technical feasibility assessments, allocate sufficient time for research and development.

- Time Constraints:

  Risk of not completing all planned tasks within the allocated time frame.

  Mitigation plan: Conduct realistic time estimation and prioritize tasks based on importance and dependencies. Regularly monitor progress, adjust timelines if necessary, and consider agile methodologies that allow for iterative development and reprioritization.

6. HIGH LEVEL SYSTEM ANALYSIS

6.1. USER CHARACTERISTICS

All users of the system are expected to have basic knowledge of using a computer and basic knowledge in English language.

Users of the system:

- Administrators: Administrators have overall control and management of the carbon footprint web app.
- Organization Managers: These users are responsible for managing carbon footprint data and activities within their respective organizations

6.2. SUMMARY OF SYSTEM FEATURES/ FUNCTIONAL REQUIREMENTS

The main modules are:

6.2.1. Registration Module

New users must register to the site by filling in an online registration form. They must enter details like username, name of organization, email id, location, and password. Once registered, the users can log in to the site by entering their username and password.

6.2.2. Carbon Footprint Calculation Module

The user enters data related to their energy use, transportation, waste and other activities and the carbon footprint for each category is calculated.

6.2.3. Footprint Category Module

The footprint category module classifies the different components that contribute to an organization’s carbon footprint. This module helps users understand and analyze their carbon emissions across various categories, allowing them to prioritize and target specific areas for reduction efforts.

6.2.4. Carbon Offset Module

The carbon offset module integrates with a secure payment processing system that helps users to offset their carbon emissions by supporting environmental projects or initiatives that help reduce greenhouse gas emissions.

6.2.5. Carbon Reduction Module

This module provides users with recommendations for reducing their carbon footprint based on their data inputs. The users can track their actions.

6.2.6. Reports Module

This module generates reports for the user and the admin. The user can view their carbon footprint report which shows their total footprint along with the breakdown, the total carbon offset and total eco-actions committed to. They can also view their offset transaction report.
On the admin's side, the user report shows the total organizations signed up, the average footprint of the users and the total actions committed to. The carbon offset report gives the total green projects supported, total amount donated and the total carbon offset. The admin can also view the transactions per organization and per green project.

6.2.7. Blog Module

The blog module provides users with relevant information, news and insights related to sustainability, carbon footprint reduction and environmental best practices. This module allows the admin to create and publish blog articles.

6.3. NON-FUNCTIONAL REQUIREMENTS/ SUPPLEMENTARY SPECIFICATIONS

The non-functional requirements which define the system performance are:

Performance: The system should be capable of handling many users and data inputs without significant performance degradation. It should provide quick response times and smooth navigation throughout the application.

Security: The system should ensure the confidentiality, integrity, and availability of user data. It should implement appropriate security measures such as encryption, secure authentication mechanisms, and access control to protect sensitive information.

Reliability: The system should be always reliable and available for users, minimizing downtime and interruptions. It should have appropriate backup and recovery mechanisms in place to ensure data integrity and continuity of service.

Usability: The system should have a user-friendly interface that is intuitive and easy to navigate. It should require minimal training for users to understand and operate the various functionalities of the application.
Compliance: The system should comply with relevant environmental regulations, data protection laws, and industry standards. It should facilitate proper data management, privacy protection, and adherence to carbon reporting guidelines.

Integration: The system should have the ability to integrate with other relevant systems or databases, such as third-party carbon data providers. This allows for seamless data exchange and enhances the overall functionality of the application.

Maintainability: The system should be designed and developed in a way that facilitates easy maintenance and future enhancements. It should have well-documented code, modular architecture, and clear separation of concerns to enable efficient troubleshooting, bug fixes, and system updates.

6.4. GLOSSARY

- **Carbon Footprint**: The total amount of greenhouse gas emissions, primarily carbon dioxide (CO2), associated with the activities of an organization. It includes direct emissions from sources owned or controlled by the organization, as well as indirect emissions from the consumption of electricity, transportation, waste management, and other activities.

- **Emission Factors**: Numerical values used to calculate greenhouse gas emissions associated with specific activities or processes. Emission factors represent the average emissions produced per unit of activity, such as the amount of CO2 released per kilowatt-hour of electricity consumed or per mile traveled by a vehicle.

- **Carbon Offset**: The process of compensating for greenhouse gas emissions by investing in projects or activities that reduce or remove an equivalent number of emissions from the atmosphere. Carbon offset projects can include renewable energy generation, afforestation projects, and methane capture initiatives.

- **Carbon Reduction Strategies**: Actions and initiatives implemented by organizations to reduce their carbon footprint. These strategies can involve energy efficiency improvements, adoption of renewable energy sources, waste reduction and recycling programs, transportation optimization, and employee engagement campaigns.

- **Sustainability Performance**: The measurement and evaluation of an organization's efforts to operate in an environmentally responsible and sustainable manner. It considers factors such as energy efficiency, waste management, carbon reduction, social responsibility, and corporate governance.

- **Regulatory Compliance**: Adherence to environmental laws, regulations, and standards related to carbon emissions and sustainability. Organizations must comply with local, regional, and national requirements, report their carbon footprint data, and meet emission reduction targets set by regulatory bodies.

- **Stakeholder Engagement**: The process of involving and communicating with individuals and groups that have an interest or influence in an organization's carbon footprint and sustainability efforts. Stakeholders can include employees, customers, investors, regulatory bodies, environmental organizations, and local communities.

- **Offset Providers**: Certified organizations or projects that offer carbon offset opportunities. These providers ensure that offset projects adhere to recognized standards, such as the Verified Carbon Standard (VCS) or Gold Standard, and facilitate the purchase and retirement of carbon offsets on behalf of organizations.

- **Sustainable Supply Chain**: The integration of environmental considerations into the procurement and management of goods and services across the supply chain. It involves collaborating with suppliers to reduce carbon emissions, improve resource efficiency, promote sustainable practices, and ensure the traceability and sustainability of products.

- **Environmental Impact Assessment**: The evaluation and assessment of the potential environmental consequences of an organization's activities, including the calculation and analysis of the carbon footprint. It helps identify areas of high environmental impact, prioritize mitigation strategies, and monitor the effectiveness of sustainability initiatives.

6.5. BUSINESS RULES

The business rules specific to the user’s domain/organization that need to be satisfied by this system include:

- Emission Calculation Methodology:

The system must adhere to the specific emission calculation methodology or standards adopted by the user's domain/organization. This includes using recognized emission factors, measurement protocols, and calculation methodologies to ensure accurate and consistent carbon footprint calculations.

- Data Privacy and Security:

The system should comply with applicable data privacy laws and regulations to protect the confidentiality of user data. It should have appropriate security measures in place, including user authentication, access controls, encryption, and data backup, to prevent unauthorized access, data breaches, and data loss.

- Reporting and Compliance:

The system should generate comprehensive reports and facilitate compliance with regulatory requirements. It should enable users to generate accurate and timely carbon footprint reports for submission to regulatory bodies, auditors, or other stakeholders. It should also support compliance with carbon reduction targets or reporting obligations set by regulatory frameworks or industry initiatives.

- Carbon Offset Integration:

If carbon offsetting is part of the user's sustainability strategy, the system should integrate with certified carbon offset providers. It should facilitate the calculation of offset requirements based on the organization's carbon footprint and provide options for purchasing and retiring carbon offsets within the system.

- Scalability and Performance:

The system should be capable of handling a growing volume of data as the organization expands its operations or tracks carbon footprint data at a granular level. It should be scalable, reliable, and capable of handling concurrent user requests without significant performance degradation.

- User-Friendly Interface:

The system should provide a user-friendly interface that is intuitive and easy to navigate. It should present information in a clear and organized manner, making it easy for users to input data, generate reports, and access relevant features and functionalities.

- Audit Trail and Data History:

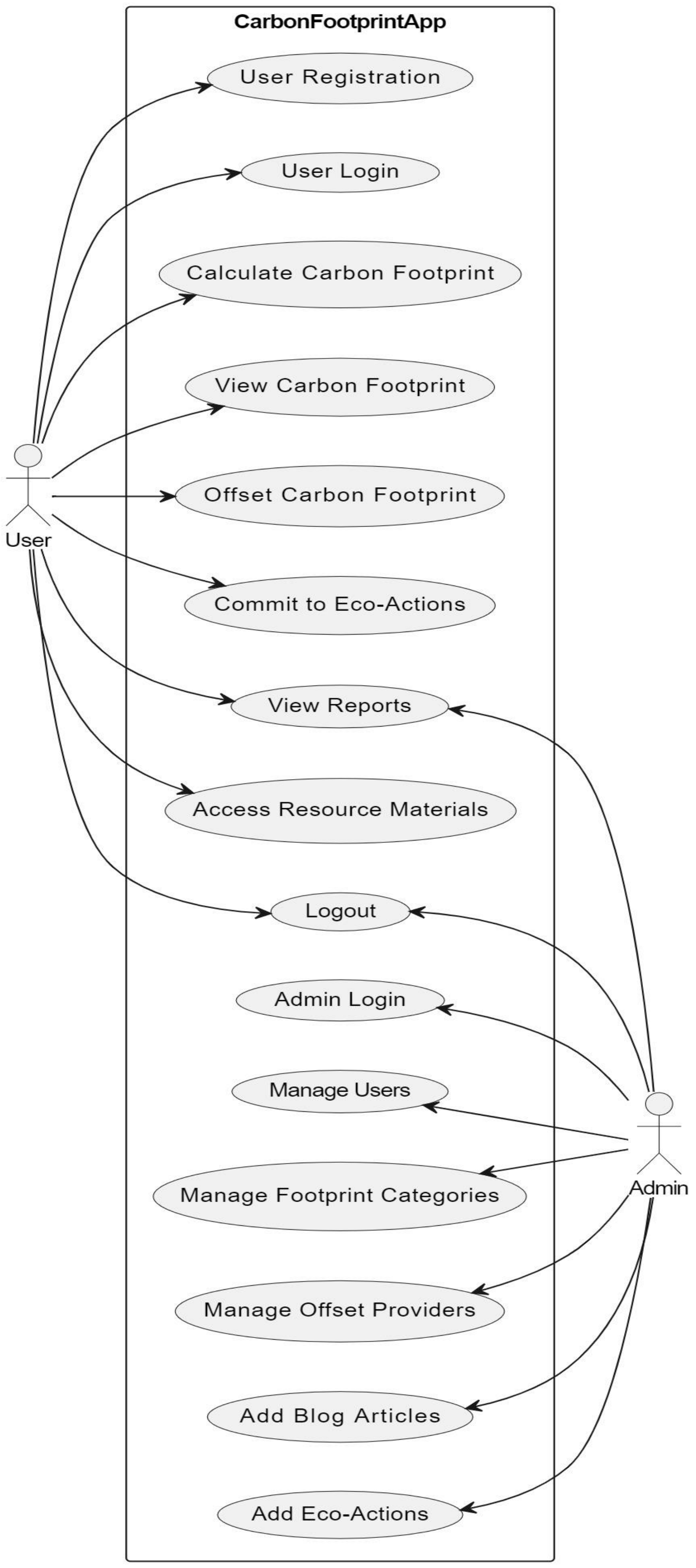
The system should maintain an audit trail or data history, capturing changes, updates, and user activities related to carbon footprint calculations and data management. This helps in tracking and auditing changes made to the system and provides transparency and accountability.

6.6. USE CASES

The use cases for the carbon footprint web app are:

- User Registration: Allows users to create an account by providing necessary information such as name, email address, location, and password.
- User Login: Enables users to log into their accounts using their credentials to access personalized features and data.
- Calculate Carbon Footprint: Allows users to input relevant data such as energy consumption, transportation usage, and waste generation to calculate their carbon footprint.
- Track Carbon Footprint: Provides users with a dashboard or visual representation to track and monitor their carbon footprint over time.
- Carbon Offset Purchasing: Provides users with the option to invest in renewable energy projects to offset their carbon emissions.
- Commit to actions: Provides users with a set of carbon reduction actions which they can commit to.
- Reporting and Analytics: Generates comprehensive reports and analytics on carbon footprint data, including emissions breakdown by category, progress towards sustainability goals, and comparison with industry benchmarks.
- Educational Resources: Provides users with access to educational materials, and resources to enhance their understanding of carbon footprints and sustainable practices.

6.7. USE CASE DIAGRAM



7. USE CASE MODEL

7.1. USE CASE TEXT

The use case text for few of the critical use cases of the web application are as follows:

7.1.1. Use Case: User Login

Description: This use case allows a user to log in to their existing account in the carbon footprint web app.

Actors:

- User

Preconditions:

- The user is accessing the login page of the web app.

Postconditions:

- The user is successfully logged in and gains access to their account dashboard.

Flow of Events:

1. The user navigates to the login page of the web app.
2. The user enters their login credentials such as email address and password.
3. The user submits the login form.
4. The system validates the user's credentials.
5. If the credentials are valid:
   a. The system logs the user in.
   b. The system redirects the user to their account dashboard.
6. If the credentials are invalid:
   a. The system displays an error message indicating incorrect credentials.
   b. The user can retry entering their credentials.

Non-Functional Requirements:

- The login process should be completed within a reasonable time frame (e.g., under 1 minute).
- The system should securely store and handle user login credentials.
-

User Inputs:

- Email Address: Text input (e.g., john.doe@example.com)

- Password: Text input (e.g., Password123)

Error Messages and Information Texts:

- "Wrong email address or password.”

External System Interface Requirements:

- None

7.1.2. Use Case: Calculate Carbon Footprint

Description: This use case allows users to calculate their carbon footprint based on their inputs and activities.

Actors:

- User

Preconditions:

- The user is logged in to their account.

Postconditions:

- The carbon footprint calculation is performed and displayed to the user.

Flow of Events:

1. The user navigates to the carbon footprint calculation page.
2. The user enters information and selects options related to their activities.
3. The user submits the calculation form.
4. The system processes the inputs and performs the carbon footprint calculation.
5. The system displays the calculated carbon footprint to the user, including:
   - Total carbon emissions in CO2 equivalent units.
   - Breakdown of emissions by category (transportation, energy, materials).
6. The user can view and analyze the results.

Non-Functional Requirements:

- The carbon footprint calculation should be accurate and based on reliable data and calculation methodologies.
- The system should handle a wide range of user inputs and activities.
- The calculation process should be efficient and provide real-time results.

User Inputs:

- Transportation: Distance (numeric input), Mode of transportation (dropdown selection), Fuel efficiency (numeric input), etc.

- Energy Consumption: Electricity usage (numeric input).
- Material Usage: Type of industrial activity (dropdown), Material used (dropdown), Amount (numeric input) etc.
- Fuels Combusted: Fuel type (dropdown), Quantity (numeric input), etc.

Error Messages and Information Texts:

- "Please provide valid inputs for all required fields."

External System Interface Requirements:

- None

7.1.3. Use Case: Offset Carbon Emissions

Description: This use case allows users to offset their carbon emissions by participating in carbon offsetting projects.

Actors:

- User
- Carbon Offset Provider

Preconditions:

- The user is logged in to their account.
- The user has identified the amount of carbon emissions they want to offset.

Postconditions:

- The user's carbon emissions are offset through participation in carbon offsetting projects.

Flow of Events:

1. The user navigates to the carbon offset section within the web app.
2. The system presents available carbon offsetting projects.
3. a. The user chooses a specific carbon offsetting project.
   b. The system provides information about the project, its environmental impact, and the associated cost.
   c. The user confirms their participation in the project and proceeds to payment.
4. The user provides the necessary payment information.
5. The system processes the payment and confirms the successful offset of the user's

carbon emissions.
6. The user receives a confirmation message with details about the offsetting process and any associated certificates or documentation.
7. The user can navigate back to other sections or perform additional actions within the web app.

Non-Functional Requirements:

- The carbon offsetting process should be secure and provide a seamless user experience.
- The system should provide clear information about the available carbon offsetting options and their impact.
- The payment process should be reliable and support different payment methods.

User Inputs:

- Selection of specific project

Error Messages and Information Texts:

- "Payment failed. Please check your payment information and try again."
- "Unable to process carbon offsetting request. Please contact customer support."

External System Interface Requirements:

- Integration with a payment gateway for secure payment processing.

7.2. OPERATION CONTRACTS

Here are some operation contracts for the key operations in the carbon footprint web application:

1. **User Registration:**

Operation: RegisterUser(username, password, email, location)

Preconditions:

- The username, password, email, and location provided by the user are valid and meet the specified criteria.

Postconditions:

- A new user account is created with the provided username, password, email, and location.
- The user is successfully registered and can access the application's features.

Error Handling:

- If the username or email is already taken, display an error message indicating that the username or email is already in use.

2. **Calculate Carbon Footprint:**

Operation: CalculateCarbonFootprint(userId, data)

Preconditions:

- The user is logged in with a valid user email.
- The data provided for calculation is valid and complete.

Postconditions:

- The carbon footprint of the user's activities is calculated based on the provided data.
- The calculated carbon footprint value is stored in the user's account for future reference and reporting.

Error Handling:

- If the provided data is incomplete or contains invalid values, display an error message indicating the specific data validation error.

7.3. REPORTS

User Reports:

The user can view reports on total carbon footprint along with the breakdown by category, the total carbon offset, the offset transactions, and the eco-actions taken.

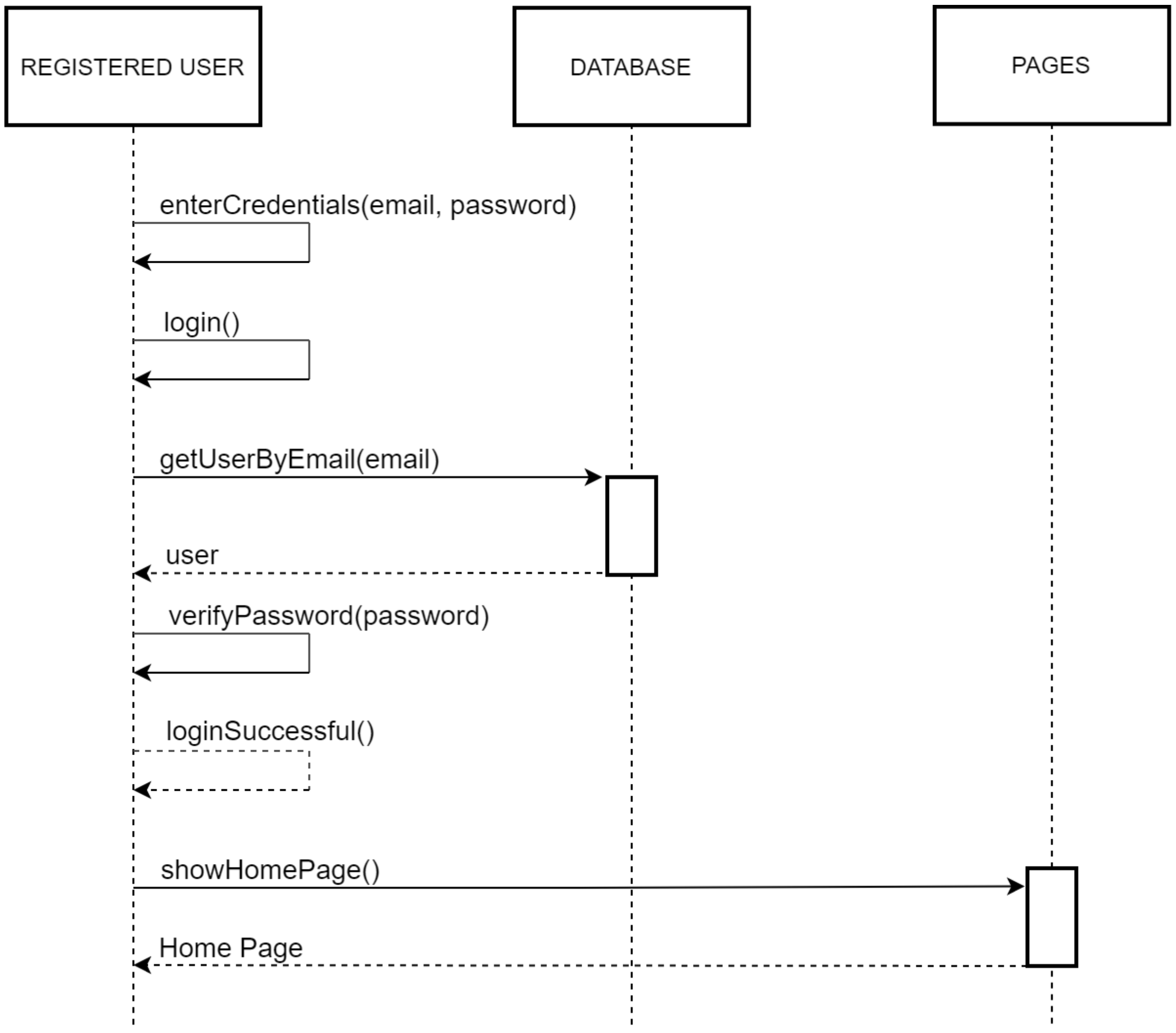
Admin Reports:

The admin can view reports on average carbon footprint of all users, the total offset projects supported along with the total amount raised for the projects and the total carbon offset. The offset transaction reports are generated per user and per project.

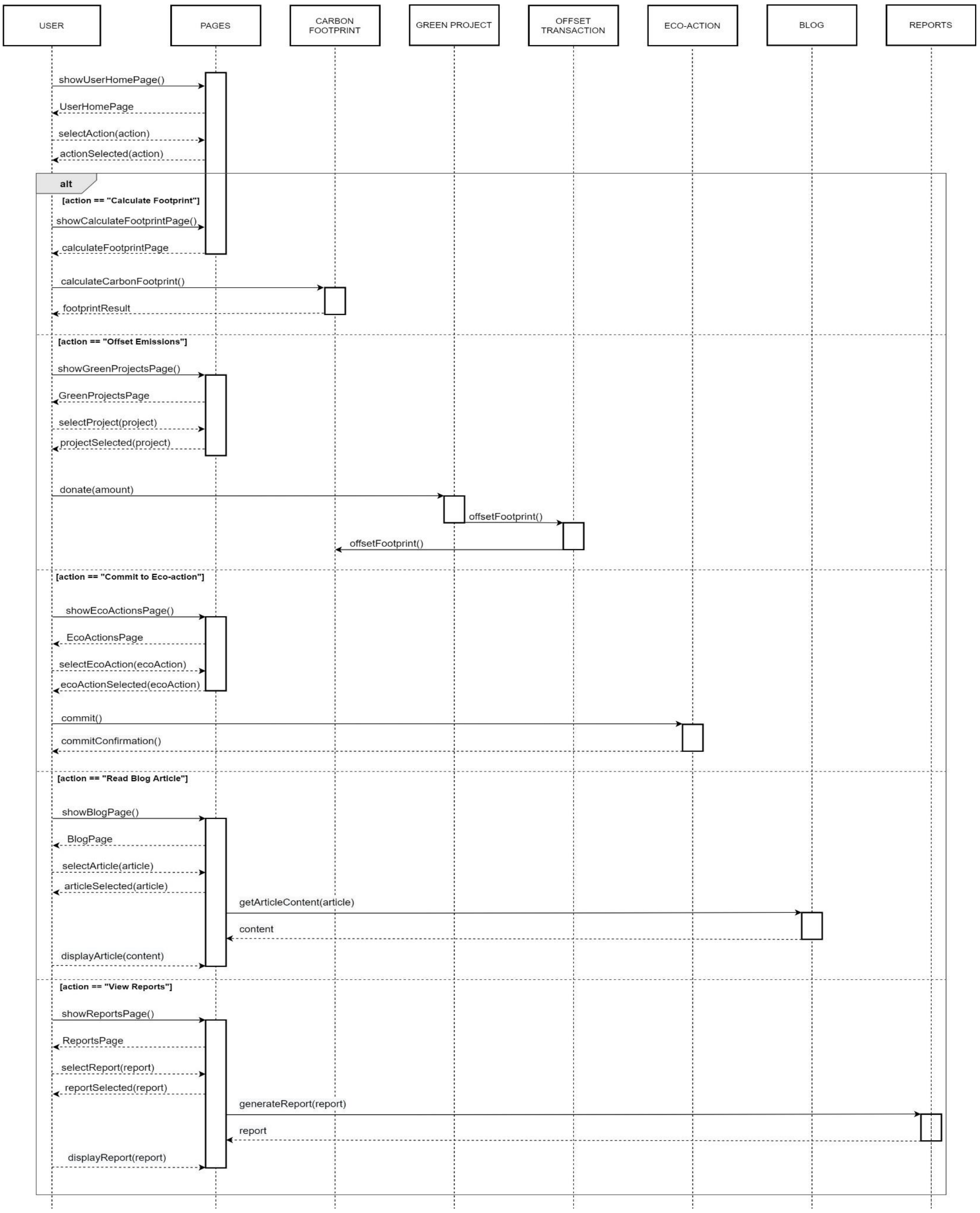
DESIGN MODEL

7.4. SEQUENCE DIAGRAMS

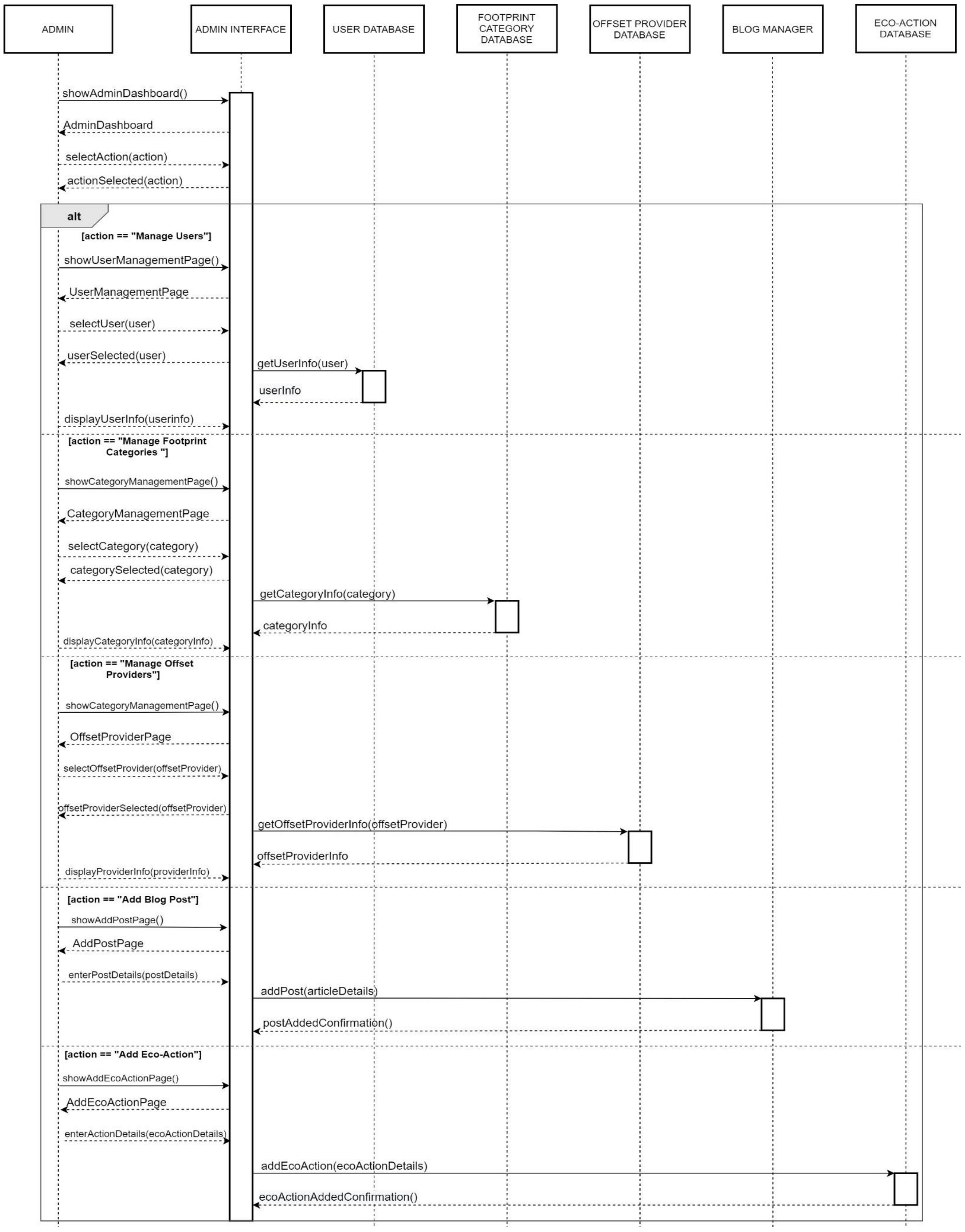
LOGIN



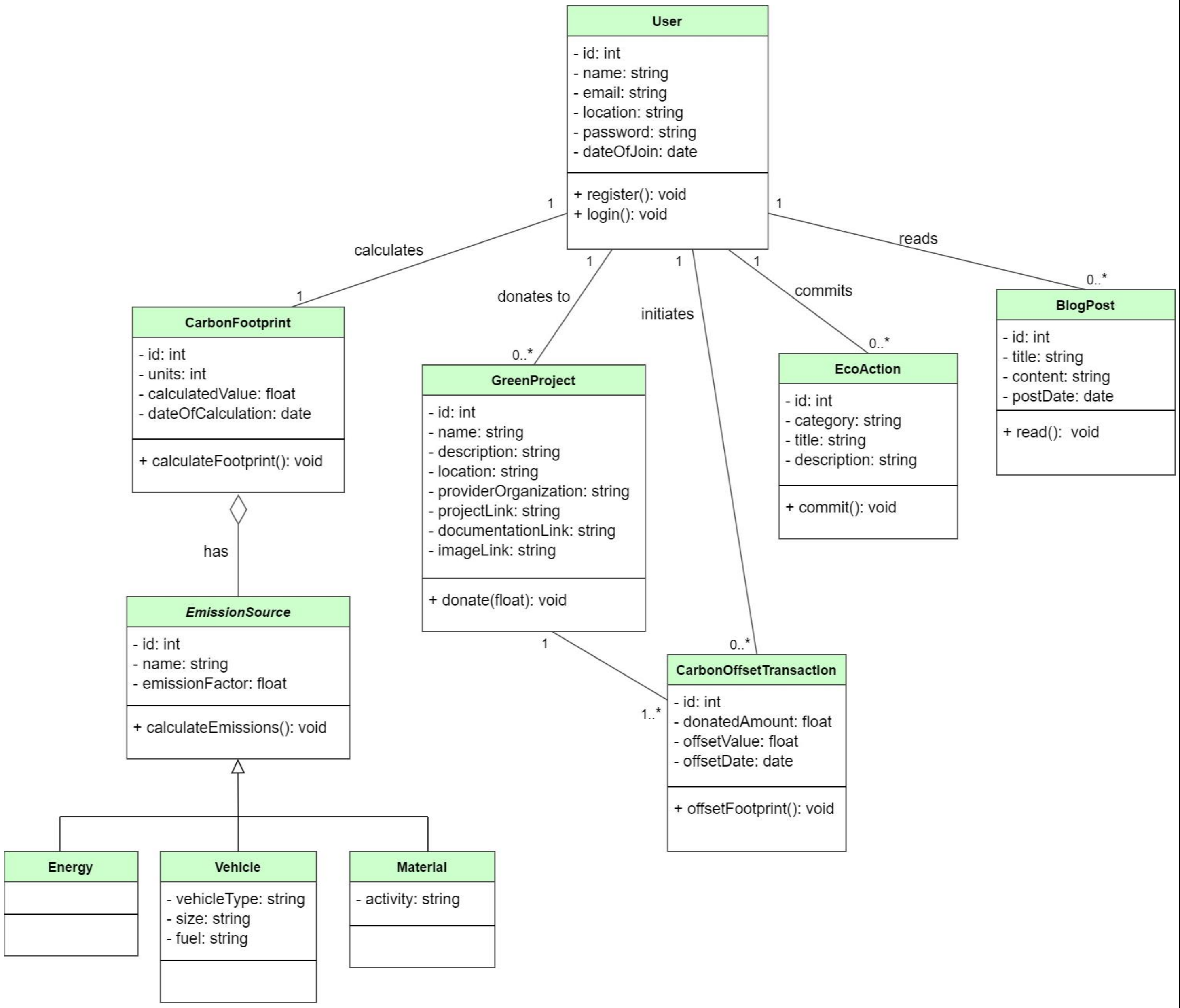
USER


USER
PAGES
CARBON FOOTPRINT
GREEN PROJECT
OFFSET TRANSACTION
ECO-ACTION
BLOG
REPORTS
showUserHomePage()
UserHomePage
selectAction(action)
actionSelected(action)
alt
[action == "Calculate Footprint"]
showCalculateFootprintPage()
calculateFootprintPage
calculateCarbonFootprint()
footprintResult
[action == "Offset Emissions"]
showGreenProjectsPage()
GreenProjectsPage
selectProject(project)
projectSelected(project)
donate(amount)
offsetFootprint()
offsetFootprint()
[action == "Commit to Eco-action"]
showEcoActionsPage()
EcoActionsPage
selectEcoAction(ecoAction)
ecoActionSelected(ecoAction)
commit()
commitConfirmation()
[action == "Read Blog Article"]
showBlogPage()
BlogPage
selectArticle(article)
articleSelected(article)
getArticleContent(article)
content
displayArticle(content)
[action == "View Reports"]
showReportsPage()
ReportsPage
selectReport(report)
reportSelected(report)
generateReport(report)
report
displayReport(report)

ADMIN


ADMIN
ADMIN INTERFACE
USER DATABASE
FOOTPRINT CATEGORY DATABASE
OFFSET PROVIDER DATABASE
BLOG MANAGER
ECO-ACTION DATABASE
showAdminDashboard()
AdminDashboard
selectAction(action)
actionSelected(action)
alt
[action == "Manage Users"]
showUserManagementPage()
UserManagementPage
selectUser(user)
userSelected(user)
getUserInfo(user)
userInfo
displayUserInfo(userinfo)
[action == "Manage Footprint Categories "]
showCategoryManagementPage()
CategoryManagementPage
selectCategory(category)
categorySelected(category)
getCategoryInfo(category)
categoryInfo
displayCategoryInfo(categoryInfo)
[action == "Manage Offset Providers"]
showCategoryManagementPage()
OffsetProviderPage
selectOffsetProvider(offsetProvider)
offsetProviderSelected(offsetProvider)
getOffsetProviderInfo(offsetProvider)
offsetProviderInfo
displayProviderInfo(providerInfo)
[action == "Add Blog Post"]
showAddPostPage()
AddPostPage
enterPostDetails(postDetails)
addPost(articleDetails)
postAddedConfirmation()
[action == "Add Eco-Action"]
showAddEcoActionPage()
AddEcoActionPage
enterActionDetails(ecoActionDetails)
addEcoAction(ecoActionDetails)
ecoActionAddedConfirmation()

7.5. CLASS DIAGRAM



7.6. THEORETICAL BACKGROUND

Several technologies, tools, and algorithms were utilized to develop the carbon footprint web application. Here are the key components:

1. Technology Stack:

- Front-end: The user interface was created using HTML5, CSS3, and JavaScript, to provide an interactive experience. React.js a popular JavaScript library is used to create reusable UI components and manage the application's state.
- Back-end: The application is built using Node.js, which allows us to use JavaScript for server-side scripting and handle the application's logic. The Express.js framework is used to build the web application's server-side components, handle routing, and manage HTTP requests and responses (REST API).
- Database: MySQL is used to store and manage the application's data.

2. Calculations:

The carbon footprint calculations in the application are based on established methodologies and formulas and the emission factors are based on the GHG reporting.

3. APIs:

- The offset projects were provided by the Global Giving API.
- The Stripe API was used to integrate the payment processing feature into the project.

4. Version Control System:

GitHub was used to do code versioning and track changes during development.

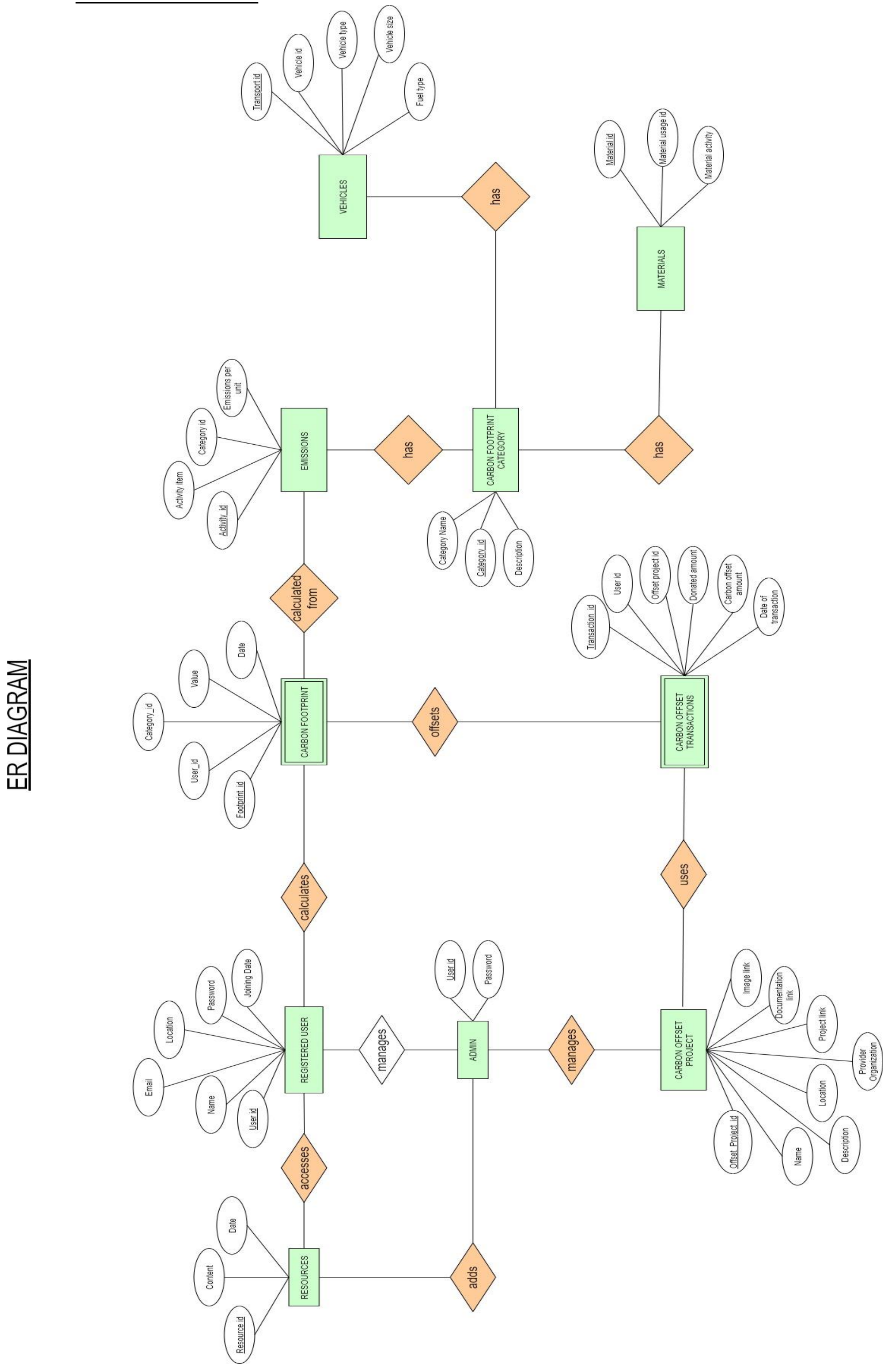
5. IDE:

The environment for coding and debugging was provided by Visual Studio Code IDE.

6. UI/UX Design Tools:

Figma was used to create wireframes and visual designs for the user interface.

7.7. ER DIAGRAM



7.8. DATABASE DESIGN

7.8.1. User Table

Field	**Type (length)**	**Constraint**	**Description**
Userid	Int	Primary key	
Name	Varchar(45)		Organization name
Email	Varchar(45)		Organization email
Location	Varchar(45)		Organization location
Password	Varchar(500)		
Join Date	Date		Date of creation of account

7.8.2. Footprint Category Table

Field	**Type (length)**	**Constraint**	**Description**
Category id	Int	Primary key	
Category name	Varchar(45)		Footprint category name such as Electricity, Fuel, Water, etc.
Category_description	Varchar(255)		Description of category

7.8.3. Emission Source Table

Field	**Type (length)**	**Constraint**	**Description**
Emission id	Int	Primary key	
Source name	Varchar(45)		Item whose emissions are under consideration
Category id	Int	Foreign key	
Emission factor	Float		Emissions per unit

7.8.4. Vehicles Table

Field	**Type (length)**	**Constraint**	**Description**
Vehicle id	Int	Primary key	
Emission id	Int	Foreign key	ID of vehicle in Emissions table
Vehicle type	Varchar(45)		Type of vehicle (car, truck, etc)
Vehicle size	Varchar(45)		Size of vehicle (small, medium, large)
Fuel type	Varchar(45)		Type of fuel (petrol, diesel, etc)

7.8.5. Materials Table

Field	Type (length)	Constraint	Description
Material id	Int	Primary key	
Emission id	Int	Foreign key	ID of material in Emissions table
Material activity	Varchar(45)		Industrial activity name

7.8.6. User Footprint Table

Field	Type (length)	Constraint	Description
Footprint id	Int	Primary key	
User id	Int	Foreign key	
Emission id	Int	Foreign key	ID of item in Emissions table
Units per quantity	Int		Units of the metric under consideration (km, liters, etc.)
Quantity	Int		Number of items
Calculated value	Float		Footprint value
Calculated date	Date		Date of calculation

7.87. Offset Project Table

Field	Type (length)	Constraint	Description
Project id	Int	Primary key	
Project Name	Varchar(45)		Name of offset project
Project Description	Text		Description of project
Location	Varchar(45)		Project location
Provider Organization	Varchar(255)		Name of organization running the project
Project link	Varchar(255)		Project website link
Documentation link	Varchar(255)		Link of additional documentation if any
Image url	Varchar(255)		Link of project image

7.8.8. Carbon Offset Table

Field	Type (length)	Constraint	Description
Offset id	Int	Primary key	ID of offset transaction
User id	Int	Foreign key	
Project id	Int	Foreign key	ID of project in Offset Project table
Donated amount	Int		Amount donated to project
Offset value	Float		Amount of carbon offset
Offset date	Date		Date of transaction

7.8.9. Blog Post Table

Field	Type (length)	Constraint	Description
Post id	Int	Primary key	
Post title	Text		Title of the blog post
Post content	Text		Body of the post
Post date	Date		Date of posting

7.8.10. Actions Table

Field	Type (length)	Constraint	Description
Action id	Int	Primary key	
Category id	Int	Foreign key	Footprint category id
Action title	Varchar(255)		Name of the carbon reduction action
Action description	Text		Description of the action

8. TESTING

8.1. TEST CASES

8.1.1. Test Case: User Registration

Description: Verify that a user can successfully register for an account in the carbon footprint web app, including validation checks.

Test Steps:
1. Open the carbon footprint web app.
2. Navigate to the registration page.
3. Enter valid user details in the registration form, including a unique email address.
4. Verify that the email address entered is in the correct format.
5. Verify that the password meets the specified complexity requirements.
6. Submit the registration form.

Inputs:
- Name: ABC Company
- Email: abc@example.com
- Location: India
- Password: abc@1234
- Confirm Password: abc@1234

Expected Output:
- The inputs entered are in the correct format.
- The password meets the specified complexity requirements.
- The confirm password field input matches the password field input.
- The user registration form is submitted successfully without any errors.

Pass/Fail Criteria:
- The test case passes if all steps are executed successfully and the expected outputs are achieved.
- The test case fails if any step fails or the expected outputs are not met.

8.1.2. Test Case: User Login

Description: Verify that a user can successfully login to the carbon footprint web app with valid credentials.

Test Steps:

1. Open the carbon footprint web app.
2. Navigate to the login page.
3. Enter valid login credentials (email and password).
4. Submit the login form.
5. Verify that the user is successfully logged in and redirected to the app's dashboard or main page.
6. Verify that the user's name or profile information is displayed correctly.
7. Attempt to access a protected page or perform a restricted action.
8. Verify that the user is granted access and able to perform the desired action without any errors.

Inputs:

- Email: abc@example.com
- Password: Test123

Expected Output:

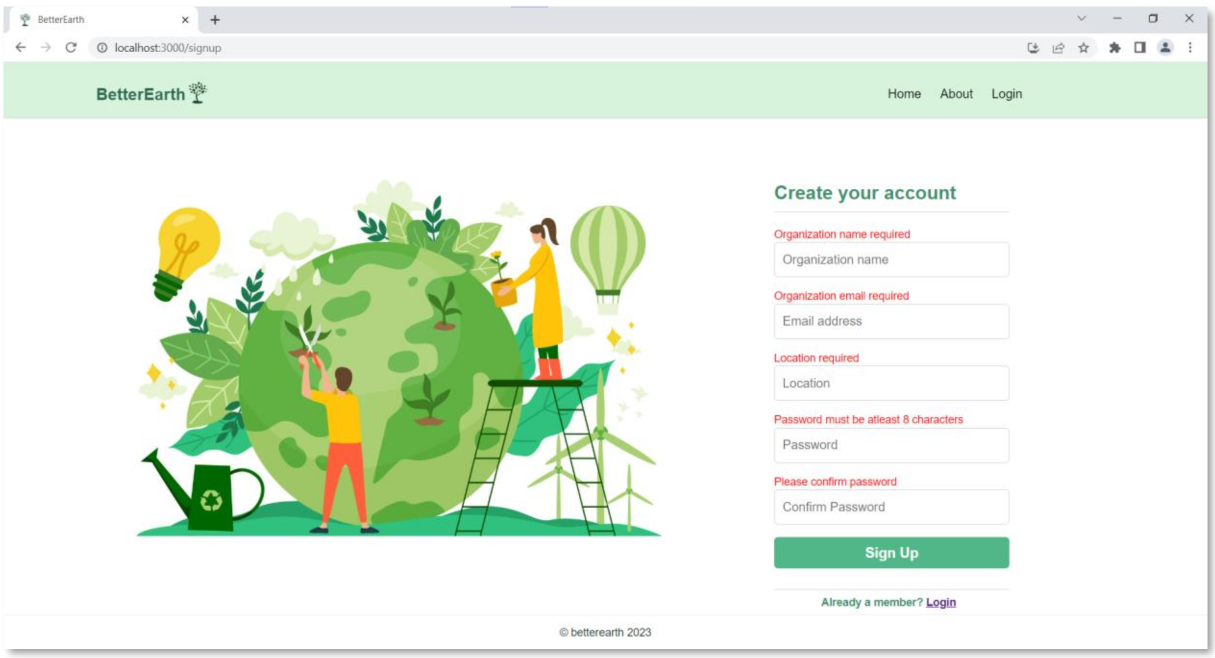
- The user is successfully logged in and redirected to the app's dashboard or main page.
- The user's name or profile information is displayed correctly, indicating a successful login.
- The user is granted access to protected pages or actions and can perform them without encountering any errors.

Pass/Fail Criteria:

- The test case passes if all steps are executed successfully and the expected outputs are achieved.
- The test case fails if any step fails or the expected outputs are not met.

8.2. TEST REPORTS

a. Empty fields



BetterEarth

localhost:3000/signup

BetterEarth Home About Login

Create your account

Organization name required

Organization name

Organization email required

Email address

Location required

Location

Password must be atleast 8 characters

Password

Please confirm password

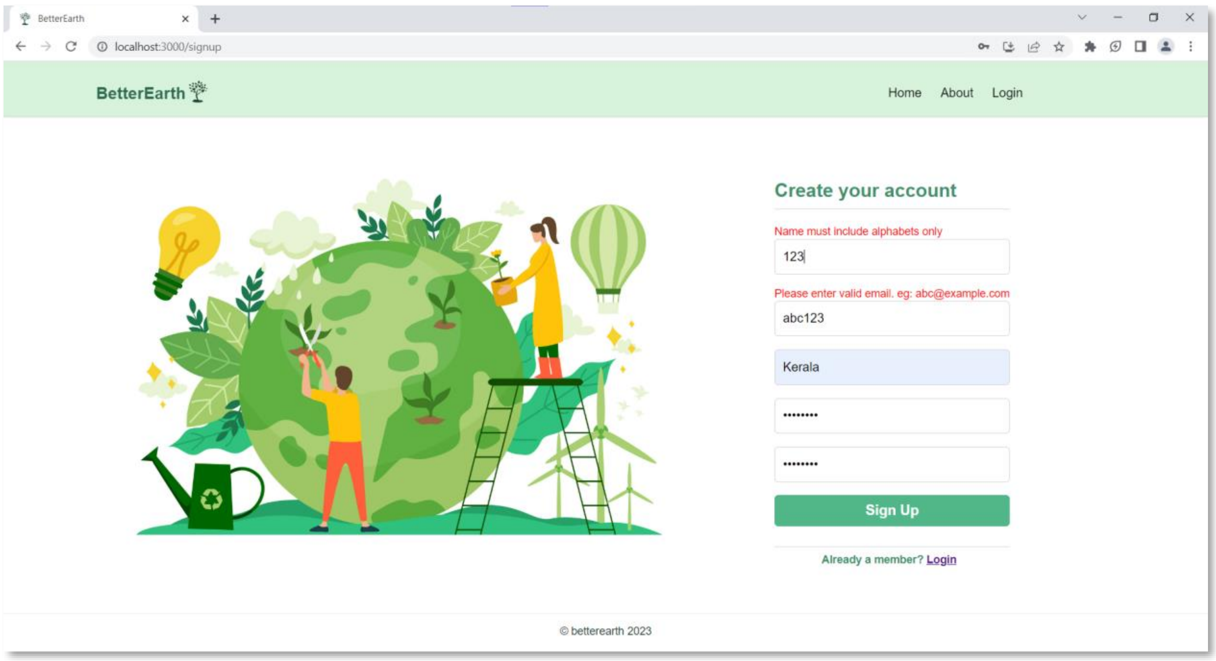
Confirm Password

Sign Up

Already a member? Login

© betterearth 2023

b. Inputs entered in wrong format



BetterEarth

localhost:3000/signup

BetterEarth Home About Login

Create your account

Name must include alphabets only

123

Please enter valid email. eg: abc@example.com

abc123

Kerala

••••••••

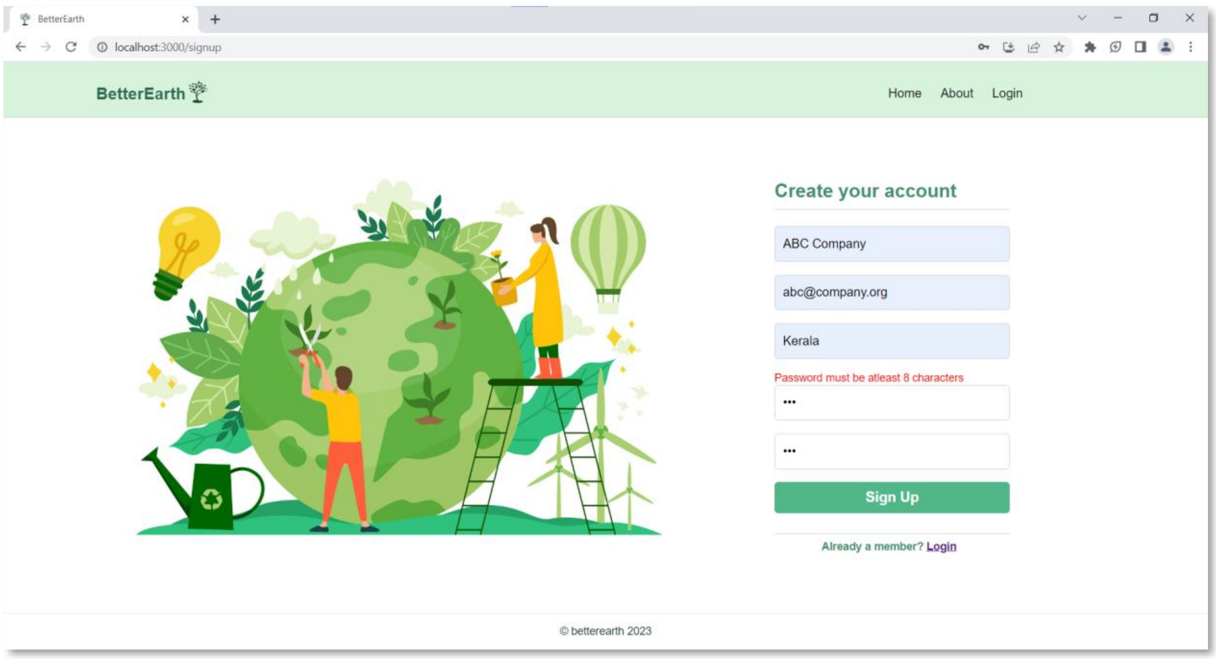
••••••••

Sign Up

Already a member? Login

© betterearth 2023

c. The password does not meet the specified complexity requirements.



Create your account

ABC Company

abc@company.org

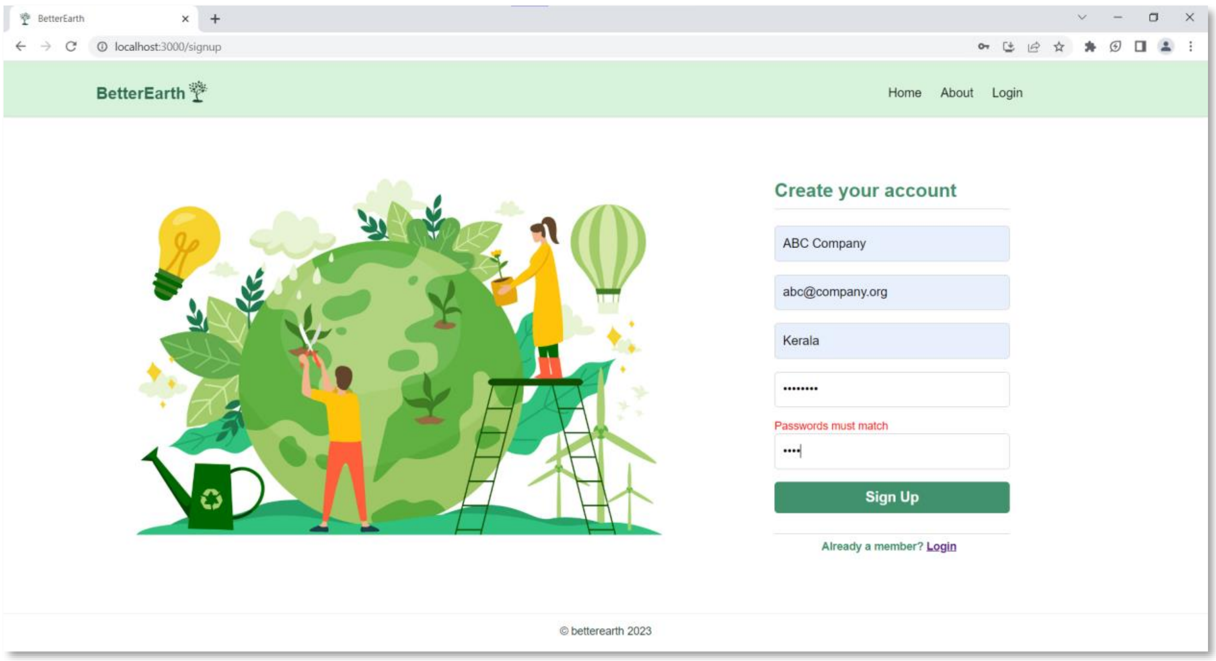
Kerala

Password must be atleast 8 characters

•••

•••

d. The confirm password field input does not match the password field input.



Create your account

ABC Company

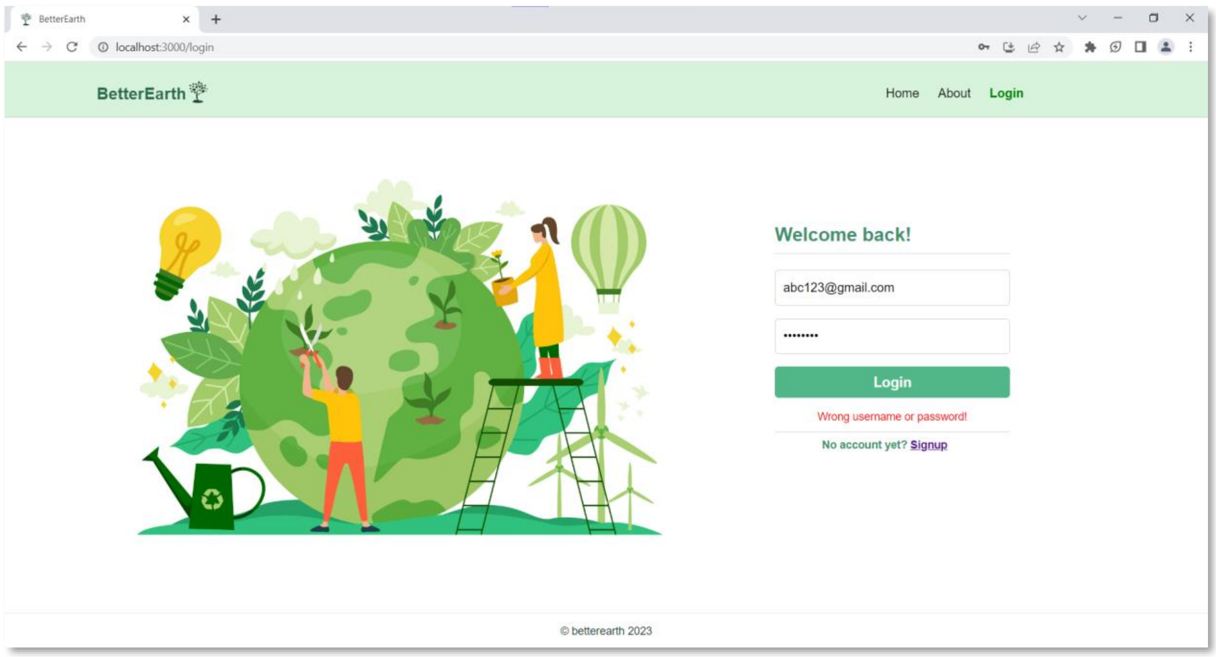
abc@company.org

Kerala

••••••••

Passwords must match

••••



8.3. SAMPLE CODE USED FOR TESTING

```
const schema = yup.object().shape({
    name: yup.string().matches(/^[aA-zZ\s]+$/, "Name must include alphabets
    only").required("Organization name required"),
    email: yup.string().email("Please enter valid email. eg:
    abc@example.com").matches(/^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-
    Z]{2,4}$/i).required("Organization email required"),
    location: yup.string().required("Location required"),
    password: yup.string().min(8, "Password must be atleast 8
    characters").max(20).required("*Password required"),
    confirmPassword: yup.string().oneOf([yup.ref("password"), null], "Passwords must
    match").required("Please confirm password"),
      });
```

9. TRANSITION

9.1. SYSTEM IMPLEMENTATION

The implementation mechanisms refer to the methods and approaches used to develop and deploy a software application, such as a carbon footprint web application. The common implementation mechanisms are:

1. Requirement Analysis: The first step is to gather and analyze the requirements of the carbon footprint web application. This involves understanding the goals, functionalities, and features that the application needs to provide.

2. System Design: Based on the requirements, a system design is created. This includes defining the overall architecture of the application, deciding on the technologies and frameworks to be used, and designing the user interface (UI) and database schema.

3. Front-end Development: The front-end development involves implementing the user interface using HTML, CSS, and JavaScript. This includes designing and coding the UI elements, user interactions, and visual components of the application.

4. Back-end Development: The back-end development focuses on implementing the server-side logic and functionality. It involves choosing a suitable programming language (JavaScript), a runtime environment (Node.js) and a web framework (Express.js) to handle server-side operations, data processing, and integration with databases and external systems.

5. Database Development: The implementation of the database includes designing the database schema, creating the necessary tables, and defining relationships between entities. This involves selecting a suitable database management system such as MySQL and utilizing database query languages such as SQL to handle data storage, retrieval, and manipulation.

6. Integration with APIs and Services: The carbon footprint web application may require integration with external APIs or services to fetch data or provide additional functionality. This involves using API documentation to establish connections, make requests, and handle responses from the external services.

7. Testing and Quality Assurance: Testing is a crucial part of the implementation process to ensure the application functions as expected. It involves various types of testing, including unit testing to verify individual components, integration testing to test the interaction between different modules, system testing to validate the overall system behavior, and user acceptance testing to ensure the application meets the user's requirements.

8. Deployment and Release: Once the application has been thoroughly tested, it is deployed to a production environment. This involves setting up servers, configuring the application, and ensuring it is accessible to users.

9. Maintenance and Updates: After the initial release, ongoing maintenance and updates are required to address bugs, add new features, and improve performance. This includes monitoring the application, applying security patches, and conducting periodic maintenance activities to ensure the application remains secure and up-to-date.

9.2. SYSTEM MAINTENANCE

The maintenance plan for the carbon footprint web application ensures its ongoing functionality, performance, security, and updates. Here is an outline of the plan, including relevant documents and training for future maintenance:

1. Documentation:
   - System Documentation: Provide detailed documentation that describes the architecture, components, and functionality of the carbon footprint web application. This documentation includes system diagrams, database schemas, API documentation, and any relevant technical specifications.
   - Code Documentation: Document the source code of the application, including comments, code structure, and any necessary instructions for maintenance and troubleshooting.
   - User Documentation: Create user manuals or guides that explain how to use the application, its features, and any specific user workflows.

2. Bug Tracking and Issue Management:
   - Establish a bug tracking system or utilize an issue tracking tool to track reported

   bugs, system errors, and user feedback.
- Define a process for prioritizing and addressing reported issues, ensuring timely resolution.

3. Version Control:
   - Utilize a version control system, such as Git, to manage the source code and track changes made to the application.
   - Establish branching strategies and guidelines for managing different versions or feature branches of the application.

4. Regular Updates and Maintenance:
   - Plan and schedule regular updates and maintenance activities for the application to ensure it remains secure, stable, and up-to-date with the latest technologies and frameworks.
   - Stay informed about new security patches, bug fixes, and updates for the underlying technologies used in the application and apply them as necessary.

5. Backup and Disaster Recovery:
   - Implement a backup strategy to regularly back up the application's data and ensure its availability in case of data loss or system failure.
   - Establish a disaster recovery plan to recover the application in the event of a catastrophic event or system failure.

6. Continuous Monitoring and Performance Optimization:
   - Set up monitoring tools and systems to track the performance and health of the application, including server monitoring, database monitoring, and application performance monitoring.
   - Monitor key metrics, such as response time, resource utilization, and user activity, and take proactive measures to optimize performance and address any bottlenecks.

7. Training for Future Maintenance:
   - Provide comprehensive training to the relevant individuals or teams responsible for future maintenance of the application.
   - Conduct training sessions to familiarize them with the system architecture,

codebase, deployment processes, and any specific maintenance procedures.

- Ensure that the maintenance team has access to the relevant documentation, user manuals, and resources to effectively maintain and support the carbon footprint web application.

By following this maintenance plan and providing the necessary documentation and training, the carbon footprint web application can be efficiently maintained and supported in the long term, ensuring its continued functionality and usability.

10. CONCLUSION

In conclusion, this documentation provides a comprehensive overview of the system's design, development, and implementation. It includes various sections that cover the project's scope, objectives, requirements, architecture, testing, and maintenance plan.

The carbon footprint web application helps organizations to track and reduce carbon emissions. The system's functionalities include calculating and analyzing carbon footprints, providing recommendations for emission reduction, and generating reports. The documentation also highlights the technologies, methodologies, and tools used in the project, including the web application framework, database management system, and version control. The constraints and risks along with the mitigation strategies and contingency plans have been provided to ensure smooth project execution and minimize risks. Furthermore, the project documentation emphasizes the iterative and incremental development approach, allowing for continuous feedback and improvement throughout the development process. The testing strategies, including unit, integration, system, and user acceptance testing, have been outlined to ensure the quality and reliability of the system. The documentation also covers the plan for ongoing maintenance and support.

Overall, the project documentation serves as a comprehensive reference for stakeholders involved in the Carbon Footprint Web Application project. It provides a clear understanding of the system's functionality, implementation mechanisms, and maintenance plan. With this documentation, organizations can effectively track and reduce their carbon emissions, contributing to a more sustainable future.

11. ANNEXURE

11.1. REFERENCES

Websites

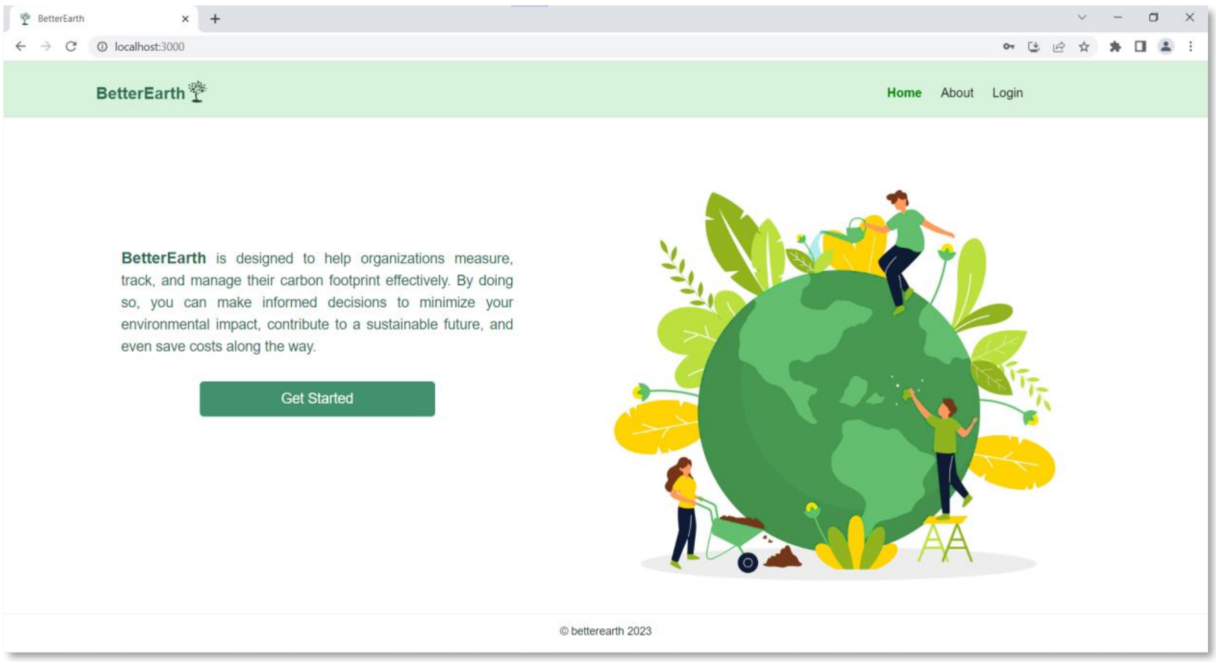
- https://www.globalgiving.org/api/methods/get-all-projects-for-a-theme
- https://www.britannica.com/science/carbon-footprint
- Government conversion factors for company reporting of greenhouse gas emissions - (www.gov.uk)
- https://ourworldindata.org/
- https://unfccc.int/climate-action/climate-neutral-now/reduce-your-emissions/tips-to-reduce-your-emissions
- Carbon footprint - Wikipedia

11.2. USER INTERVIEW QUESTIONNAIRES

1. How would you approach the system?
2. What about usability of this system?
3. What are normal project requirements?

11.3. OUTPUT SCREENS

Home



BetterEarth

localhost:3000

BetterEarth

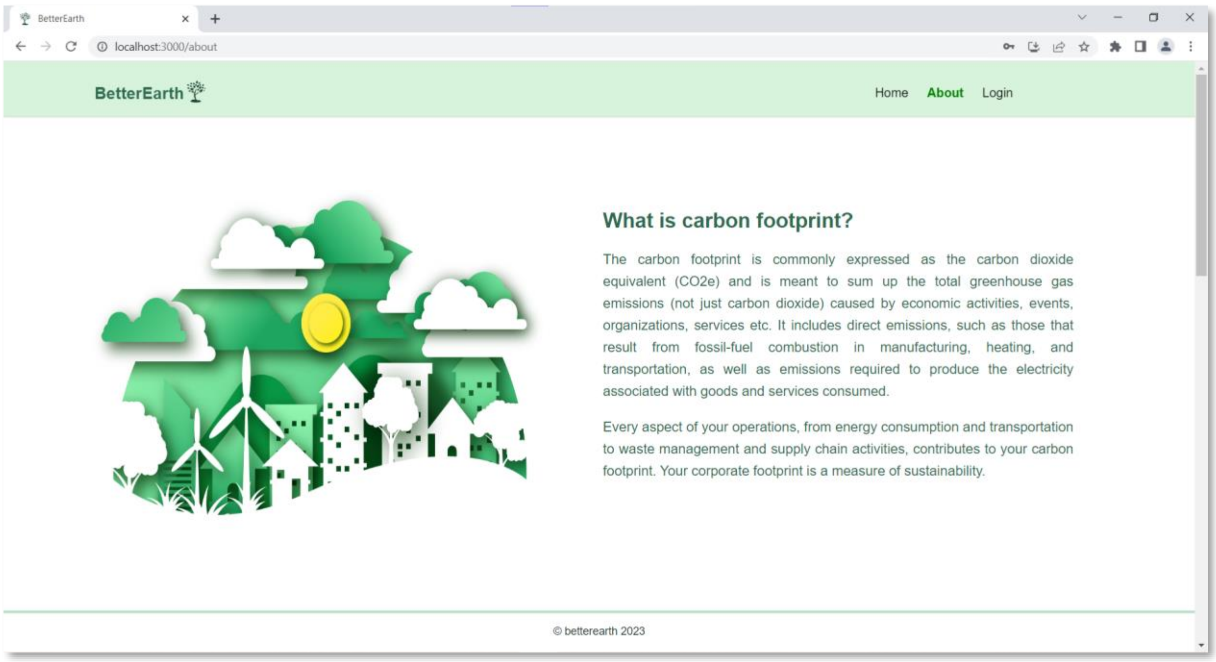
Home About Login

BetterEarth is designed to help organizations measure, track, and manage their carbon footprint effectively. By doing so, you can make informed decisions to minimize your environmental impact, contribute to a sustainable future, and even save costs along the way.

Get Started

© betterearth 2023

About



BetterEarth

localhost:3000/about

BetterEarth

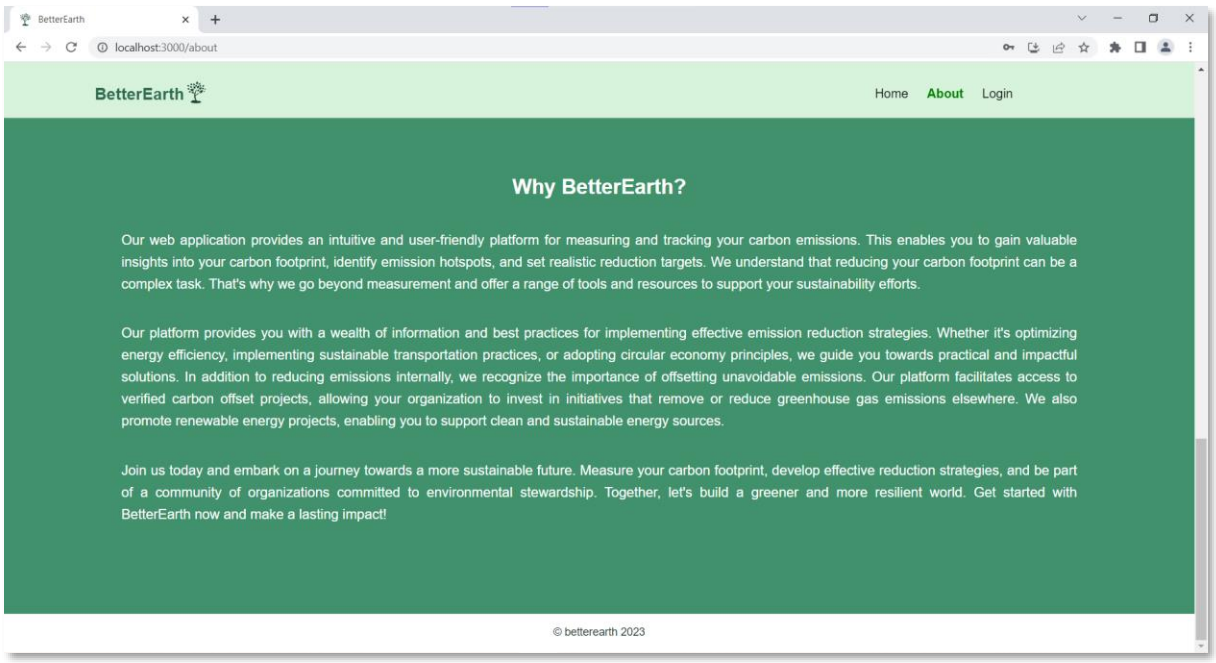
Home About Login

What is carbon footprint?

The carbon footprint is commonly expressed as the carbon dioxide equivalent (CO2e) and is meant to sum up the total greenhouse gas emissions (not just carbon dioxide) caused by economic activities, events, organizations, services etc. It includes direct emissions, such as those that result from fossil-fuel combustion in manufacturing, heating, and transportation, as well as emissions required to produce the electricity associated with goods and services consumed.

Every aspect of your operations, from energy consumption and transportation to waste management and supply chain activities, contributes to your carbon footprint. Your corporate footprint is a measure of sustainability.

© betterearth 2023



Why BetterEarth?

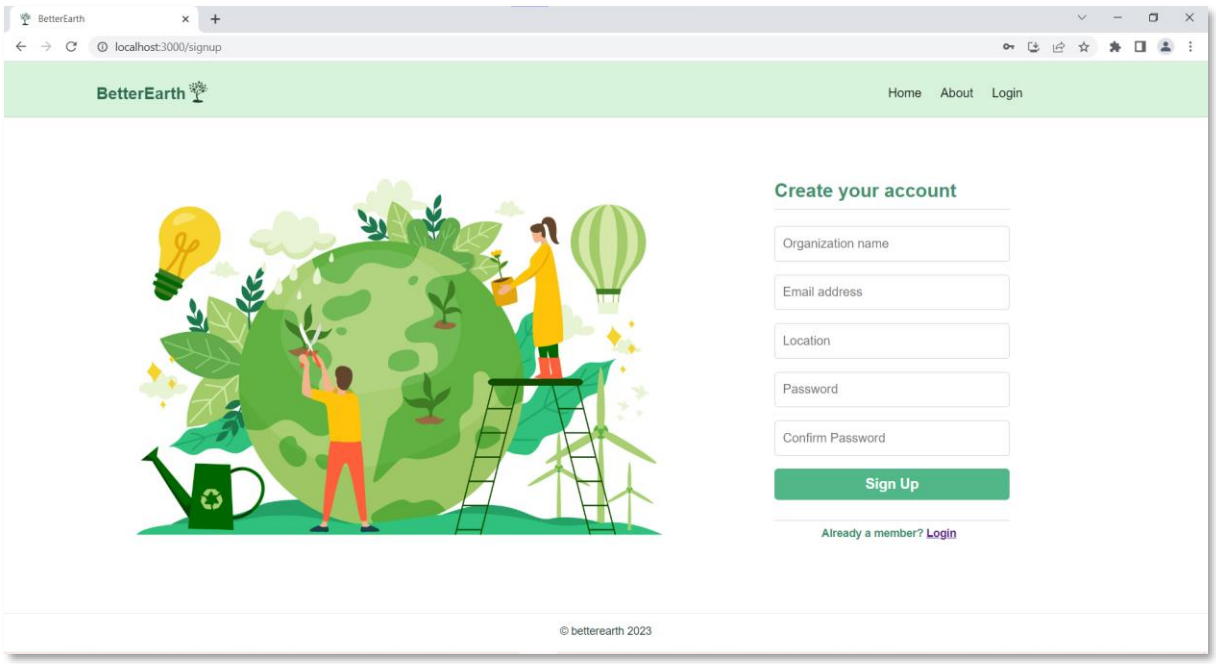
Our web application provides an intuitive and user-friendly platform for measuring and tracking your carbon emissions. This enables you to gain valuable insights into your carbon footprint, identify emission hotspots, and set realistic reduction targets. We understand that reducing your carbon footprint can be a complex task. That's why we go beyond measurement and offer a range of tools and resources to support your sustainability efforts.

Our platform provides you with a wealth of information and best practices for implementing effective emission reduction strategies. Whether it's optimizing energy efficiency, implementing sustainable transportation practices, or adopting circular economy principles, we guide you towards practical and impactful solutions. In addition to reducing emissions internally, we recognize the importance of offsetting unavoidable emissions. Our platform facilitates access to verified carbon offset projects, allowing your organization to invest in initiatives that remove or reduce greenhouse gas emissions elsewhere. We also promote renewable energy projects, enabling you to support clean and sustainable energy sources.

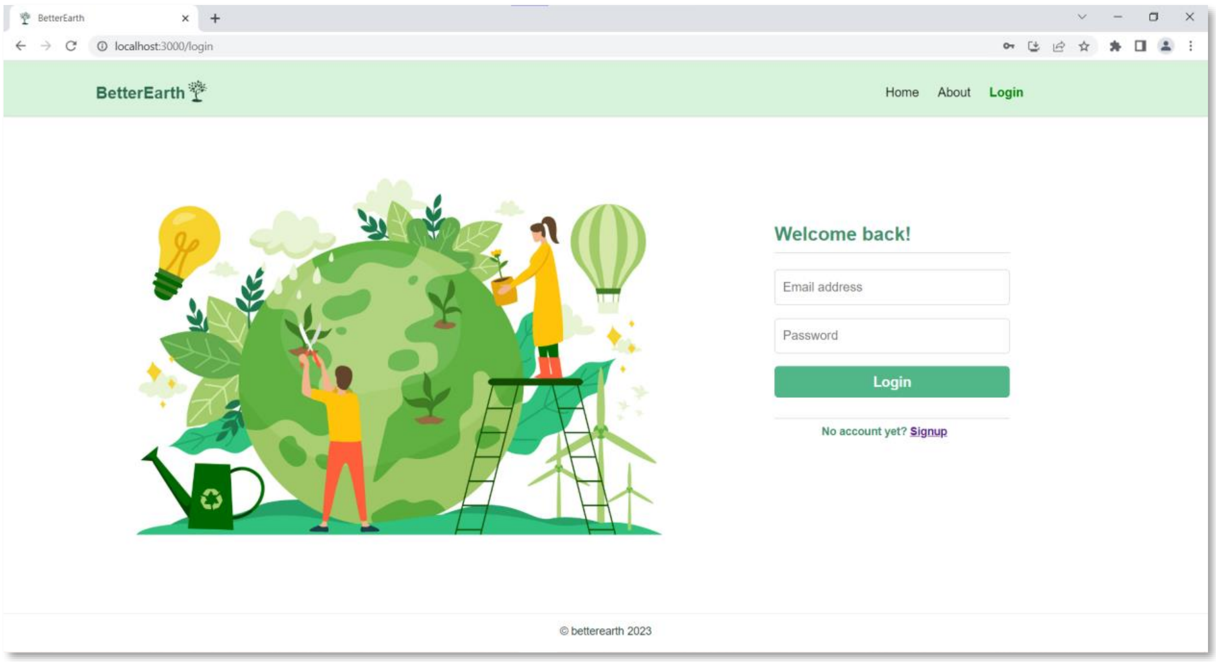
Join us today and embark on a journey towards a more sustainable future. Measure your carbon footprint, develop effective reduction strategies, and be part of a community of organizations committed to environmental stewardship. Together, let's build a greener and more resilient world. Get started with BetterEarth now and make a lasting impact!

© betterearth 2023

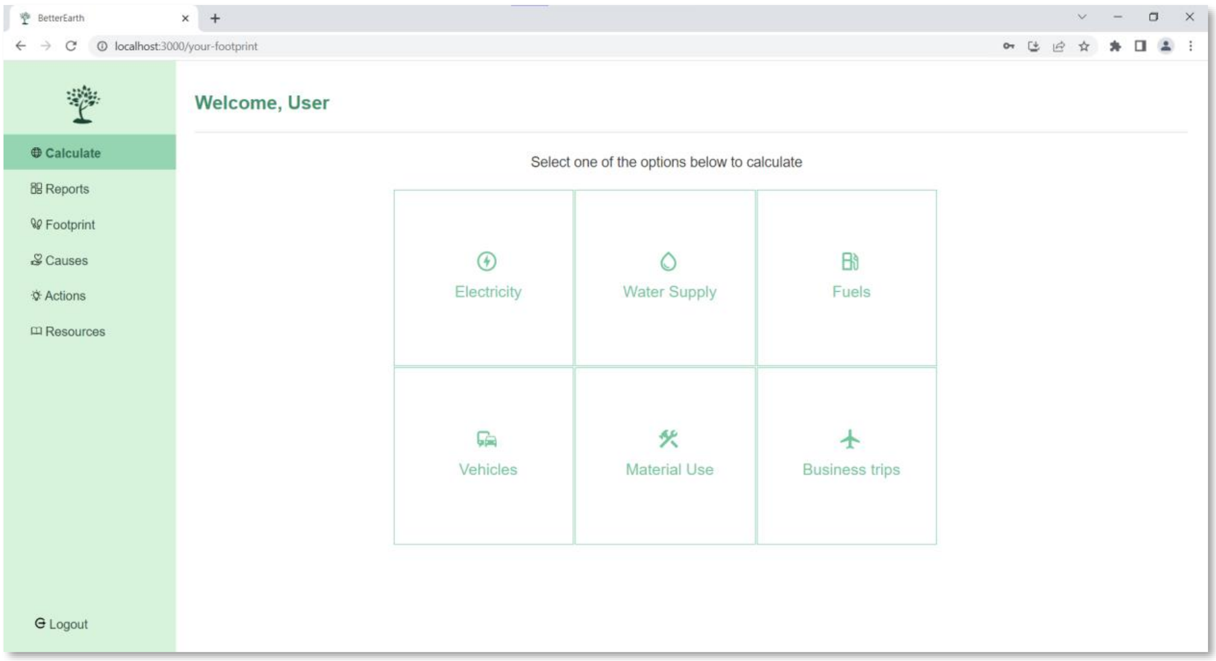
Sign Up



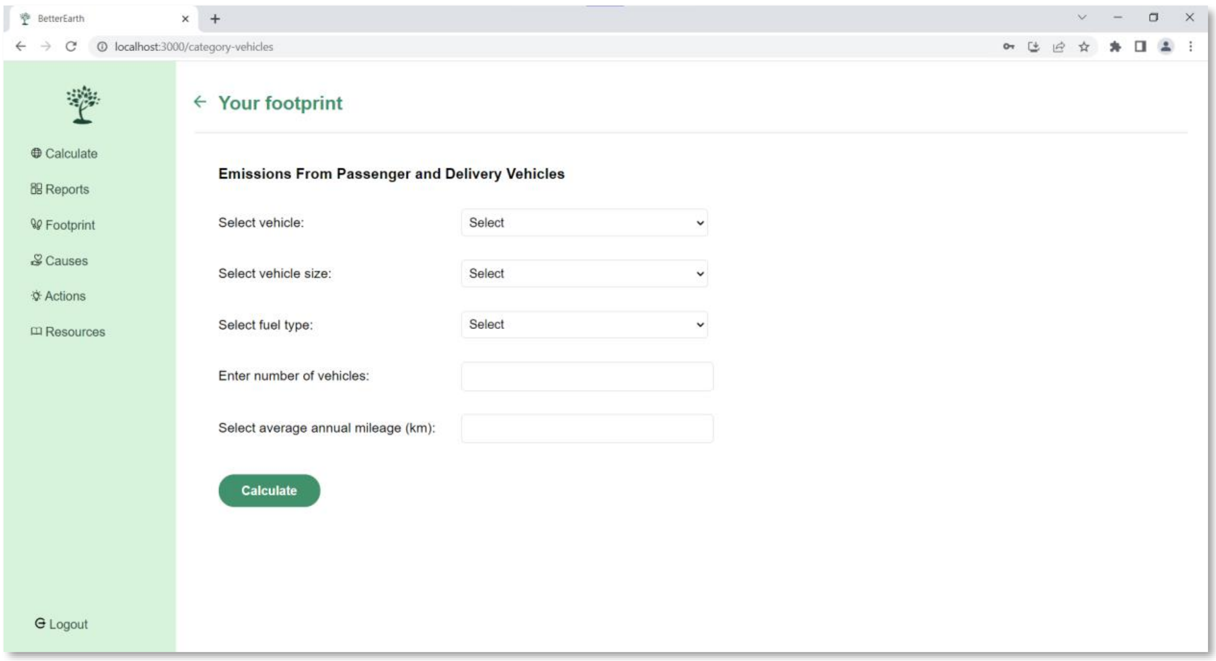
Login



Carbon Footprint Calculation



Vehicle Emissions Calculation


BetterEarth

localhost:3000/category-vehicles

Calculate
Reports
Footprint
Causes
Actions
Resources
Logout

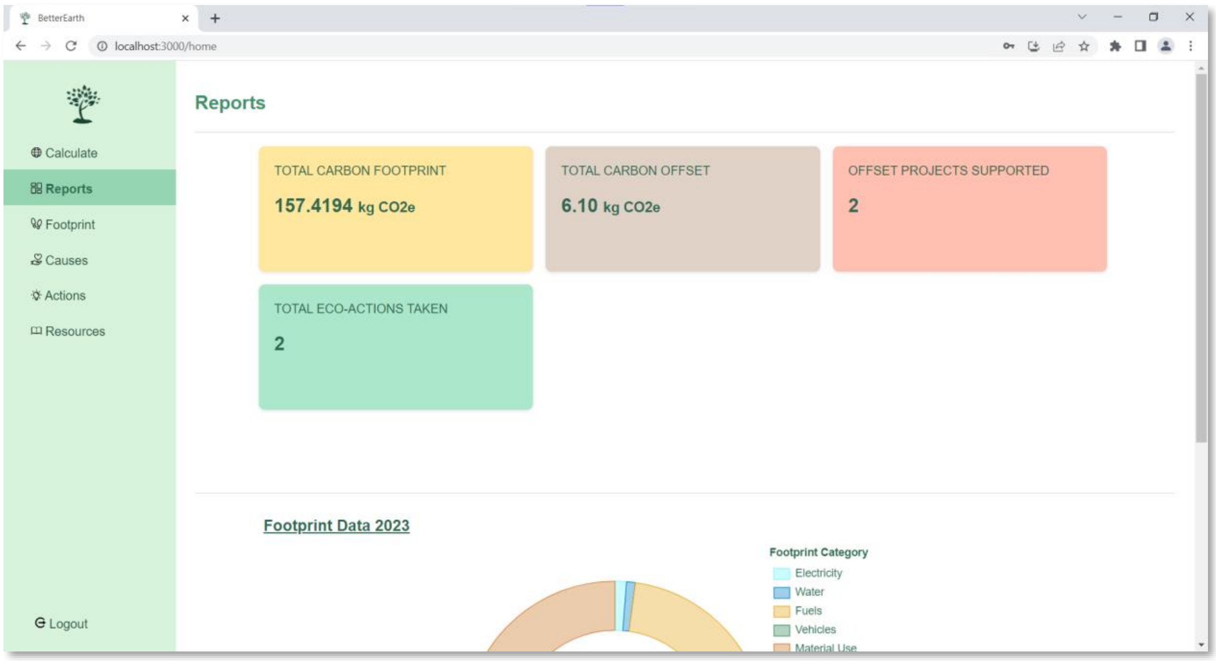
Your footprint

Emissions From Passenger and Delivery Vehicles

Select vehicle:	Select
Select vehicle size:	Select
Select fuel type:	Select
Enter number of vehicles:	
Select average annual mileage (km):	

Calculate

User Reports


BetterEarth

localhost:3000/home

Calculate
Reports
Footprint
Causes
Actions
Resources
Logout

Reports

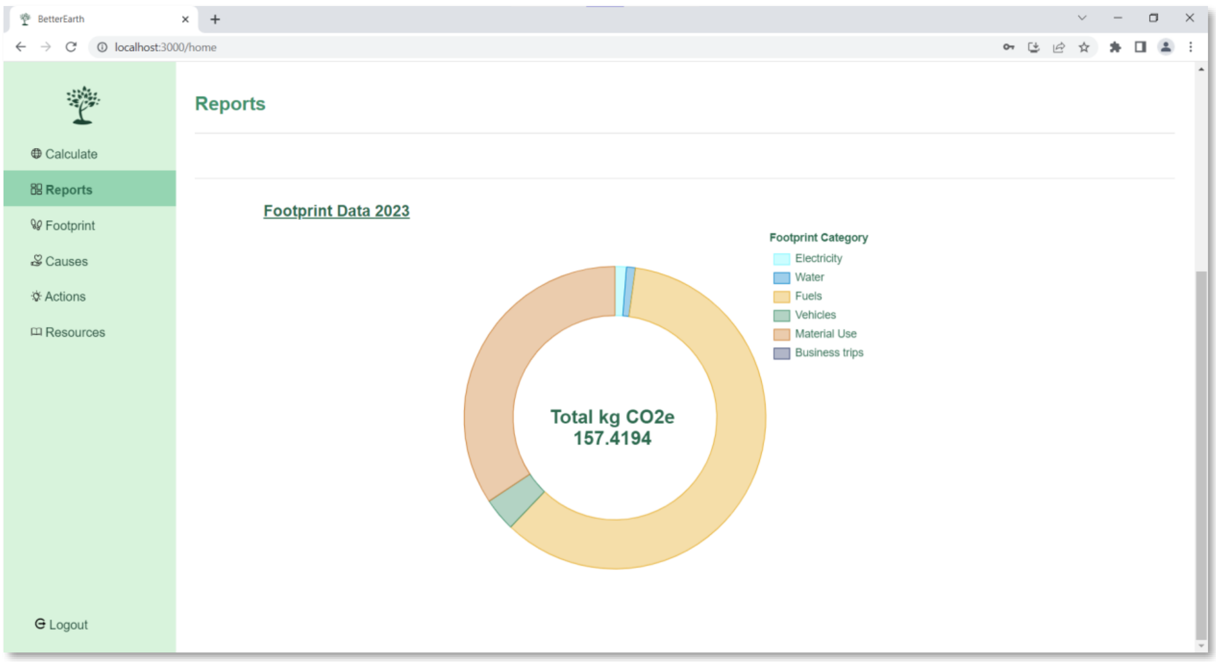
TOTAL CARBON FOOTPRINT
157.4194 kg CO2e

TOTAL CARBON OFFSET
6.10 kg CO2e

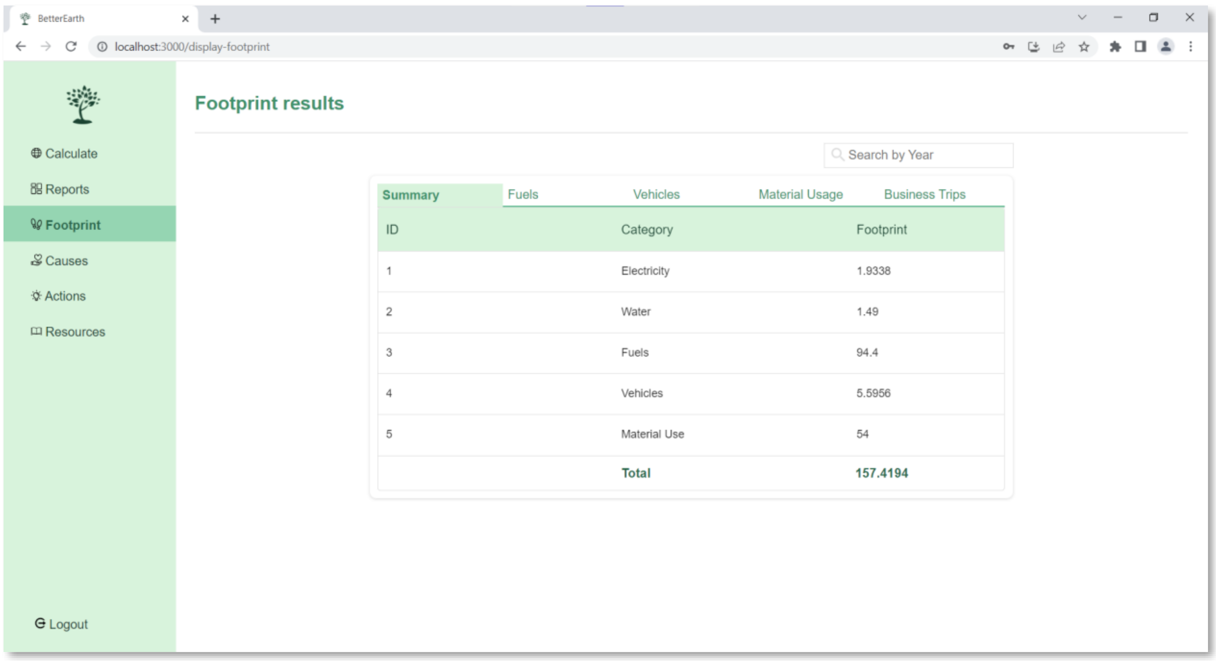
OFFSET PROJECTS SUPPORTED
2

TOTAL ECO-ACTIONS TAKEN
2

Footprint Data 2023



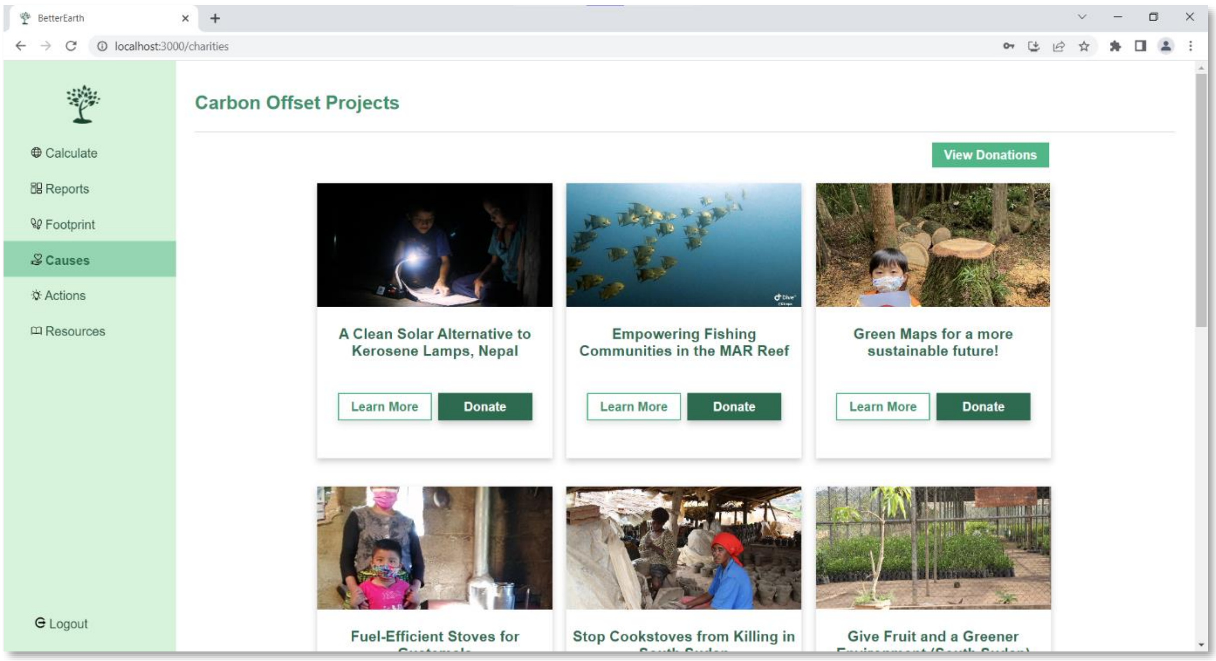
Footprint Results



Footprint results

ID	Category	Footprint
1	Electricity	1.9338
2	Water	1.49
3	Fuels	94.4
4	Vehicles	5.5956
5	Material Use	54
	Total	**157.4194**

Carbon Offset Projects



Carbon Offset Projects

A Clean Solar Alternative to Kerosene Lamps, Nepal

Empowering Fishing Communities in the MAR Reef

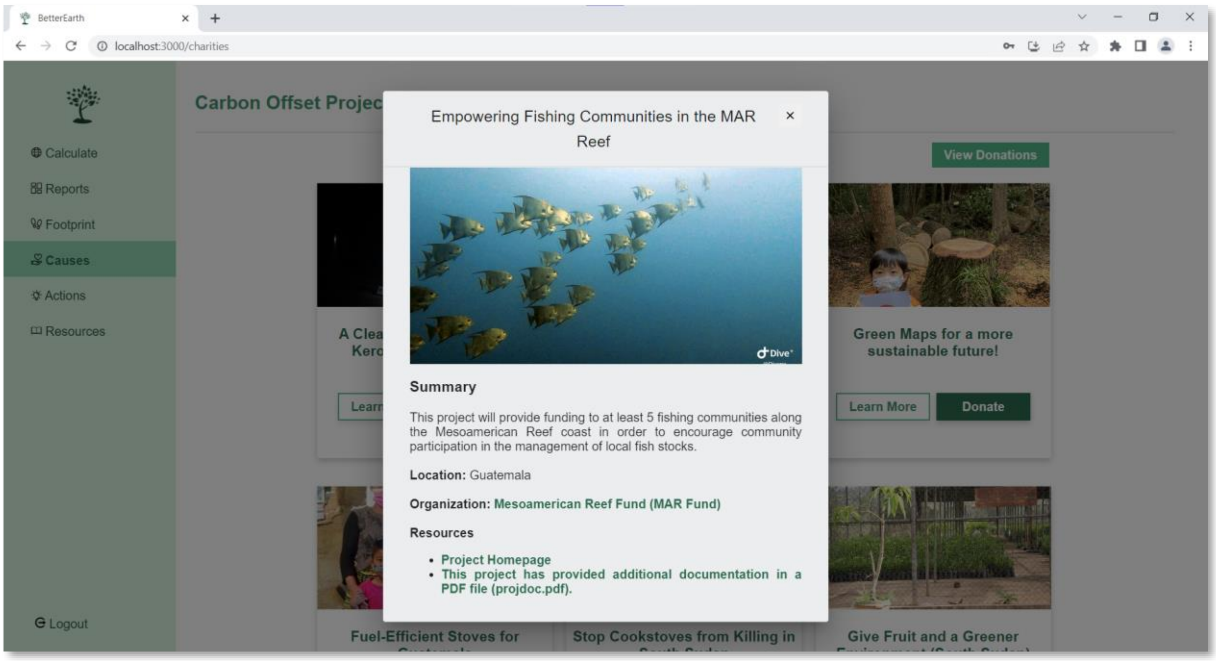
Green Maps for a more sustainable future!

Fuel-Efficient Stoves for

Stop Cookstoves from Killing in

Give Fruit and a Greener

Learn More



Empowering Fishing Communities in the MAR Reef

Summary

This project will provide funding to at least 5 fishing communities along the Mesoamerican Reef coast in order to encourage community participation in the management of local fish stocks.

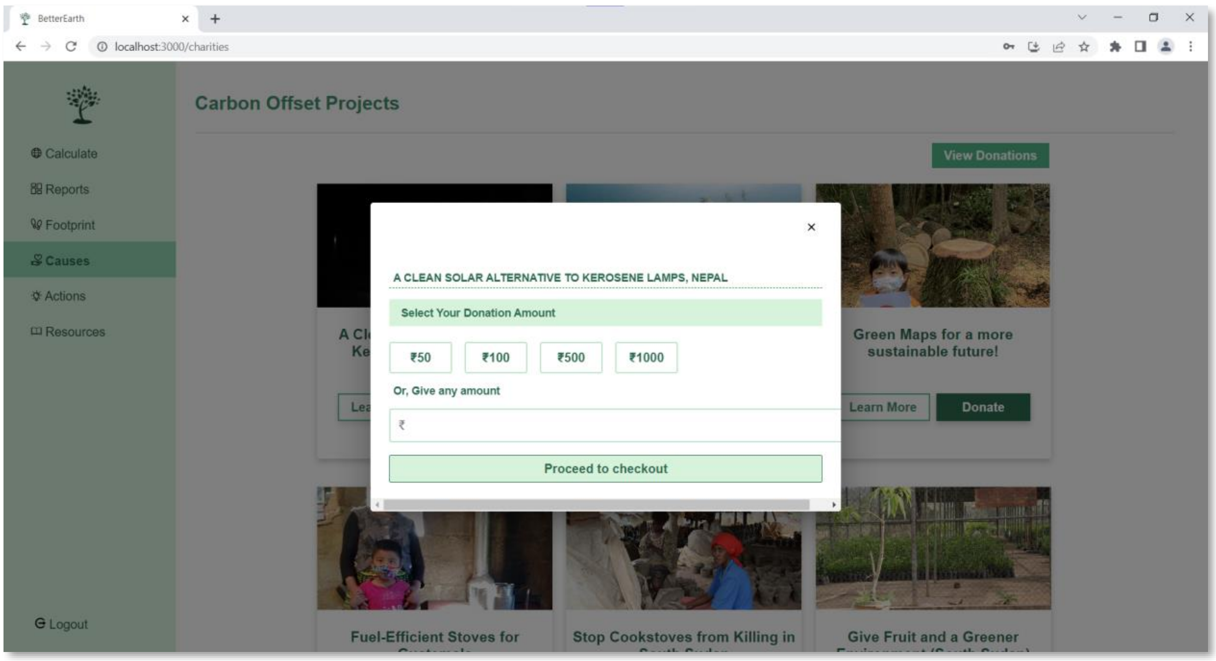
Location: Guatemala

Organization: Mesoamerican Reef Fund (MAR Fund)

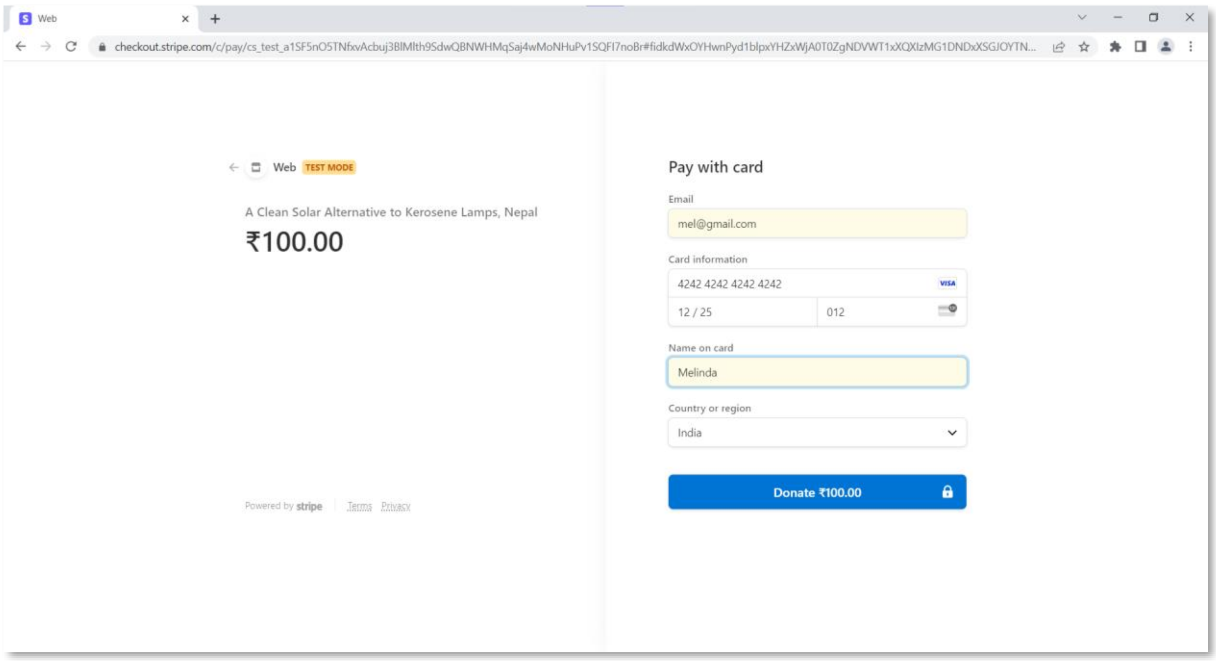
Resources

- Project Homepage
- This project has provided additional documentation in a PDF file (projdoc.pdf).

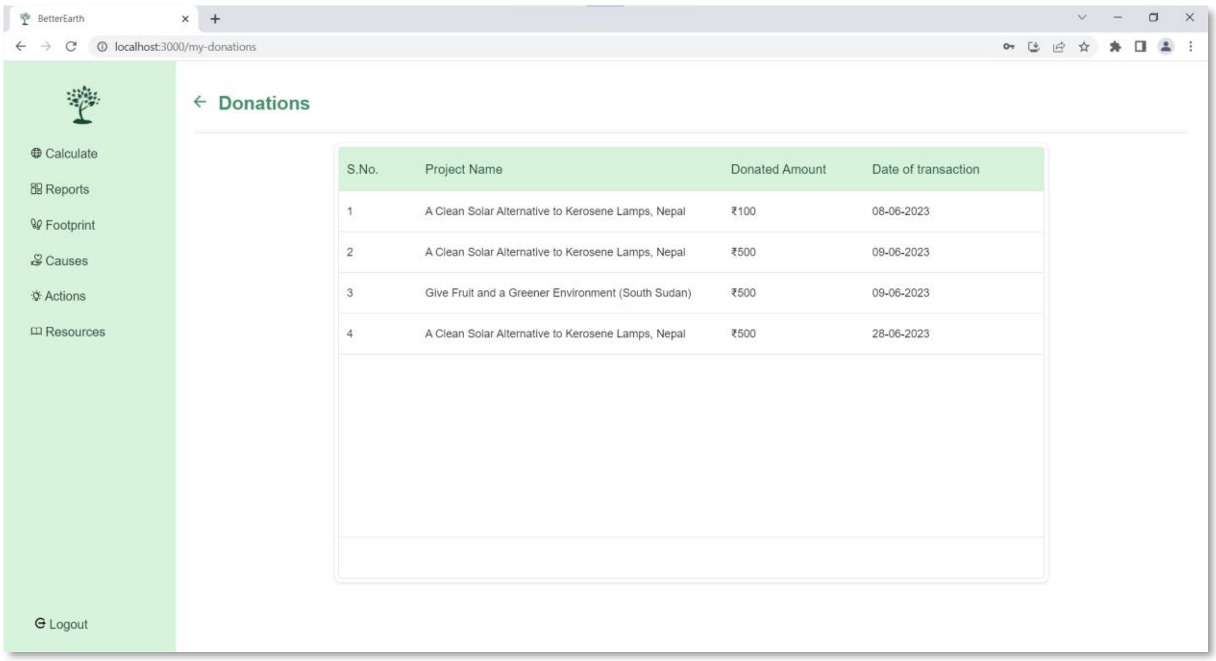
Donate



Payment


Web
checkout.stripe.com/c/pay/cs_test_a1SF5nO5TNfxvAcbuj3BlMlth95dwQBNWHMqSaj4wMoNHuPv1SQFI7noBr#fidkdWxOYHwnPyd1blpxYHZxWjA0T0ZgNDVWT1xXQXlzMG1DNDxXSGJOYTN...
Web TEST MODE
A Clean Solar Alternative to Kerosene Lamps, Nepal
₹100.00
Powered by stripe
Terms
Privacy
Pay with card
Email
mel@gmail.com
Card information
4242 4242 4242 4242
VISA
12 / 25
012
Name on card
Melinda
Country or region
India
Donate ₹100.00

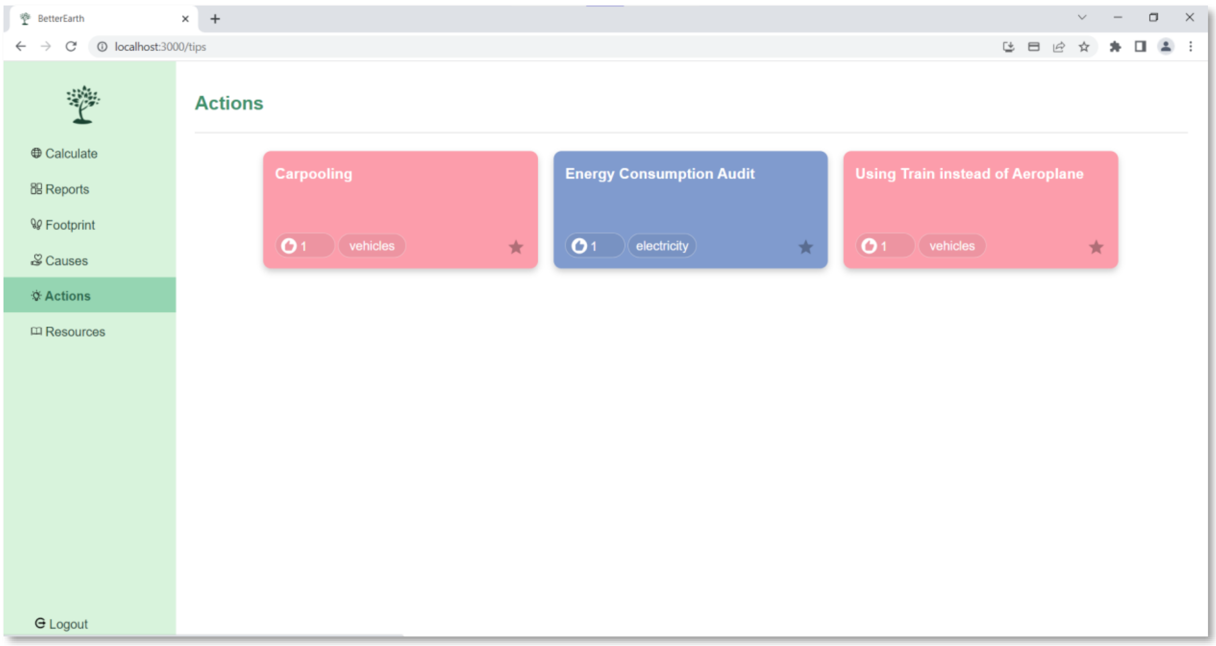
View Donations



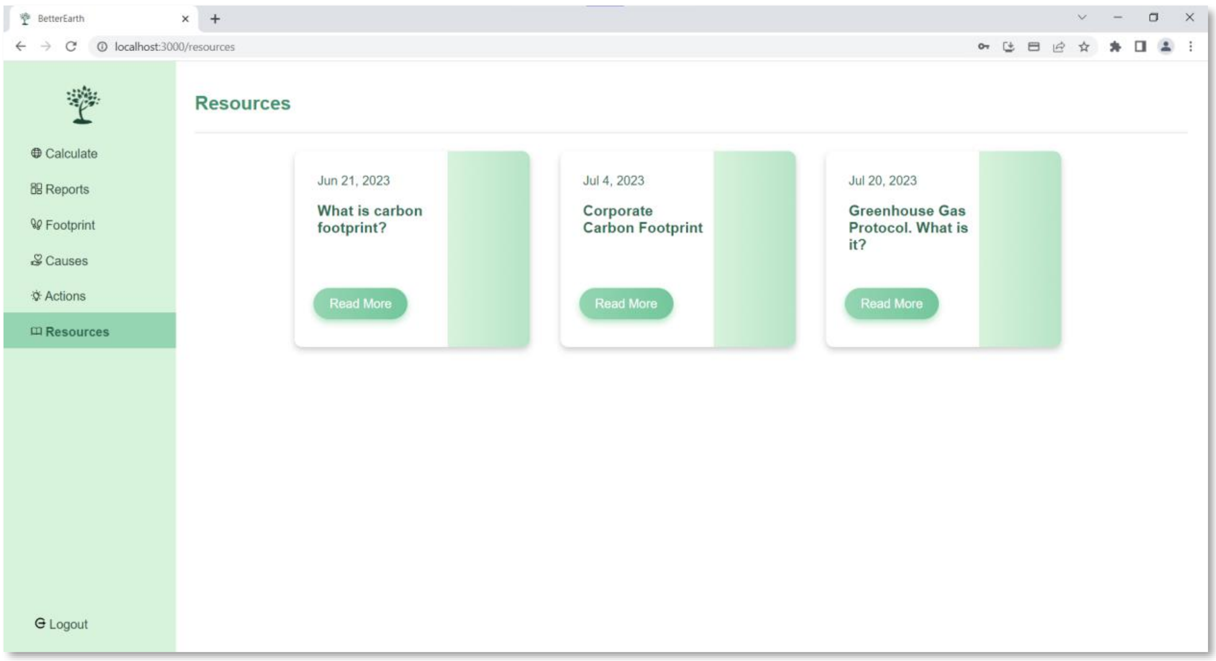
Donations

S.No.	Project Name	Donated Amount	Date of transaction
1	A Clean Solar Alternative to Kerosene Lamps, Nepal	₹100	08-06-2023
2	A Clean Solar Alternative to Kerosene Lamps, Nepal	₹500	09-06-2023
3	Give Fruit and a Greener Environment (South Sudan)	₹500	09-06-2023
4	A Clean Solar Alternative to Kerosene Lamps, Nepal	₹500	28-06-2023

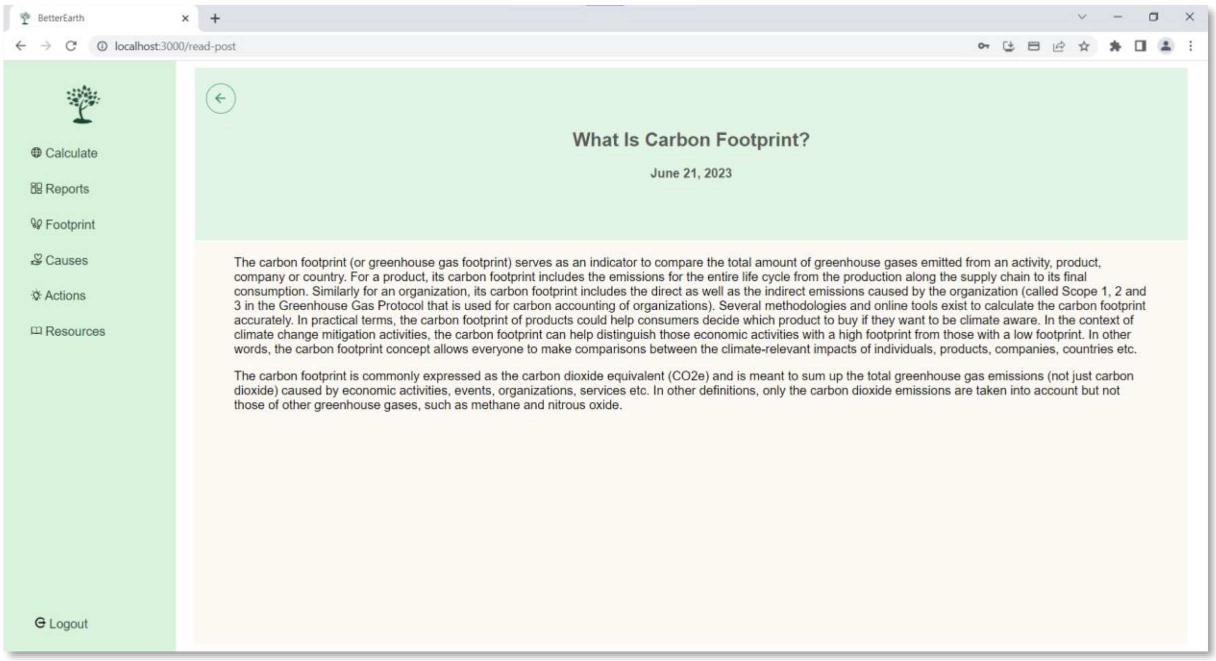
Actions



Resources



View Blog Post



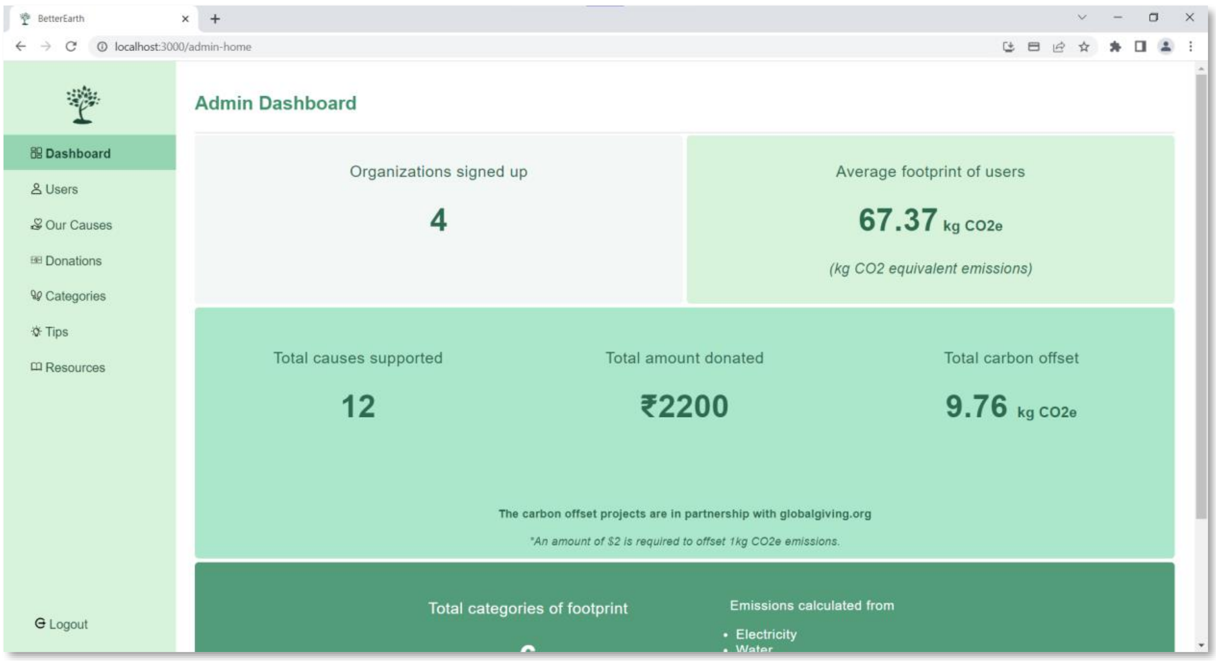
What Is Carbon Footprint?

June 21, 2023

The carbon footprint (or greenhouse gas footprint) serves as an indicator to compare the total amount of greenhouse gases emitted from an activity, product, company or country. For a product, its carbon footprint includes the emissions for the entire life cycle from the production along the supply chain to its final consumption. Similarly for an organization, its carbon footprint includes the direct as well as the indirect emissions caused by the organization (called Scope 1, 2 and 3 in the Greenhouse Gas Protocol that is used for carbon accounting of organizations). Several methodologies and online tools exist to calculate the carbon footprint accurately. In practical terms, the carbon footprint of products could help consumers decide which product to buy if they want to be climate aware. In the context of climate change mitigation activities, the carbon footprint can help distinguish those economic activities with a high footprint from those with a low footprint. In other words, the carbon footprint concept allows everyone to make comparisons between the climate-relevant impacts of individuals, products, companies, countries etc.

The carbon footprint is commonly expressed as the carbon dioxide equivalent (CO2e) and is meant to sum up the total greenhouse gas emissions (not just carbon dioxide) caused by economic activities, events, organizations, services etc. In other definitions, only the carbon dioxide emissions are taken into account but not those of other greenhouse gases, such as methane and nitrous oxide.

Admin Dashboard



Admin Dashboard

Organizations signed up
4

Average footprint of users
67.37 kg CO2e
(kg CO2 equivalent emissions)

Total causes supported
12

Total amount donated
₹2200

Total carbon offset
9.76 kg CO2e

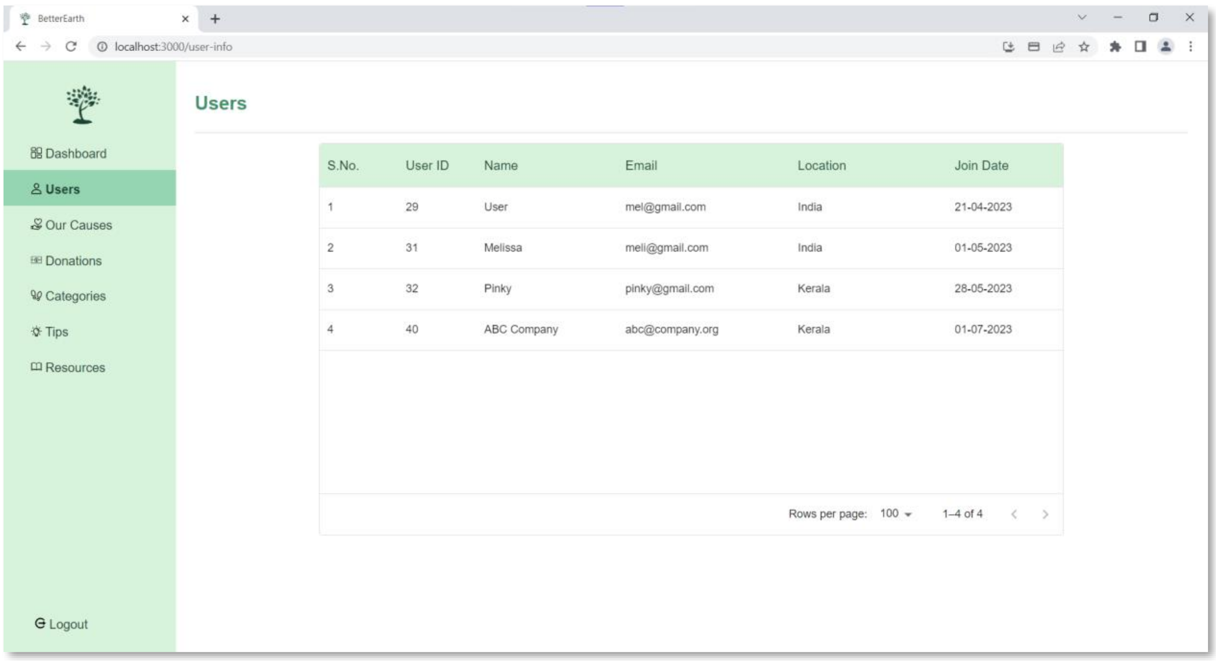
The carbon offset projects are in partnership with globalgiving.org

*An amount of $2 is required to offset 1kg CO2e emissions.

Total categories of footprint

Emissions calculated from
- Electricity
- Water

User Info

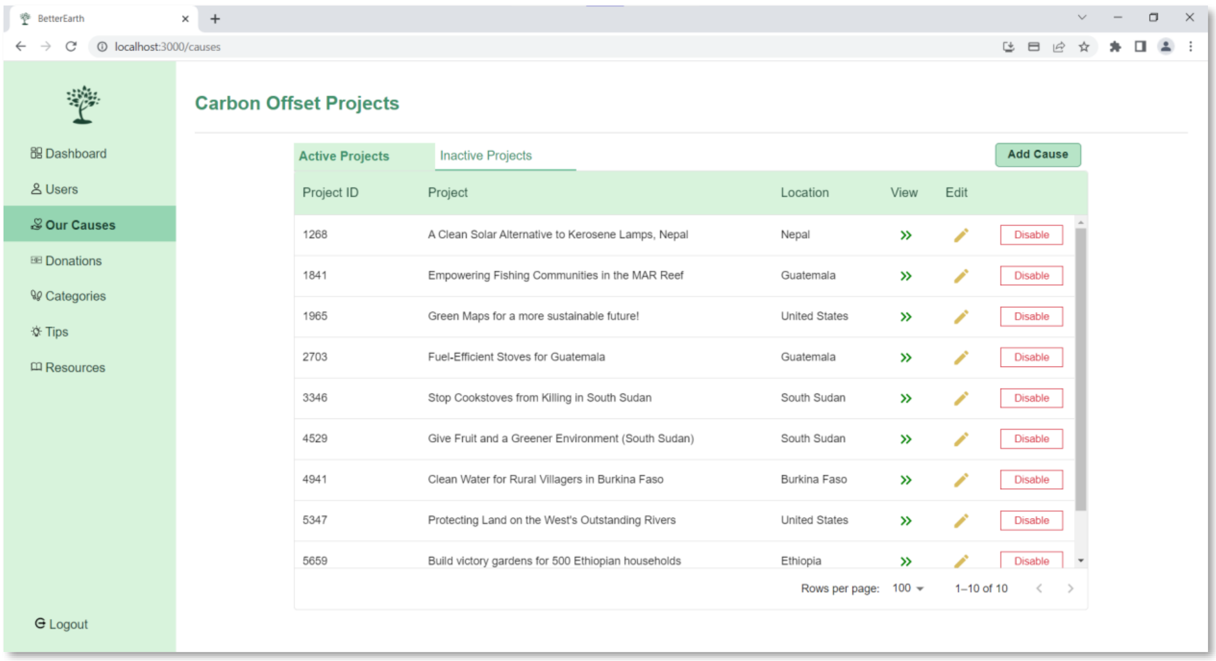


Users

S.No.	User ID	Name	Email	Location	Join Date
1	29	User	mel@gmail.com	India	21-04-2023
2	31	Melissa	meli@gmail.com	India	01-05-2023
3	32	Pinky	pinky@gmail.com	Kerala	28-05-2023
4	40	ABC Company	abc@company.org	Kerala	01-07-2023

Rows per page: 100 1–4 of 4

Carbon offset projects



BetterEarth

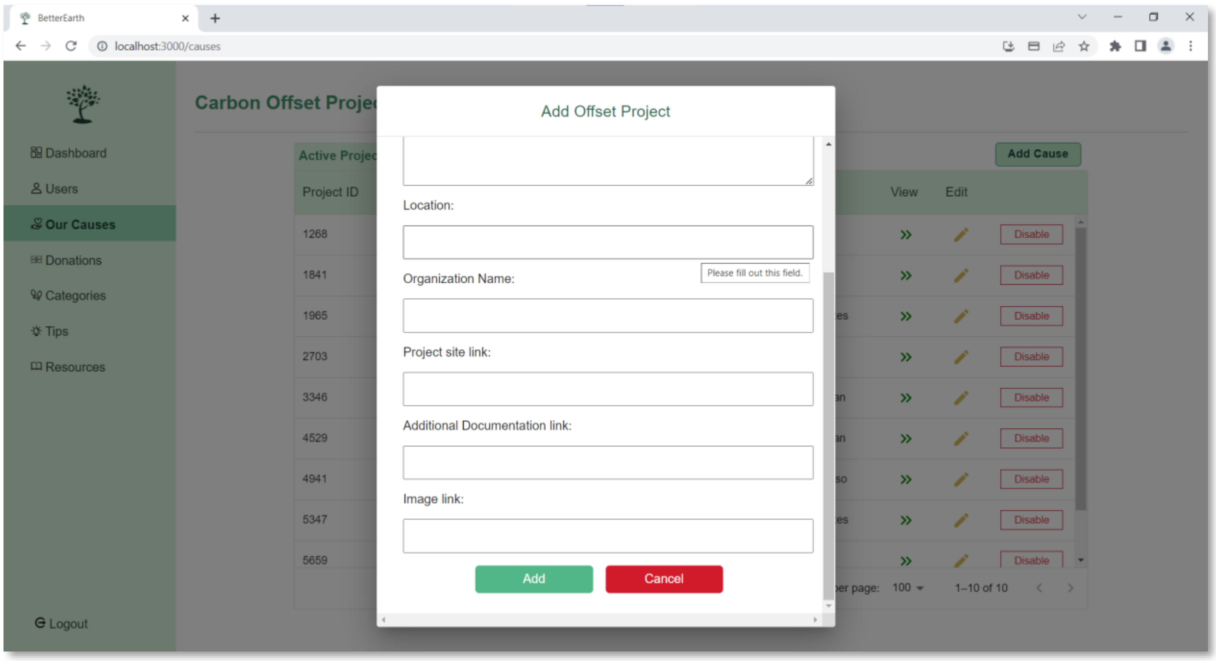
localhost:3000/causes

Carbon Offset Projects

Project ID	Project	Location	View	Edit	
1268	A Clean Solar Alternative to Kerosene Lamps, Nepal	Nepal	»	✎	Disable
1841	Empowering Fishing Communities in the MAR Reef	Guatemala	»	✎	Disable
1965	Green Maps for a more sustainable future!	United States	»	✎	Disable
2703	Fuel-Efficient Stoves for Guatemala	Guatemala	»	✎	Disable
3346	Stop Cookstoves from Killing in South Sudan	South Sudan	»	✎	Disable
4529	Give Fruit and a Greener Environment (South Sudan)	South Sudan	»	✎	Disable
4941	Clean Water for Rural Villagers in Burkina Faso	Burkina Faso	»	✎	Disable
5347	Protecting Land on the West's Outstanding Rivers	United States	»	✎	Disable
5659	Build victory gardens for 500 Ethiopian households	Ethiopia	»	✎	Disable

Rows per page: 100 1–10 of 10

Add Offset Project



Add Offset Project

Location:

Organization Name:

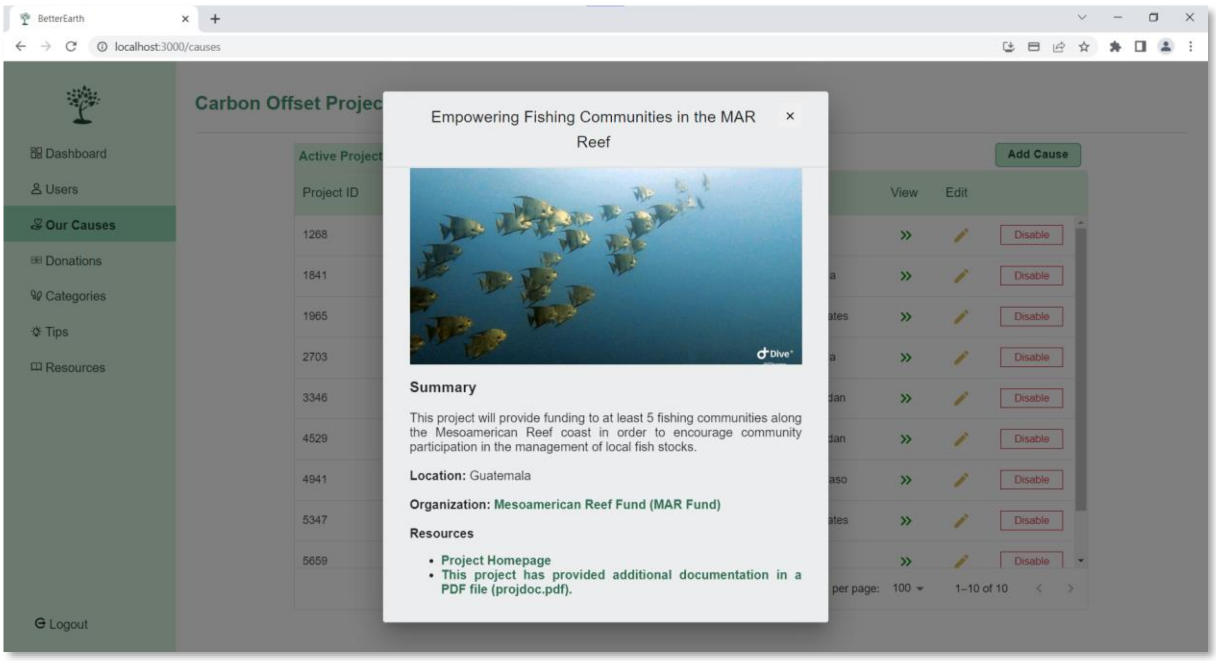
Please fill out this field.

Project site link:

Additional Documentation link:

Image link:

View Offset Project Details



Empowering Fishing Communities in the MAR Reef

Summary

This project will provide funding to at least 5 fishing communities along the Mesoamerican Reef coast in order to encourage community participation in the management of local fish stocks.

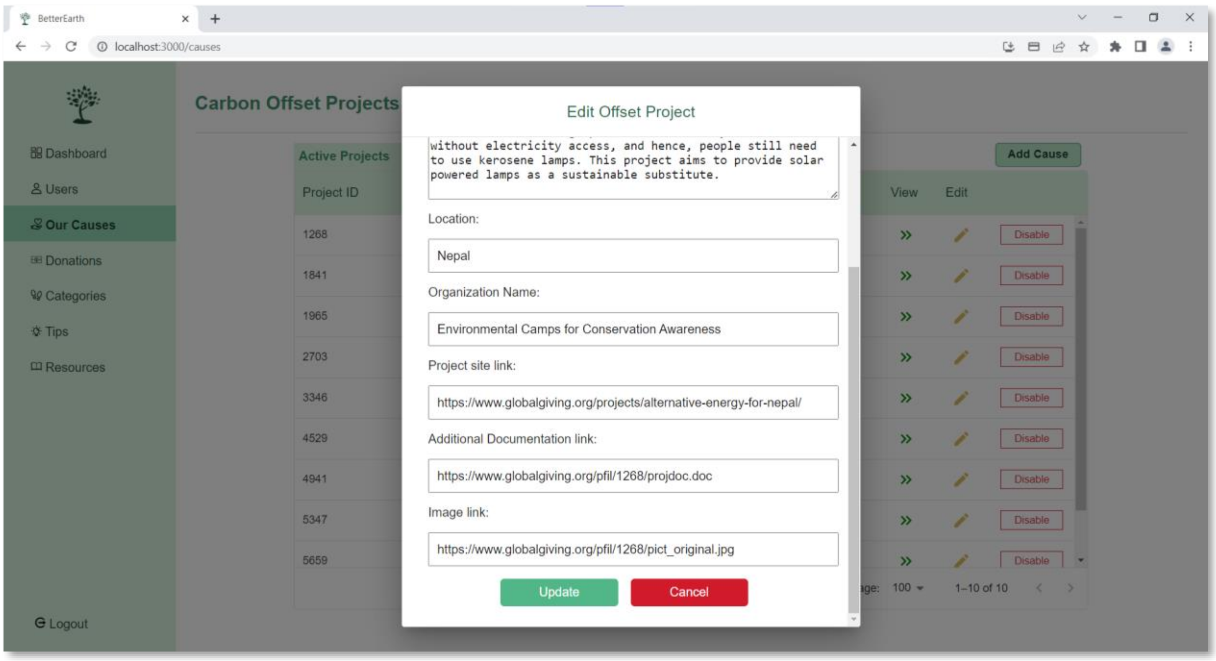
Location: Guatemala

Organization: **Mesoamerican Reef Fund (MAR Fund)**

Resources

- **Project Homepage**
- **This project has provided additional documentation in a PDF file (projdoc.pdf).**

Edit Project Details



Edit Offset Project

```
without electricity access, and hence, people still need
to use kerosene lamps. This project aims to provide solar
powered lamps as a sustainable substitute.
```

Location: Nepal

Organization Name: Environmental Camps for Conservation Awareness

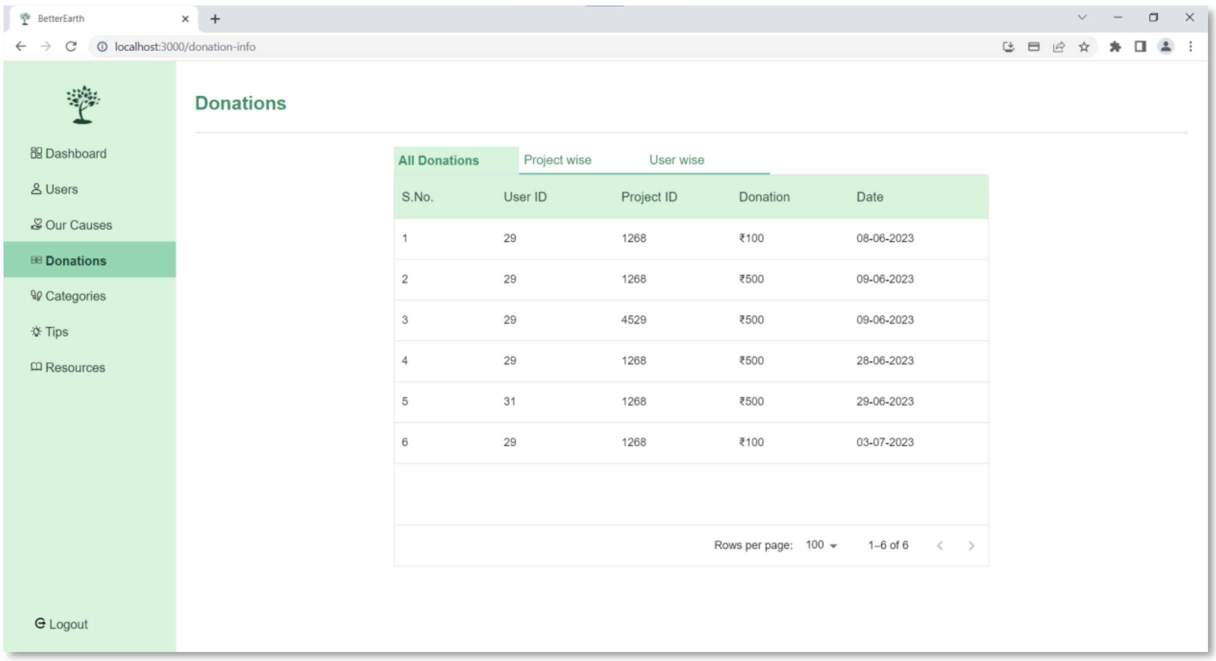
Project site link: https://www.globalgiving.org/projects/alternative-energy-for-nepal/

Additional Documentation link: https://www.globalgiving.org/pfil/1268/projdoc.doc

Image link: https://www.globalgiving.org/pfil/1268/pict_original.jpg

Update Cancel

Donations Info



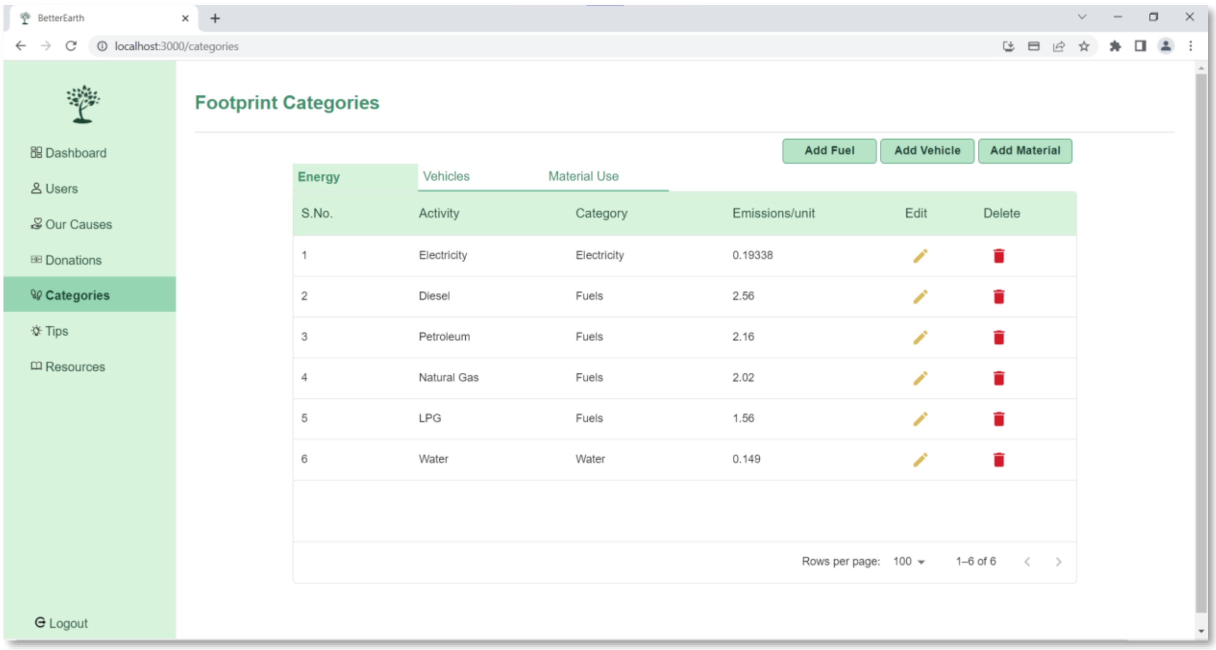
Donations

All Donations | Project wise | User wise

S.No.	User ID	Project ID	Donation	Date
1	29	1268	₹100	08-06-2023
2	29	1268	₹500	09-06-2023
3	29	4529	₹500	09-06-2023
4	29	1268	₹500	28-06-2023
5	31	1268	₹500	29-06-2023
6	29	1268	₹100	03-07-2023

Rows per page: 100 1–6 of 6

Footprint Categories



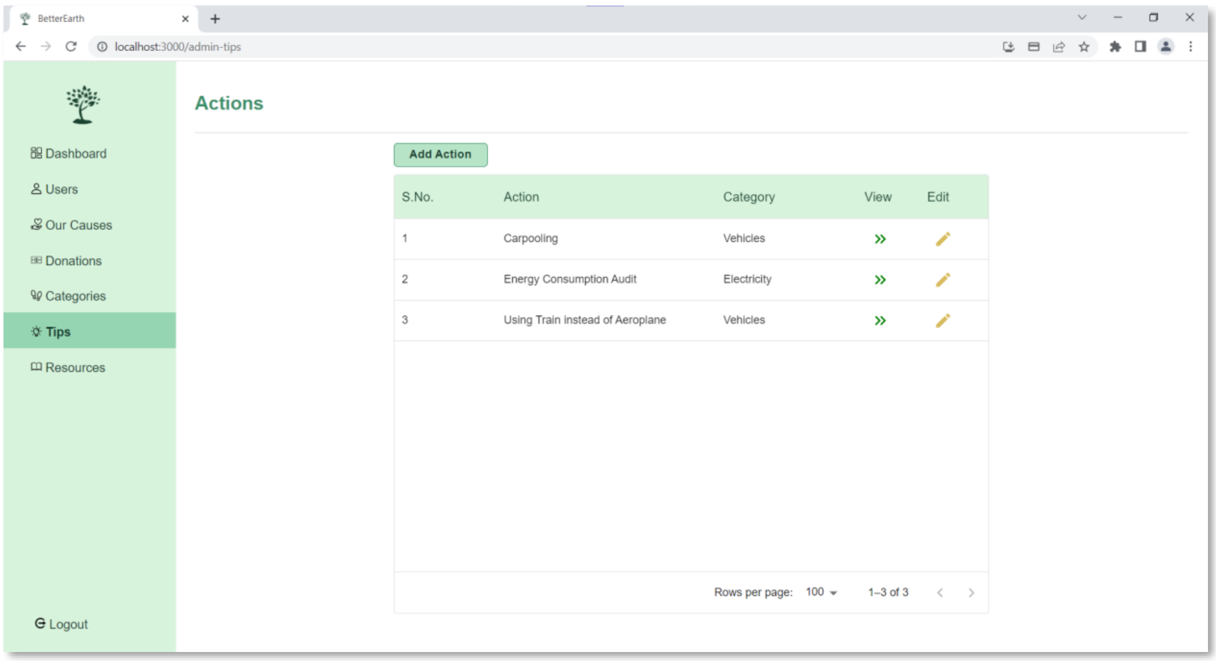
Footprint Categories

Energy | Vehicles | Material Use

S.No.	Activity	Category	Emissions/unit	Edit	Delete
1	Electricity	Electricity	0.19338	✎	🗑
2	Diesel	Fuels	2.56	✎	🗑
3	Petroleum	Fuels	2.16	✎	🗑
4	Natural Gas	Fuels	2.02	✎	🗑
5	LPG	Fuels	1.56	✎	🗑
6	Water	Water	0.149	✎	🗑

Rows per page: 100 1–6 of 6

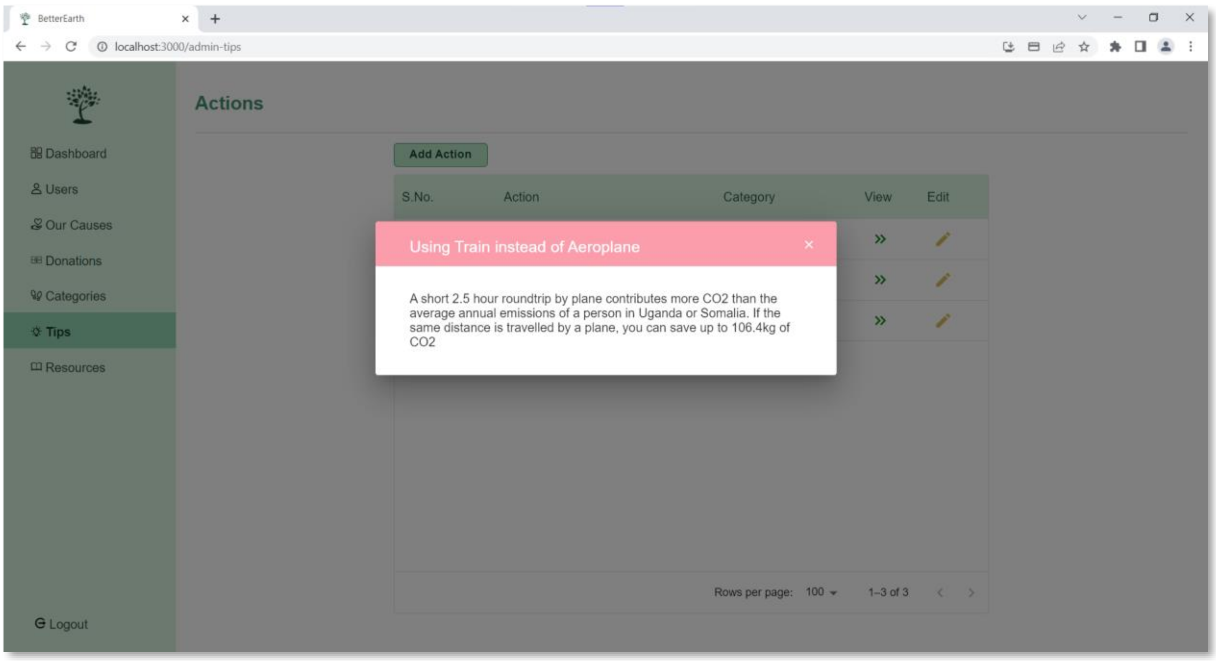
Actions



Actions

S.No.	Action	Category	View	Edit
1	Carpooling	Vehicles	»	✎
2	Energy Consumption Audit	Electricity	»	✎
3	Using Train instead of Aeroplane	Vehicles	»	✎

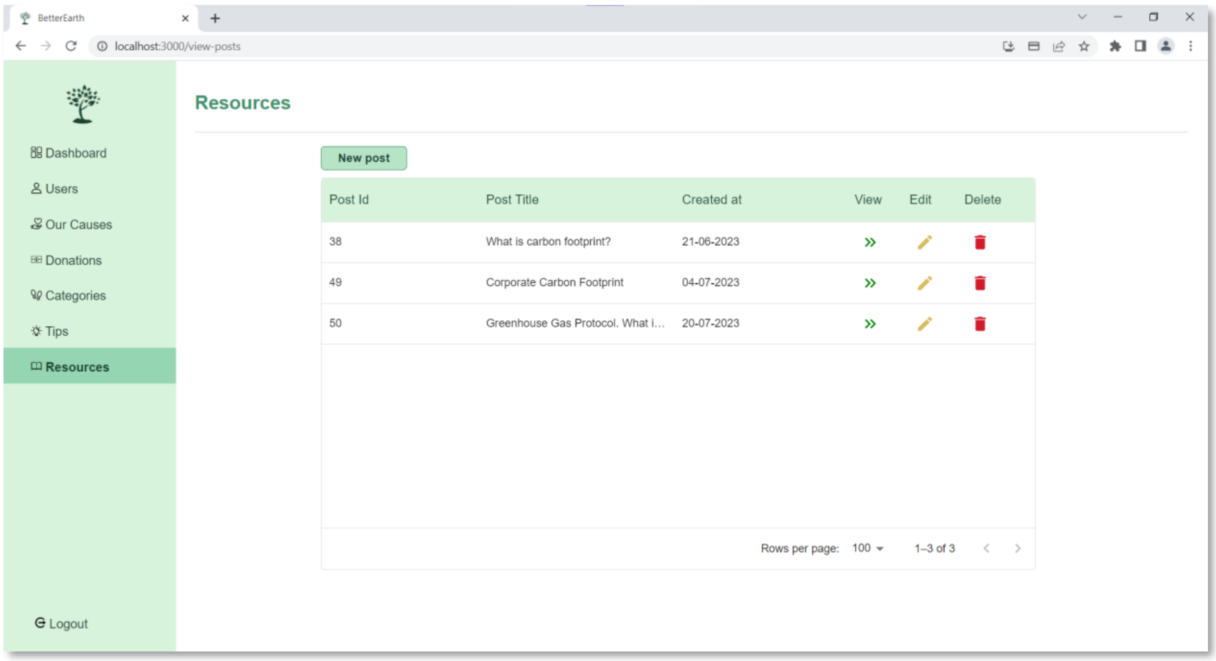
View Action



Using Train instead of Aeroplane

A short 2.5 hour roundtrip by plane contributes more CO2 than the average annual emissions of a person in Uganda or Somalia. If the same distance is travelled by a plane, you can save up to 106.4kg of CO2

Resources



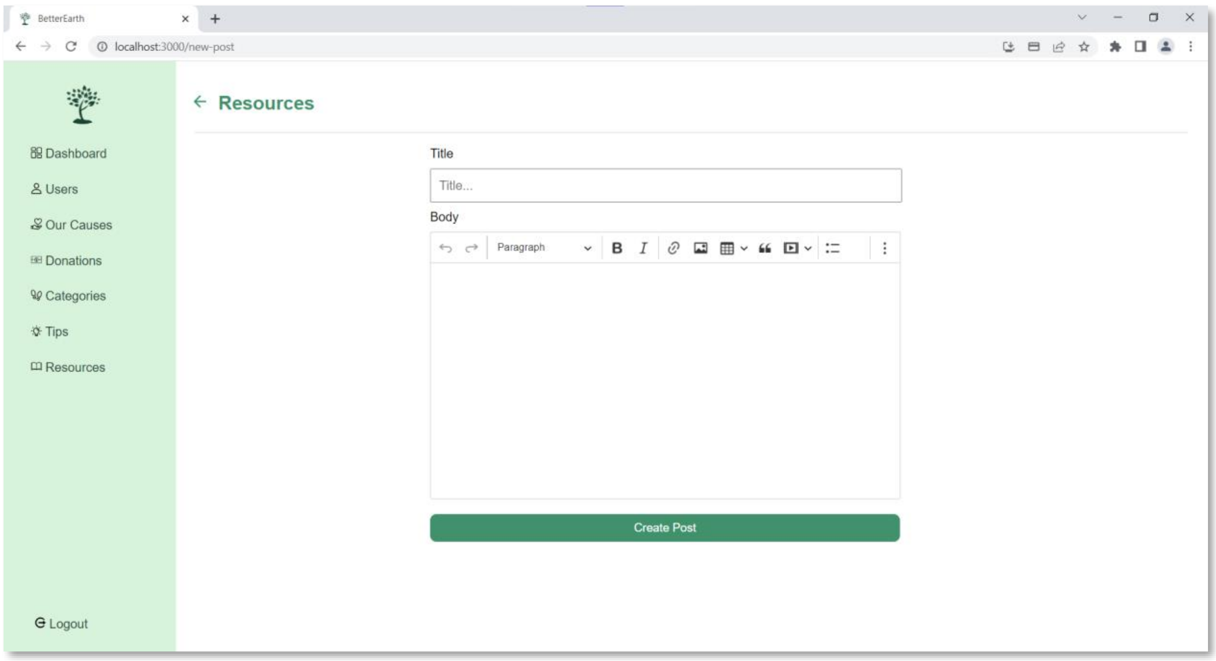
Resources

New post

Post Id	Post Title	Created at	View	Edit	Delete
38	What is carbon footprint?	21-06-2023	»	✎	🗑
49	Corporate Carbon Footprint	04-07-2023	»	✎	🗑
50	Greenhouse Gas Protocol. What i…	20-07-2023	»	✎	🗑

Rows per page: 100 1–3 of 3

New Post



Title

Title...

Body

11.4. SAMPLE PROJECT CODE

App.js

```
import React, { useEffect, useState } from "react";
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import { useCookies } from "react-cookie";
import PrivateRoutes from "./pages/components/ProtectedRoute.js";
import { AuthContext } from "./context/Context.js";

// * ------PAGES------ *//
import { Main } from "./pages/main-page/main.js";
import { Login } from './pages/login-page/login.js';
import { SelectUser } from "./pages/login-page/selectUser.js";
import { Signup } from "./pages/signup-page/signup.js";
import { Home } from "./pages/home-page/dashboard.js";
import { About } from "./pages/main-page/about.js";

import { Footprint } from "./pages/your-footprint-page/footprint.js";
import { CategoryWater } from "./pages/your-footprint-page/category-water.js";
import { CategoryElectricity } from "./pages/your-footprint-page/category-electricity.js";
import { CategoryFuels } from "./pages/your-footprint-page/category-fuels.js";
import { CategoryVehicles } from "./pages/your-footprint-page/category-vehicles.js";
import { CategoryMaterials } from "./pages/your-footprint-page/category-materials.js";
import { DisplayFootprint } from "./pages/your-footprint-page/display.js";
import { ImpactTracker } from "./pages/impact-tracker-page/impact.js";
import { Tips } from "./pages/tips-page/tips.js";

import { Leaderboard } from "./pages/leaderboard-page/leaderboard.js";
import { Resources } from "./pages/resources-page/resources.js";

import { Charities } from "./pages/charities-page/charities.js";
import { Donation } from "./pages/charities-page/donation.js";
import { Success } from "./pages/charities-page/success.js";
import { MyDonations } from "./pages/donations-made.js";

import { AdminHome } from "./admin/adminHome.js";
import { Users } from "./admin/users.js";

import { ReadPost } from "./pages/resources-page/read-post.js";
import { Blog } from "./admin/blog/blog.js";
import { ViewPosts } from "./admin/blog/view-post.js";
import { ReadPostAdmin } from "./admin/blog/read-post-admin.js";
import { OffsetProjects } from "./admin/projects/view-projects.js";
import { DonationInfo } from "./admin/donations/donations.js";
import { AdminLeaderboard } from "./admin/admin-leaderboard.js";
import { AdminTips } from "./admin/tips/view-tips.js";
import { ViewTip } from "./pages/tips-page/view-tip.js";
import { CategoryBusinessTrips } from "./pages/your-footprint-page/category-business.js";
import { Categories } from "./admin/footprint/category.js";
```

```
function App() {

  const [authState, setAuthState] = useState(false); //authState is a boolean
  const [cookies, ] = useCookies(["access_token"]);

  useEffect(() => {
    if(cookies["access_token"] != '')
    {
      console.log("hey"+cookies["access_token"]);
      setAuthState(true);
    }
    else if(cookies["access_token"] == '')
    {
      console.log("bye")
      setAuthState(false);
    }


  }, []);

  return (
    <AuthContext.Provider value={{authState, setAuthState}}>
    <div className="App">
      <Router>
        <Routes>
         <Route path="/" element={<Main />} />
         <Route path="/about" element={<About />} />
          <Route path="/login" element={<Login />} />
          <Route path="/signup" element={<Signup />} />


          <Route element={<PrivateRoutes /> }>
            <Route path="/select-user" element={<SelectUser/> } />
            <Route path="/home" element={<Home />} />
            <Route path="/your-footprint" element={<Footprint />} />
            <Route path="/impact-tracker" element={<ImpactTracker />} />

            <Route path="/tips" element={<Tips />} />

            <Route path="/charities" element={<Charities />} />
            <Route path="/donate" element={<Donation />} />
            <Route path="/success" element={<Success />} />
            <Route path="/my-donations" element={<MyDonations />} />

            <Route path="/leaderboard" element={<Leaderboard />} />
            <Route path="/resources" element={<Resources />} />
            <Route path="/read-post" element={<ReadPost />} />

            <Route path="/category-water" element={<CategoryWater />} />
            <Route path="/category-electricity" element={<CategoryElectricity />} />
            <Route path="/category-fuels" element={<CategoryFuels />} />
            <Route path="/category-vehicles" element={<CategoryVehicles />} />
            <Route path="/category-materials" element={<CategoryMaterials />} />
            <Route path="/category-business-trips" element={<CategoryBusinessTrips />} />
            <Route path="/display-footprint" element={<DisplayFootprint />} />

            <Route path="/admin-home" element={<AdminHome/>}/>
            <Route path="/user-info" element={<Users />} />
            <Route path="/blog" element={<Blog />} />
            <Route path="/view-posts" element={<ViewPosts />} />
            <Route path="/read-post-admin" element={<ReadPostAdmin />} />
            <Route path="/causes" element={<OffsetProjects />} />
            <Route path="/donation-info" element={<DonationInfo />} />
            <Route path="/categories" element={<Categories />} />
            <Route path="/admin-tips" element={<AdminTips />} />
            <Route path="/admin-leaderboard" element={<AdminLeaderboard />} />
          </Route>

        </Routes>
      </Router>
    </div>
    </AuthContext.Provider>
  );
}

export default App;
```

Login.js

```
import React, { useContext, useState } from "react";
import { Navbar } from "../navbar";
import "./login-styles.css";
import Axios from "axios";
import { Link, useNavigate } from "react-router-dom";
import { useCookies } from 'react-cookie';
import { AuthContext } from "../../context/Context";
import { Footer } from "../footer";
import image from "../../assets/images/pic1.jpg"
import { useForm } from "react-hook-form";
import { yupResolver} from '@hookform/resolvers/yup'
import * as yup from 'yup';

export const Login = () => {
  const[email, setEmail] = useState("");
  const[password, setPassword] = useState("");
  const {setAuthState} = useContext(AuthContext);

  const [error, setError] = useState("");

  const schema = yup.object().shape({
      email: yup.string().email("Please enter valid email").required("Please enter organization
email"),
      password: yup.string().min(8, "Password must be atleast 8
characters").max(20).required("Please enter password"),
  });
  const { register, handleSubmit, formState: {errors} } = useForm({
      resolver: yupResolver(schema)
  });

  const [ ,setCookies] = useCookies(["access_token"])
  const navigate = useNavigate();

  const login = async (event) => {
      const response = await Axios.post("/login",{
          email: email,
          password: password
      });
      if(!response.data.message){
          setCookies("access_token", response.data.token);
          window.localStorage.setItem("user", JSON.stringify(response.data.details));
          setAuthState(true);
          // console.log(response.data);
          navigate("/select-user");
      }

      else setError(response.data.message);

  };

  return (
      <>
          <Navbar />

          <div className="login-page">
              <div className="login-pic">
                  <img src={image} alt="environment"/>
              </div>
              <div className="login">
                  <h2>Welcome back!</h2>
                  <form onSubmit={handleSubmit(login)}>
                      <div className="input-container">

                          <span className="error-msg">{errors.email?.message}</span>
                          <input placeholder="Email address" {...register("email")}
                          onChange={(event)=>{
                              setEmail(event.target.value);
                          }}/>

                          <span className="error-msg">{errors.password?.message}</span>
                          <input type="password" placeholder="Password" {...register("password")}
                        onChange={(event)=>{
                        setPassword(event.target.value);
                      }}
                      />

                      <button>Login</button>
                      <span className="error-msg" style={{margin: "auto", marginTop:
"0px"}}>{error? error: null}</span>
                    </div>
                    <h4>No account yet? <Link to="/signup">Signup</Link></h4>
                  </form>
                </div>
              </div>
              <Footer />

          </>
    );
}
```

Server.js

```
// * importing the express module and creating the express object
  import express from "express"
  import bodyParser from "body-parser";
  import cookieParser from "cookie-parser";
  import cors from "cors";
  import { signup, login } from "./controllers/auth.js";
  import { checkoutSession } from "./controllers/checkout-session.js";
  import { userInfo } from "./controllers/users.js";
  import { createPost, viewPost, displayPost, deletePost } from "./controllers/blog.js";

  import { DisableCause, EnableCause, addCause, addCauses, addTransaction,
getAllTransactions, getProject, getProjectWiseTransactions, getTransactions, updateCause,
viewActiveCauses, viewCause, viewInactiveCauses,} from "./controllers/offset-projects.js";

  import { getEmissions, getFootprint, getFootprintCategoryWise, getCategory, getFuels,
addFootprint, getMaterialActivity, getMaterial, addMaterialFootprint,
getFootprintCategoryWise2, getVehicles, getVehicleSize, getFuelType, addVehicleFootprint,
getFuelDetails, getVehicleDetails, getMaterialDetails, getTotalFootprint } from
"./controllers/calculations.js";

import { adminDashboard } from "./controllers/admin.js";
import { adminLeaderboard } from "./controllers/leaderboard.js";
import { addTip, getTips, updateTip, viewTip, viewTips } from "./controllers/tips.js";

import { addEmissionsFuel, addEmissionsMaterial, addEmissionsVehicle, deleteFuel,
deleteMaterial, deleteVehicle, getEmissionsInfo, getMaterialsInfo, getVehiclesInfo } from
"./controllers/emissions.js";
import { getOffsetReport } from "./controllers/userReports.js";

  const app = express();
```

```
    const port = 3002;

    app.use(cors({
        origin: ("http://localhost:3000"),
        methods: (["PUT", "POST", "DELETE"]),
        credentials: true
    }));

    app.use(bodyParser.urlencoded({extended: true}));
    app.use(express.json());
    app.use(cookieParser());


// * route definition: the path and the callback function
    app.post("/api/signup", signup);
    app.post("/api/login", login)

    app.post('/api/checkout-session', checkoutSession);

// * Users page - admin
    app.get('/api/user-info', userInfo);

// *Resources page - admin
    app.post('/api/create-blog', createPost);
    app.get('/api/view-post', viewPost);
    app.get('/api/display-post/:id', displayPost);
    app.delete('/api/delete-post/:postId', deletePost);

// * Carbon offset projects - admin
    app.post('/api/causes', addCauses);
    app.get('/api/view-active-causes', viewActiveCauses);
    app.get('/api/view-inactive-causes', viewInactiveCauses);

app.get('/api/view-cause/:id', viewCause);
    app.get('/api/get-project/:id', getProject);
    app.put('/api/disable-cause', DisableCause);
    app.put('/api/enable-cause', EnableCause);

    app.get('/api/get-transactions/:userid', getTransactions);
    app.get('/api/get-all-transactions', getAllTransactions);
    app.get('/api/get-transactions-total-project', getProjectWiseTransactions);
    app.post('/api/add-transaction', addTransaction);

    app.post('/api/add-cause', addCause);
    app.put('/api/update-cause', updateCause);

// *Carbon footprint calculations
    app.get('/api/energy-emissions/:activity', getEmissions);

    app.get('/api/user-footprint/:userid', getFootprint);
    app.get('/api/categorywise-footprint/:userid', getFootprintCategoryWise); //returns json
    app.get('/api/categorywise-footprint2/:userid', getFootprintCategoryWise2); //returns array
    app.get('/api/category', getCategory);
    app.get('/api/get-total-footprint/:userid', getTotalFootprint);

    app.get('/api/fuels', getFuels);
    app.post('/api/add-footprint', addFootprint);

    app.get('/api/material-activity', getMaterialActivity);
    app.get('/api/get-materials/:selectedActivity', getMaterial);
    app.post('/api/add-material-footprint', addMaterialFootprint);

    app.get('/api/get-vehicles', getVehicles);
    app.get('/api/get-vehicle-size', getVehicleSize);
    app.get('/api/get-fuel-type', getFuelType);
    app.post('/api/add-vehicle-footprint', addVehicleFootprint);

    app.get('/api/fuel-details/:id', getFuelDetails);
    app.get('/api/vehicle-details/:id', getVehicleDetails);
    app.get('/api/material-details/:id', getMaterialDetails);

// * User Reports
    app.get('/api/get-total-offset/:userid', getOffsetReport);

// * Admin Dashboard
    app.get("/api/admin-dashboard", adminDashboard);

// * Admin Leaderboard
    app.get("/api/admin-leaderboard", adminLeaderboard);


// * Tips
    app.post("/api/add-tip", addTip);
    app.get("/api/view-tips", viewTips);
    app.get("/api/view-tip/:id", viewTip);
    app.put("/api/update-tip", updateTip);
    app.get("/api/get-tips", getTips);

// * Category Emissions in Admin
    app.get("/api/emissions-info", getEmissionsInfo);
    app.get("/api/vehicles-info", getVehiclesInfo);
    app.get("/api/materials-info", getMaterialsInfo);

    app.post("/api/add-fuel", addEmissionsFuel);
    app.post("/api/add-vehicle", addEmissionsVehicle);
    app.post("/api/add-material", addEmissionsMaterial);

    app.delete('/api/delete-fuel/:id', deleteFuel);
    app.delete('/api/delete-vehicle/:id', deleteVehicle);
    app.delete('/api/delete-material/:id', deleteMaterial);
```

```
// * the server is running on the specified port
   app.listen(port, () => {
      console.log(`The server is running on port ${port}`);
   });
```

Calculation.js

```
import { db } from '../db.js';

export const getEmissions = (req, res) => {
   console.log(req.params.activity)
   db.query("Select emissions_per_unit, activity_id from energy_usage where activity_item =
? ",
      req.params.activity, (err, result) => {
         // console.log(result);
         res.send(result);
      }
   )
}


export const getFootprint = (req, res) => {
   db.query("Select activity_id, calculated_value from user_footprint where userid = ?",
req.params.userid,
   (err,result) => {
      // console.log(result)
      res.send(result);
   })
}

export const getFootprintCategoryWise = (req, res) => {

   var electricity = 0;
   var water = 0;
   var fuels = 0;
   var vehicles = 0;
   var materials = 0;

   db.query("Select sum(user_footprint.calculated_value) AS total,
energy_usage.category_id, footprint_category.category_name \
    from user_footprint inner join energy_usage on user_footprint.activity_id =
energy_usage.activity_id inner join footprint_category \
    on footprint_category.category_id = energy_usage.category_id where
user_footprint.userid = ? group by category_id order by category_id ",
    req.params.userid,
    (err, result) => {

      result.forEach(x => {
         if(x.category_id == 1) electricity = x.total;
```

```
            if(x.category_id == 2) water = x.total;
            if(x.category_id == 3) fuels = x.total;
            if(x.category_id == 4) vehicles = x.total;
            if(x.category_id == 5) materials = x.total;
        })
        const total = electricity + water + fuels + vehicles + materials;

        // console.log(result)
        res.send({
            "electricity": electricity,
            "water": water,
            "fuels": fuels,
            "vehicles": vehicles,
            "materials": materials,
            "total": total,
        });

    })

}

//total footprint by category of user
export const getFootprintCategoryWise2 = (req, res) => {

    db.query("Select sum(user_footprint.calculated_value) AS total,\
             energy_usage.category_id, footprint_category.category_name \
            from user_footprint inner join energy_usage on user_footprint.activity_id = 
energy_usage.activity_id inner join footprint_category \
            on footprint_category.category_id = energy_usage.category_id where 
user_footprint.userid = ? group by category_id order by category_id ",
            req.params.userid,
      (err, result) => {
        // console.log(result)
        res.send(result);

    })

}

export const getTotalFootprint = (req, res) => {
    db.query("Select sum(calculated_value) as total from user_footprint where userid = ?", 
req.params.userid, (err, result) => {
        if(err) console.log(err);
        else {
            const total = result[0].total === null ? 0 : result[0].total;

           res.send({total});
        }
    })
}
```

```
export const getCategory = (req, res) => {
   db.query("Select * From footprint_category", (err,result) => {
      res.send(result);
   })
}

export const getFuels = (req, res) => {
   db.query("Select * from energy_usage where category_id = 3", (err, result) => {
      res.send(result);
   })
}

export const addFootprint = (req, res) => {
   const activity = req.body.activity;
   const amount = req.body.amount;
   const userid = req.body.userid;

   db.query("Select activity_id, emissions_per_unit from energy_usage where
activity_item=?",activity, (err, result) => {
      // console.log(result[0].emissions_per_unit);
      const activityid = result[0].activity_id;
      const emissions = result[0].emissions_per_unit;
      const total = amount * emissions;
      // console.log(total)
      if(activityid ==  1 || activityid == 2)
      {
         db.query("Select * from user_footprint where userid=? and activity_id=? and
YEAR(calculated_date)=YEAR(CURDATE())",
               [userid,activityid], (err, result) => {
                  if(result.length>0) {res.send({"message": "Already exists"}) }
                  else if(result.length===0){
                     db.query("Insert into user_footprint(userid, activity_id, calculated_value,
calculated_date) values(?,?,?,CURDATE())",
                     [userid,activityid,total], (err, result) => {
                        console.log("Inserted successfully")
                     })
                  }
               })
      }
      else{
         db.query("Insert into user_footprint(userid, activity_id, calculated_value,
calculated_date) values (?,?,?,CURDATE())",
         [userid, activityid, total], (err,result) => {
         console.log("Inserted successfully");
      })
      }

   } )
```

```
}

export const getMaterialActivity = (req, res) => {
   db.query("Select DISTINCT material_activity from materials", (err, result) => {
      res.send(result);
   })
}

export const getMaterial = (req, res) => {

   const activity = req.params.selectedActivity;

   db.query("Select energy_usage.activity_item from energy_usage inner join materials \
   on energy_usage.activity_id = materials.material_usage_id where \
   materials.material_activity = ? ", activity, (err, result) => {
     res.send(result);
   })
}

export const addMaterialFootprint = (req, res) => {
   const userid = req.body.userid;
   const material_activity = req.body.material_activity;
   const material = req.body.material;
   const amount = req.body.amount;

   db.query("Select energy_usage.emissions_per_unit, energy_usage.activity_id from
energy_usage inner join materials \
   on energy_usage.activity_id = materials.material_usage_id where
materials.material_activity = ? \
   and energy_usage.activity_item = ? ", [material_activity, material], (err, result) => {

      const emissions = result[0].emissions_per_unit;
      const activityid = result[0].activity_id;
      const total = amount * emissions;

      db.query("Insert into user_footprint(userid, activity_id, calculated_value,
calculated_date) values (?,?,?,CURDATE())",
      [userid, activityid, total], (err, result) => {
         console.log("Inserted Successfully");
      })
   })
}

export const getVehicles = (req, res) => {
   db.query("Select distinct vehicle_type as type from transportation", (err, result) =>{
      res.send(result)
   })
}

export const getVehicleSize = (req, res) => {
    db.query("Select distinct vehicle_size as size from transportation", (err, result) =>{
        res.send(result)
    })
}

export const getFuelType = (req, res) => {
    db.query("Select distinct fuel_type as fuel from transportation", (err, result) =>{
        res.send(result)
    })
}

export const addVehicleFootprint = (req, res) => {

    const userid = req.body.userid;
    const type = req.body.vehicle;
    const size = req.body.size;
    const fuel = req.body.fuel;
    const amount = req.body.amount;
    const mileage = req.body.mileage;

    db.query("Select transportation.vehicle_id, energy_usage.emissions_per_unit from \
    transportation inner join energy_usage on transportation.vehicle_id =
energy_usage.activity_id \
    where transportation.vehicle_type=? and transportation.vehicle_size=? and
transportation.fuel_type=? ",
    [type, size, fuel], (err, result) => {

        const id = result[0].vehicle_id;
        const emissions = result[0].emissions_per_unit;
        const total = amount * mileage * emissions;

        db.query("Insert into user_footprint(userid, activity_id, calculated_value,
calculated_date)\
        values(?,?,?,CURDATE())", [userid,id,total], (err,result) => {
            console.log("Inserted Successfully");
        })

    })
}

export const getFuelDetails = (req, res) => {

    db.query("Select ROW_NUMBER() OVER (ORDER BY energy_usage.activity_item) as
sno, sum(user_footprint.calculated_value) as sum, energy_usage.activity_item from
user_footprint\
    inner join energy_usage on user_footprint.activity_id = energy_usage.activity_id\
     where energy_usage.category_id = 3 and user_footprint.userid = ? group by
energy_usage.activity_item", req.params.id, (err, result) => {
        res.send(result);
     })
```

```
}

export const getVehicleDetails = (req, res) => {
   db.query("Select ROW_NUMBER() OVER (ORDER BY energy_usage.activity_item) as
sno, sum(user_footprint.calculated_value) as sum, transportation.vehicle_type, \
   transportation.vehicle_size, transportation.fuel_type from user_footprint inner join
energy_usage on user_footprint.activity_id = energy_usage.activity_id \
   inner join transportation on transportation.vehicle_id = energy_usage.activity_id \
   where energy_usage.category_id = 4 and user_footprint.userid = ? group by
energy_usage.activity_item", req.params.id, (err, result) => {

      res.send(result);
    })
}

export const getMaterialDetails = (req, res) => {
   db.query("Select ROW_NUMBER() OVER (ORDER BY energy_usage.activity_item) as
sno, energy_usage.activity_item as item, \
   sum(user_footprint.calculated_value) as sum, materials.material_activity as type \
   from user_footprint inner join energy_usage on user_footprint.activity_id =
energy_usage.activity_id \
   inner join materials on materials.material_usage_id = energy_usage.activity_id \
   where energy_usage.category_id = 5 and user_footprint.userid = ? group by
energy_usage.activity_id", req.params.id, (err, result) => {
      // console.log(result);
      res.send(result);
    })
}
```