GreenNudge: Developing a Sustainable Digital Platform for SMEs in the Textile Industry

Cathy Fu*, Emma Wang*, Hatty He*, Yuning Mu*, Yvonne Coady*†
*Department of Computer Science, Northeastern University, Vancouver, BC, Canada
†Department of Computer Science, University of Victoria, Victoria, BC, Canada

***Abstract*—The GreenNudge project aims to develop a sustainable digital platform connecting Small and Medium-sized Enterprises (SMEs) in the textile industry to greener supply chains. The platform addresses the pressing environmental challenges faced by the textile sector, including high greenhouse gas emissions and waste. Leveraging AWS cloud services, green computing principles, and energy-efficient algorithms, the platform minimizes its digital footprint while supporting SMEs in transitioning to sustainable operations. Key features include supplier matching, carbon footprint monitoring, and real-time energy usage analysis. By integrating caching techniques and AI-driven monitoring systems, GreenNudge ensures scalability and operational efficiency with minimal environmental impact. This project serves as a blueprint for incorporating sustainability into digital infrastructure, aligning with global decarbonization goals and fostering greener business practices in the textile industry.**

I. Introduction to the Problem

A. Background

As companies aim to meet global decarbonization targets like the 2050 Paris Agreement, which calls for net-zero emissions by 2050, the textile sector remains a significant contributor to environmental pollution. The industry generates substantial greenhouse gas emissions, water pollution, and waste, underscoring the urgent need for sustainable practices. While many businesses are committing to greener operations, transitioning to sustainable supply chains is complex. Small and Medium-sized Enterprises (SMEs) face distinct challenges, including limited access to verified eco-friendly suppliers, insufficient tools for assessing environmental impact, and a lack of resources to support their shift to sustainable alternatives. Additionally, despite the growing emphasis on sustainability, many developers building digital platforms still prioritize efficiency and user experience, often neglecting green computing principles. This oversight in reducing the digital footprint of these platforms represents a missed opportunity to mitigate the environmental impact of digital technology itself.

B. Problem Statement

The GreenNudge project seeks to develop a digital platform that connects Small and Medium-sized Enterprises (SMEs) to facilitate a transition towards greener supply chains. The platform's primary objective is to support SMEs in becoming sustainable suppliers by providing them with the necessary digital tools and resources. Given the complexities involved in system design and the increasing use of AI, critical questions arise: What are the most effective strategies to minimize the platform's carbon footprint while supporting sustainable transitions? What choices and trade-offs must developers consider to balance efficiency, user needs, and environmental sustainability? Developers often prioritize user experience and functionality, neglecting green computing principles, which raises the need for a holistic approach that integrates sustainability into digital platform design.

C. Importance of the Problem

This issue is crucial from both environmental and economic perspectives. Without adopting sustainable digital practices, the textile industry risks further contributing to climate change. The GreenNudge platform offers an opportunity to set a precedent in the technology sector, demonstrating how green computing principles can be integrated into digital infrastructure. By addressing this problem, the project can provide valuable insights not only to the textile industry but also to other developers, encouraging the adoption of sustainable practices that maintain low carbon footprints while enabling efficient and impactful business operations.

II. Proposed Solution

While there is extensive research on building sustainable websites or optimizing supply chains within specific industries, we found limited existing work to guide us in achieving the ambitious goals of our client, GreenNudge. Therefore, our research will focus on filling this gap by identifying effective strategies to build a sustainable web platform that empowers and connects small to medium-sized enterprises in the textile industry, considering the business constraints faced by GreenNudge, the startup. Our solution will concentrate on three areas: a) third-party services we need to utilize, b) principles of website design and implementation, and c) the algorithms chosen for implementing the features.

A. Literature Review

To develop the GreenNudge web platform connecting SMEs in the textile industry, we reviewed literature on green software engineering and supply chain management. We also examined documentation and performance reports of web hosting services and evaluated middleware relevant to our development process. Additionally, we maintained close communication with our primary client, GreenNudge, and our end users, the SMEs. Our approach was inspired by the Web Sustainability

Guidelines from the Sustainable Web Design Community Group[1].

The Web Sustainability Guidelines defined six principles of web sustainability: clean, efficient, open, honest, regenerative, and resilient. The guideline also specifies 4 main areas that assist web designers to develop more efficient websites: 1) user experience design, 2) web development, 3) hosting, infrastructure and systems, 4) business strategy and product management.

To address the first area, user experience, we must understand the website's end users. Our product should align with their business needs, use appropriate terminologies, and incorporate recognized design patterns to ensure ease of use and convenience. Additionally, we researched supply chain management in the textile industry and studied lean management theory in business administration. Several studies highlight how sustainable supply chain management can reduce the environmental footprint of industries, including textiles. For example, in 2007, Koprulu and Albayrakoglu proposed a supplier selection model that helps companies integrate environmental criteria sustainability into their supply chains when choosing suppliers, which directly influences GreenNudge's supplier matching feature [2]. In addition, Jodlbauer et al. emphasized the current challenges and recommendations for action in supply chain management, particularly in the context of lean management and the reduction of waste and energy consumption [3]. Their findings supports GreenNudge's scenario modeling tool, which helps SMEs simulate various supply chain options to optimize sustainability.

To address the second area and the third area, web development, hosting, infrastructure and systems, energy efficiency in software implementation and cloud infrastructure is crucial. While software itself does not emit greenhouse gasses, it operates on hardware in data centers and drives data through transmission networks, each contributing around 1 percent to global energy-related greenhouse gas emissions. According to Ecograder, a website scoring tool, factors such as unused code, improperly sized images, and third-party scripts can increase a website's carbon footprint. At the software level, we decided to follow existing best practices of green computing described in this excellent lean software article[4]. Moreover, we will use the mentioned tools to measure the greenness of our software development process.

In 2024, Amazon announced its achievement of matching 100% of its energy use with renewable sources across all operations, including AWS data centers, seven years ahead of schedule. This milestone reflects Amazon's strong commitment to sustainability, with more than 500 solar and wind projects worldwide, which significantly contributes to reducing the carbon footprint of its digital infrastructure [5]. AWS's eco-friendly cloud services, such as S3 for static content, CloudFront for content delivery and Lambda for serverless computing, could ensure scalable, energy-efficient operations [6].

To address the fourth area, business strategy and product management, we focused on developing efficient algorithms that serves GreenNudge's core competency: matching suppliers and buyers. We recognize that using complex models and training on large datasets can significantly contribute to carbon dioxide emissions [4]. Additionally, due to GreenNudge's start-up nature, we currently lack sufficient real-life business data to train complex models.To mitigate this, GreenNudge opts for simpler, more efficient algorithms that maximize the utility functions of the buyers using arbitrary weights [7, 8], based on our understanding of the business domain and our expertise in algorithms. Thereby we reduce the energy demand typically associated with heavy data processing. This approach aligns with the findings by Chadli et al. and García-Martín et al., which emphasize the environmental costs of engineering complex machine learning systems and the need for greener alternatives [9, 10].

To address the fourth area, business strategy and product management, we focused on developing efficient algorithms that serves GreenNudge's core competency: matching suppliers and buyers. We first drew inspiration from recommendation models in machine learning, which facilitate match-making between two parties. Examples of these models include collaborative filtering, content-based filtering, and clustering [11]. We then recognize that using complex models and training on large datasets can significantly contribute to carbon dioxide emissions [4]. Additionally, due to GreenNudge's start-up nature, we currently lack sufficient real-life business data to train complex models.To address this issue, we explored additional matching algorithms, including the Hungarian algorithm and maximum weighted matching algorithms [12] [13]. We chose to use simpler, more efficient algorithms that maximize the utility functions of the buyers by applying arbitrary weights [7] [8], based on our understanding of the business domain and our expertise in algorithms. Thereby we reduce the energy demand typically associated with heavy data processing. This approach aligns with the findings by Chadli et al. and García-Martín et al., which emphasize the environmental costs of engineering complex machine learning systems and the need for greener alternatives [9] [10].

B. Identifying Gaps and Research Questions

Despite the existing research on sustainable supply chains and energy-efficient computing, significant gaps of research remain, especially for the specific choices to make in order to create a centralized platform for connecting SMEs in the textile industry. The key research questions we aim to answer are:

a) **What third-party services can we utilize to streamline the development and deployment process?**
To streamline the development, deployment, and scaling of the GreenNudge platform, leveraging third-party services is essential. These services can significantly reduce the complexity and time required to build, maintain, and scale the platform. For instance, cloud computing services offer scalable infrastructure that can handle varying loads efficiently, ensuring that the platform remains responsive

and reliable as user demand grows. Additionally, utilizing content delivery networks (CDNs) can enhance the platform's performance by reducing latency and improving load times for users globally, which is crucial for maintaining a seamless user experience at scale.

Middleware solutions can further streamline development by handling common tasks like routing, session management, and data validation, enabling developers to focus on core functionalities. By integrating these third-party services and middleware solutions, we can focus on our core mission of designing and implementing the software.

b) **What principles and best practices should we follow when designing and implementing the website?**

When designing and implementing the GreenNudge platform, it is crucial to follow principles and best practices that ensure the platform is efficient, user-friendly, and sustainable. This involves prioritizing user-centered design to meet the needs of brands and SMEs, integrating green computing principles to minimize environmental impact, and ensuring scalability to handle growth. Additionally, we aim for a design that facilitates easier updates and maintenance, and continuous improvement practices ensure the platform evolves based on user feedback and technological advancements.

c) **How should we design the algorithms that support the main features of our website?**

Designing the algorithms that support the main features of the GreenNudge platform involves addressing several critical considerations to ensure they are effective, efficient, and aligned with the platform's goals. This includes determining the most suitable algorithms for tasks such as matching brands with eco-friendly SMEs, optimizing supply chain logistics, and analyzing user data for insights. The question also encompasses evaluating the trade-offs between algorithm complexity and performance, ensuring scalability to handle increasing data volumes, and integrating sustainability principles to minimize computational resources and energy consumption. Overall, this question seeks to explore the comprehensive approach needed to develop robust algorithms that enhance the platform's functionality while supporting its mission of promoting sustainable practices in the textile industry.

C. How GreenNudge Answers These Research Questions

1) Deploy the website to AWS and thoroughly investigate and evaluate all middleware and APIs before use: The GreenNudge website leverages the reliability and scalability of AWS cloud services. By utilizing the Platform as a Service (PaaS) model, we can focus on software design while delegating infrastructure management to AWS experts. This approach allows us to scale our services dynamically according to demand, ensuring that the platform remains responsive and efficient as user numbers grow. Additionally, AWS provides a robust suite of tools and services that facilitate the seamless integration of advanced features based on user needs. Thoroughly investigating and evaluating all middleware and APIs before use is crucial to ensure compatibility, security, and performance. This due diligence helps us select the most suitable technologies that align with our sustainability goals and operational requirements.

2) Apply Lean Code principles and use recognized design patterns: To enhance the GreenNudge platform, we apply Lean Code principles and use recognized design patterns, both of which are essential for creating efficient, maintainable, and user-friendly software. Lean Code principles emphasize reducing unnecessary elements that can bloat the platform and degrade performance. This includes minimizing the use of excessive JavaScript, high-resolution images, and videos that can slow down load times and increase energy consumption. By focusing on simplicity and efficiency, we ensure that the platform remains fast and responsive. Additionally, reusing code through modular design and frequent refactoring helps to eliminate redundancies and improve maintainability. Common design patterns such as the Singleton, Factory, and Observer patterns facilitate this process by providing standardized solutions to recurring problems, making the codebase more robust and easier to manage.

Using recognized design patterns also plays a crucial role in enhancing the user experience. Patterns like Model-View-Controller (MVC), etc. help structure the application in a way that is intuitive for users, making it easier for them to navigate and interact with the platform. By adhering to these patterns, we create a consistent and predictable user interface, which reduces the learning curve and improves overall usability. This approach not only supports the platform's functionality but also aligns with our sustainability goals by ensuring that the digital infrastructure is both efficient and user-friendly.

3) Use simple algorithms to match SMEs and collect data in preparation for training AI models: To effectively match SMEs with brands and retailers on the GreenNudge platform, we will start by combining business domain knowledge with computer science principles to develop simple, yet effective, algorithms. Initially, we will implement a straightforward maximization algorithm that takes into account various factors such as innovation/sustainability, trust, cost and quality. This approach allows us to create a functional matching system that can operate efficiently with limited data. As the platform grows and more data is collected, we can refine and enhance these algorithms.

Once we have accumulated sufficient real-world data, we will use it to train more sophisticated AI models. This iterative process ensures that our matching system becomes increasingly accurate and effective over time. By starting with a simple algorithm, we can quickly deploy a working solution and gather valuable insights, which will inform the development of advanced AI-driven matching algorithms. This strategy not only accelerates the initial deployment but also lays a strong foundation for continuous improvement, ultimately leading to a highly efficient and intelligent matching system that supports the platform's sustainability goals.

D. AWS System Architecture

The system architecture of GreenNudge prioritizes energy efficiency, scalability, and sustainability by leveraging AWS services tailored to minimize environmental impact. The architecture is broken into two layers: static content hosting and dynamic content processing.

Frontend:

- **Amazon CloudFront** is used for content acceleration, ensuring low-latency delivery of static content globally. CloudFront caches content closer to users, reducing latency and improving performance, while optimizing energy consumption.
- **Amazon S3** serves as the storage solution for static assets such as HTML, CSS, and JavaScript files, optimizing the energy usage for delivering web content.

Backend:

- **AWS API Gateway** handles API calls efficiently, providing a scalable solution for routing requests between the frontend and backend services.
- **AWS Lambda**, enables serverless computing, handling backend logic in a manner that only consumes resources when required, thus aligning with sustainability goals. This helps minimize idle server time and resource waste.

Database:

- **Amazon DynamoDB**, a NoSQL database service, is utilized for data storage. It ensures scalability and high availability for handling user and supplier data while keeping energy consumption low through AWS's eco-friendly infrastructure.

Monitoring:

- **AWS CloudWatch** is employed to monitor system performance and track any bottlenecks in real time, ensuring optimal usage of resources and identifying areas for further energy savings.
- **AWS Carbon Footprint Tool** is used to monitor the carbon emissions of the platform during the development and operational stages. This allows GreenNudge to continuously measure its environmental impact and implement measures to reduce its carbon footprint further. This service will be activated once we have enough data on the system.

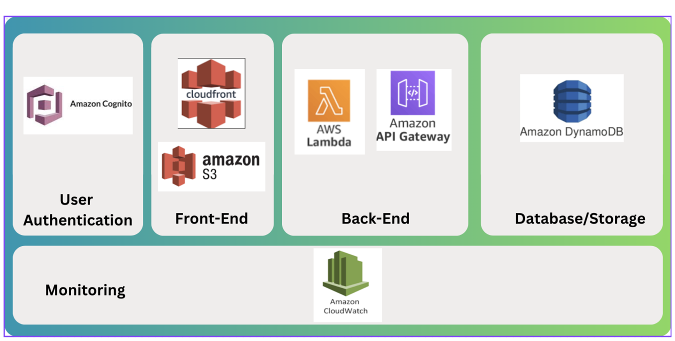

Fig. 1. GreenNudge System Architecture

These architectures ensure that GreenNudge's platform meets the highest sustainability standards, not only by facilitating sustainable supply chains for SMEs but also by operating in an energy-efficient manner.

III. Evaluation Methodology

The evaluation process for the GreenNudge platform will follow a comprehensive approach combining simulations, experiments, and user studies to assess the effectiveness of our proposed solution across three core areas: hosting and middleware services, principles of design and implementation, and algorithm development. The following subsections outline our strategy for evaluating each of these areas, detailing the tools, methodologies, and metrics employed to minimize the platform's carbon footprint while ensuring scalability and energy efficiency.

A. Literature Review Supporting Evaluation

To support the first evaluation method, research on sustainable cloud computing and energy-efficient techniques provides a solid foundation for evaluating the GreenNudge platform's sustainability efforts. Katal, Dahiya and Choudhury conducted a systematic survey on sustainable cloud computing techniques, highlighting how scalable, on-demand infrastructure, such as AWS, significantly reduces energy consumption by optimizing resource allocation and minimizing idle resource usage [14]. Studies on energy-efficient data centers further show that cloud infrastructure can lower power consumption by up to 70% compared to traditional systems [15]. These align with our platform's use of AWS CloudWatch for tracking resource consumption and the AWS Carbon Footprint Tool to quantify emissions, ensuring ongoing optimization of our platform's energy usage.

Similarly, Xing, Qian, and Zaman provide valuable insights into how cloud-based platforms can support real-time data collection for assessing environmental impacts [16]. This informs our approach to integrating third-party middleware and APIs, for instance, GreenFrame, which enables real-time monitoring, dynamic analytics, and the provision of key sustainability data, such as carbon emission factor. By adopting a similar model that facilitates real-time life-cycle data collection and collaboration, GreenNudge could empower SMEs and green suppliers to assess environmental impacts more accurately across supply chain stages. Integrating life-cycle assessment (LCA) tools and shared databases for carbon footprint analysis may provide opportunities for GreenNudge to offer dynamic, data-driven recommendations aimed at reducing emissions and optimizing resource use. This approach enhances transparency and enables informed decision-making, driving meaningful environmental improvements across the supply chain.

To effectively measure the carbon footprint of the GreenNudge platform, we also referred to the methodology proposed by Lottick et al., which emphasizes programmatic energy usage measurement and CO_2 emissions calculation tailored to the energy grid mix powering the computation [17]. Their approach uses energy measurement tools, such as the RAPL interface, to monitor energy consumption at the CPU level. The energy data, expressed in kilowatt-hours (kWh), is converted into CO_2 emissions using emission factors reflecting the local energy mix. GreenNudge adopts a similar philosophy by leveraging AWS tools for real-time energy monitoring and reporting. Additionally, during the development stages, we can convert kWh energy data into CO_2 emissions to assess and optimize the environmental impact of new features early in the design process. This iterative approach ensures our platform remains energy-efficient and aligned with our sustainability goals.

In evaluating the second approach, Şanlıalp et al. primarily explores how refactoring techniques can enhance energy efficiency in software, particularly for portable devices. By reducing redundancy and improving maintainability in object-oriented code, the study demonstrates how refactoring directly lowers energy consumption [18]. This study supports our strategy of optimizing system efficiency through continuous code refactoring, reducing unnecessary computational load, and ensuring long-term energy savings across the platform. Additionally, the study's focus on energy-efficient coding practices aligns with our platform's broader goal of integrating sustainable design patterns throughout both its back-end and front-end components. This allows us to streamline resource use while maintaining platform performance. In line with this, we have already begun refactoring key parts of our codebase to reduce redundant processes and improve overall maintainability, ensuring our platform operates more efficiently.

In addition, Kiourtis et al.'s research emphasizes that UI/UX designers play a crucial role in reducing the environmental impact of digital platforms by adopting green design principles. The study presents best practices for reducing CO_2 emissions in web applications through efficient UI/UX design, showing that such practices can significantly reduce energy consumption and carbon footprints [19]. This aligns directly with our goals for energy-efficient UI/UX in the GreenNudge platform. Accordingly, we have implemented these principles by optimizing the platform's design to minimize unnecessary interactions and reduce server requests.

In supporting the last evaluation method, Caballar stresses the importance of minimizing computational overhead through efficient algorithms [4]. In Caballar's work, it is argued that inefficient algorithms not only increase energy consumption but also place undue strain on cloud infrastructure, leading to unnecessary resource use. By designing algorithms that streamline data processing and optimize calculations, platforms can significantly reduce their carbon footprint. This principle is embedded in our platform's design and is at the core of how the platform operates. The platform is designed to minimize computational overhead by leveraging precomputed data and scenario modeling, which reduces the need for constant real-time recalculations. This ensures that our supplier matching algorithms are not only fast and accurate but also energy-efficient.

Finally, to support the evaluation of energy-efficient AI operations, Tien et al. highlight the importance of selecting suitable machine learning and deep learning models for optimizing energy usage in the built environment [20]. Inefficient models can lead to increased resource consumption, making it crucial to balance accuracy with computational efficiency. This aligns with the platform's approach, where we adopt Green AI practices, optimizing the training process and refining algorithms to reduce unnecessary computational load. By focusing on lightweight models initially and scaling with data, we ensure energy efficiency while maintaining performance.

B. Evaluation of Hosting and Middleware Services

The GreenNudge platform leverages AWS services to monitor energy consumption and its carbon footprint. The primary focus of our evaluation is on resource utilization and sustainability, assessing how effectively key AWS services are managed. Additionally, we assess the third-party middleware to understand its impact on improving platform performance and sustainability outcomes.

1) Tracking and Measuring Carbon Impact with AWS Services: To track and optimize resource consumption, we will leverage AWS CloudWatch to continuously monitor key services such as Lambda, S3, and DynamoDB. This monitoring allows us to identify patterns in energy usage and make data-driven adjustments to the resource allocation. In addition, we will employ both Cloud Carbon Footprint and the AWS Carbon Footprint Tool to quantify the emissions over time. These tools help us establish baseline carbon metrics and track progress towards long-term reduction goals, ensuring that the platform's infrastructure aligns with sustainability objectives by optimizing resource usage and minimizing energy consumption.

2) API Validation: We aim to measure the effectiveness of third-party middleware using tools such as GreenFrame and PowerAPI, which will track energy consumption at the middleware level. These APIs enhance sustainability by providing key data, such as carbon emission factors and renewable energy metrics, enabling real-time monitoring of energy consumption and emissions. This fosters collaboration and ensures data-driven recommendations for reducing emissions and optimizing resource use.

Additionally, manual calculations of carbon emissions based on CPU usage, inspired by Lottick et al., will provide further accuracy in tracking the platform's energy efficiency. Leveraging third-party APIs also reduces the need for developing in-house sustainability tools, leading to greater cost efficiency and scalability. This allows GreenNudge to focus on its core mission while encouraging broader adoption of sustainable practices among users. By comparing environmental performance metrics, we can validate the contributions of these APIs to our overall sustainability goals.

C. Evaluation of Design and Implementation Principles

The platform's design is informed by lean coding principles and the application of recognized design patterns to achieve a balance between user experience and energy efficiency.

1) Lean Code Assessment: The principle of "lean coding" emphasizes reducing redundancy, reusing code components, and refactoring to optimize system performance and minimize resource consumption. In this evaluation, we will focus on how the application of lean coding practices impacts the platform's overall energy efficiency and performance. The size of the codebase will be tracked over time to identify improvements made through refactoring and code optimization efforts. By using AWS CloudWatch, we can closely monitor key performance metrics such as execution time and memory usage, providing a detailed understanding of how efficiently the platform operates under different conditions. For instance, longer execution times or higher memory usage can indicate inefficient code that could lead to greater energy consumption. Through these metrics, we can gauge how effectively the lean coding practices have improved performance. Along with GreenFrame, which evaluates system efficiency during code execution, CodeCarbon will help measure the energy usage of the code during refactoring. This combined data will provide insights into resource-intensive areas and actionable feedback for further optimization, reducing the platform's carbon footprint.

2) Interaction Analysis: User interaction is a critical factor influencing both user experience and the platform's energy consumption. To evaluate the efficiency of user interactions, we will analyze the number of steps required to complete key tasks on the platform. This analysis involves examining how the design of the user interface (UI) influences user actions, such as the number of clicks, navigation paths, and form submissions. The goal is to minimize unnecessary actions, which in turn reduces the load on the system and the amount of energy consumed. Metrics such as time-on-task, which measures how long it takes for users to complete specific tasks, will be collected and analyzed. We will also track energy consumption data using AWS CloudWatch and GreenFrame during these user interactions to assess how recognized design patterns impact resource usage. Additionally, Ecograder will assess the overall energy efficiency of web-based interactions, helping to further minimize server requests and reduce energy usage in UI/UX design. By combining these tools, we can ensure an intuitive, energy-efficient platform that aligns with sustainable design goals.

D. Evaluation of Algorithms Design

To support SME matching and AI model development, the algorithm design focuses on achieving efficiency while integrating business domain knowledge with computer science principles.

1) Performance and Complexity Benchmarking: The effectiveness of algorithms in the GreenNudge platform is crucial to both its operational efficiency and sustainability goals. The platform's algorithms—responsible for tasks such as matching SMEs with brands—must be efficient in terms of both computational resources and energy consumption. To evaluate the performance of these algorithms, we will employ a variety of profiling tools that measure execution time, memory usage, and resource consumption during their execution. The benchmarking process will focus on two aspects: evaluating both the performance and complexity of the algorithms. This involves tracking how efficiently the algorithms execute as the data volumes increase. Algorithms that perform well with small datasets may exhibit bottlenecks when handling large volumes of data. Using profiling tools, we will measure the time it takes for the algorithm to complete its tasks under various conditions, such as different data sizes and concurrent users. Additionally, we will assess the algorithms' computational complexity using Big-O notation to understand their performance in worst-case, average-case, and best-case scenarios. By evaluating this complexity, we can refine the algorithms to optimize resource consumption and ensure they remain efficient as the platform scales.

2) Data Collection & Green AI Operations: In the early stages of the platform, the algorithms will rely on a simple maximization algorithm to collect and process initial data. This straightforward approach allows the platform to function effectively with limited data while providing the necessary insights to improve the system over time. The maximization algorithm will assign scores to SMEs based on factors such as innovation, trust, cost and quality. This allows for quick and efficient matching of SMEs to brands.

As we collect more data from user interactions and platform activities, we will gradually shift to using more advanced AI models that leverage machine learning to offer better and more personalized recommendations. However, training these models can require a lot of computational power and energy. To address this, we will adopt Green AI practices, which focus on making AI processes more energy-efficient. This means optimizing the training process so that models are only updated when necessary, specifically when there is enough new data or when significant accuracy improvements can be made. Additionally, we will prioritize the use of energy-efficient algorithms, starting with simpler models such as decision trees or linear regression, and gradually introducing more complex models, such as neural networks, as our data grows. This approach allows us to balance accuracy with energy consumption.

E. Data Sources

In the early stages, synthetic AI-generated data will be utilized to support simulations and model supply chain scenarios when real-world data is unavailable. As SMEs interact with the platform, their real-world supply chain data, including material choices and sustainability goals, will be integrated to personalize recommendations and enhance scenario modeling. In addition, GreenNudge will source from external data sources, such as established sustainability benchmarks like the Higg Index, to ensure proper environmental impact assessments of suppliers and materials.

F. Timeline for Evaluation Activities

The evaluation will begin in the first month with simulations and synthetic data collection on sustainability scenarios, focusing on system performance and preliminary energy impact analysis. Detailed user testing will be conducted to assess system usability and functionality, complemented by stakeholder interviews to gather qualitative insights. Feedback from these activities, along with simulation results, will guide UX/UI adjustments. By the second month, real-world client data will be integrated to establish a reliable baseline for carbon monitoring. Tools such as AWS CloudWatch, PowerAPI (middleware), CO2.js (code-level), and GreenFrame (web design) will be utilized for continuous energy consumption monitoring. Clear milestones for data collection, usability testing, and energy impact assessment will ensure a structured and thorough evaluation process.

IV. Representation of Results

The evaluation of the GreenNudge platform focuses on measuring, tracking, and optimizing energy consumption and carbon emissions across multiple layers—cloud infrastructure, code execution, middleware, and web design. This multi-layer approach ensures a comprehensive understanding of the platform's sustainability performance.

A. Multi-Layer Sustainability Monitoring

GreenNudge uses a multi-layered approach to monitor its sustainability, focusing on cloud infrastructure, code-level operations, middleware, and web design.

1) Cloud Infrastructure Monitoring: At the cloud infrastructure level, tools like *Cloud Carbon Footprint*, *AWS Carbon Footprint* (once feasible), and *GreenFrame* provide crucial insights into how the platform's cloud services contribute to its carbon footprint. *AWS Carbon Footprint* monitors energy consumption and emissions from AWS services, such as Amazon S3 for static content and EC2 for scalable computing tasks. This helps identify inefficiencies and optimize resource usage by reducing idle capacity and dynamically scaling services.

GreenFrame plays a vital role in simulating various usage scenarios and traffic patterns, providing detailed insights into the platform's scalability and efficiency under different conditions. It evaluates how increased traffic impacts resource usage and carbon emissions, helping us implement solutions like caching and load balancing. By understanding these patterns, GreenNudge ensures that its platform remains energy-efficient while dynamically scaling to meet user demand.

2) Code-Level Monitoring: At the code level, *CO2.js* serves as a powerful tool for calculating the carbon emissions associated with data transfer during platform operations. By analyzing emissions per gigabyte (GB) of data transfer, *CO2.js* provides a clear picture of the environmental cost of digital operations. Leveraging this tool, we have implemented green hosting solutions, optimized data transfer processes, and made further refinements in computational algorithms that can help lower energy consumption even more, ensuring the platform operates efficiently while minimizing its environmental impact. For context, transferring 1 GB of data is currently equivalent to the carbon emissions of driving a fuel-efficient car for about one mile, highlighting the importance of continuous improvements at the code level.

3) Middleware and API Monitoring: For middleware and API usage, *PowerAPI* was employed to estimate CPU energy consumption in real-time for API-related tasks, offering valuable insights into the energy costs of middleware operations. Our analysis highlights opportunities for architectural refactoring to reduce CPU load and improve energy efficiency. Additionally, manual CPU-based carbon emission calculations inspired by Lottick et al. provide complementary insights, estimating emissions based on CPU activity and energy grid mix. These combined approaches ensure API integrations align with the platform's sustainability goals while optimizing energy usage.

4) Web Design Monitoring: At the web design level, we also leveraged *GreenFrame* to evaluate energy consumption and carbon dioxide emissions based on the user journey. With steps such as minifying CSS and JavaScript, compressing visuals, and lazy-loading, we have significantly reduced energy consumption for user-facing operations. These findings highlight that every layer of our systems is connected, and small changes in one area can make a big difference overall.

Together, these tools provide a robust framework for measuring and improving the platform's energy efficiency across all operational layers.

B. Results from Evaluation Tools

The results from these tools offer valuable insights into the platform's carbon footprint, helping us continuously optimize GreenNudge for greater sustainability.

Cloud-Level Results: AWS Carbon Footprint and GreenFrame provide detailed reports on the energy usage and emissions from the platform's cloud services. such as Amazon S3 for storage and EC2 for scalable computing tasks. GreenFrame simulations have shown the impact of different traffic patterns on energy usage, prompting optimizations such as caching and load balancing to minimize energy consumption during high-demand periods. These results allow the team to strategically manage cloud resources, ensuring low carbon emissions even as user demand grows.

Code-Level Results: CO2.js was utilized to calculate emissions per gigabyte of data transfer, revealing the environmental cost of digital operations. Using this data, the team has implemented green hosting and optimized data transfers. Additionally, refining computational algorithms—particularly in supplier matching—has further reduced CO_2 emissions, equivalent to a car driving one mile for every gigabyte transferred. These optimizations align with GreenNudge's commitment to reducing the platform's digital carbon footprint.

Middleware and API-Level Results: PowerAPI and manual calculations based on CPU usage have helped identify energy-intensive middleware services. By monitoring the energy consumption of third-party APIs, the team can make informed decisions about whether to optimize or replace these services. This has led to the selection of more energy-efficient middleware options, reducing the platform's overall energy consumption. Manual carbon emission calculations further ensure that CPU usage is continuously tracked, with real-time data driving energy-saving decisions.

Web-Level Results: GreenFrame has provided valuable insights into energy consumption associated with user-facing interactions, highlighting areas like large visuals and inefficient scripts as key contributors to web-related energy usage. Through compressing images, reducing script sizes, and implementing lazy-loading techniques, the platform has achieved enhanced front-end energy efficiency. These improvements have not only reduced energy consumption but also enhanced the user experience by minimizing load times and server requests.

Overall, these results confirm that the GreenNudge platform is successfully balancing energy efficiency and scalability, ensuring that it operates with a minimal carbon footprint while meeting the needs of SMEs in the textile industry. Continuous monitoring through these tools allows the team to make data-driven decisions that align with the platform's long-term sustainability objectives.

C. Results from Evaluation Tools

1) Cloud Infrastructure and Web Design: To evaluate energy consumption and CO_2 emissions during typical user interactions, GreenFrame and AWS Carbon Footprint were utilized. The GreenFrame simulation was based on a common user journey, which involved navigating to the homepage, scrolling through content, logging in or signing up, and uploading emissions data such as utility bills.

This scenario resulted in a total energy consumption of 91 mWh, with 40 mg CO_2 equivalents emitted. The breakdown of energy usage revealed that 72% (65 mWh) was consumed by network activity, 25% (23 mWh) by screen activity, and 4% (3.2 mWh) by CPU operations. Notably, memory and disk usage were negligible, indicating efficient resource utilization for these components. The energy consumption and carbon emissions for this journey are roughly equivalent to charging a smartphone once, making it a relatively low-impact process. However, given that network activity accounted for the majority of the energy use, optimization strategies such as implementing content delivery networks (CDNs), caching mechanisms, and reducing data transfer sizes remain critical to further reducing emissions, as shown in Figure 2.

2) Middleware: PowerAPI was used to monitor the energy consumption of middleware and API-related tasks in real-time. Middleware operations, integral to GreenNudge's supplier-buyer matching engine, demonstrated an energy consumption of 350 mWh/min during typical API workloads. This energy consumption level is equivalent to keeping an LED bulb powered for 6 hours, highlighting the computational intensity of API interactions. These findings call attention to the need for optimizing middleware architecture to reduce the CPU load and energy expenditure. Potential improvements include streamlining API logic, adopting lightweight protocols, and leveraging more efficient third-party services to minimize the platform's overall energy footprint.

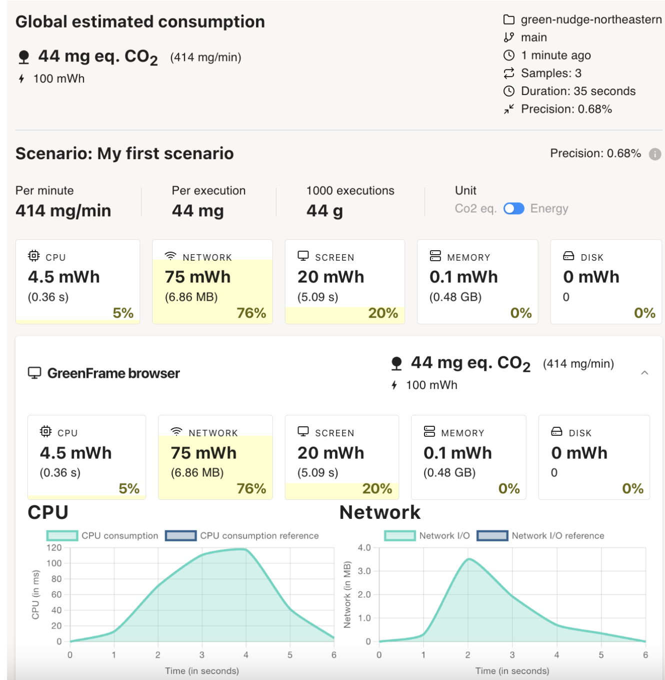

Fig. 2. GreenNudge System Architecture

3) Code-Level: CO2.js provided a detailed analysis of the platform's code-level efficiency by estimating the carbon emissions of data transfers. The findings indicate that transferring 1 GB of data generates 337.677 grams of CO_2 emissions, which is approximately equivalent to the emissions of driving a small, fuel-efficient car for 1 mile. This highlights the significant environmental impact of digital data operations, especially for platforms managing large datasets. To address this, GreenNudge has implemented green hosting using AWS, optimized its data pipelines, and reduced unnecessary data transfers. However, further efforts to refine computational algorithms and adopt energy-efficient coding practices will help to significantly lower emissions and improve sustainability.

Methodology & Results

Evaluation Level	Tools Used	Environmental Impact Per Use Case	Results	Optimization Recommendations
Cloud Infrastructure	GreenFrame AWS Carbon Footprint	User Journey: CO2 = 1 phone charge	40 mg eq. CO2; 91 mWh energy; Network: 65 mWh (72%), Screen: 23 mWh (25%), CPU: 3.2mWh (4%)	• Continue monitoring and dynamically adjust resources for efficiency.
Middleware	PowerAPI	API Tasks: Energy = 6 hours of an LED bulb	Total Energy Consumption: 350 mWh/min for API middleware tasks	• Focus on deep architectural improvements for energy efficiency.
Code-Level	CO2.js	Data Transfer: 1GB = CO2 of driving 1 mile	337.677 grams CO2 per GB	• Refine algorithms to reduce computational overhead.
Web Design	GreenFrame	User Journey: CO2 = 1 phone charge	40 mg eq. CO2; 91 mWh energy; Network: 65 mWh (72%), Screen: 23 mWh (25%), CPU: 3.2mWh (4%)	• Minify CSS/JS and adjust font-display settings for better performance.

Fig. 3. Methodology and Results

4) Key Observations and Opportunities: The results across all levels of evaluation—cloud infrastructure, middleware, code-level, and web design—revealed valuable insights into the platform's energy consumption patterns and carbon footprint:

- **Cloud and Web Design Efficiency**: While GreenFrame simulations demonstrated relatively low emissions for user journeys, network activity remains the dominant energy consumer. Strategies such as optimizing resource sizes, implementing lazy-loading techniques, and compressing media assets can further reduce the carbon footprint.
- **Middleware Optimization**: Middleware and API workloads are energy-intensive, with significant energy consumption attributed to CPU usage. Addressing this through architectural refactoring and the use of efficient protocols is a priority.
- **Code-Level Refinement**: The environmental cost of data transfer, as shown by CO2.js, emphasizes the importance of minimizing data sizes and optimizing computational processes.

5) Broader Implications: The use of tools such as GreenFrame, PowerAPI, and CO2.js not only provides a granular understanding of the platform's energy consumption but also offers actionable pathways for improvement. By combining detailed monitoring with targeted optimizations, GreenNudge is aligning its operational practices with its sustainability objectives. These evaluations and subsequent refinements ensure that the platform remains environmentally friendly while delivering high performance to users.

V. Future Work

Future developments for the GreenNudge platform will center on enhancing sustainability and operational efficiency through several strategic initiatives. First, caching techniques will be implemented to optimize performance by minimizing server requests, thereby reducing energy consumption and improving the overall user experience. Second, the integration of AI-based energy monitoring systems will enable real-time tracking of energy usage and carbon emissions. Leveraging machine learning, these systems will analyze patterns, predict consumption, and produce accurate carbon footprint reports, providing a basis for ongoing optimization efforts. Finally, to ensure sustainable scaling, additional AWS services such as Graviton processors, designed for energy efficiency, and EC2 Auto Scaling will be explored. These services will dynamically manage resources according to demand, minimizing idle energy consumption. Collectively, these advancements aim to reinforce GreenNudge's commitment to environmentally responsible growth and platform efficiency.

A. Enabling Caching for Optimized Performance

Another effective strategy for enhancing the sustainability of the GreenNudge platform is implementing caching techniques. By enabling caching, the platform can store frequently accessed content locally on users' devices after their initial visit. This allows subsequent visits to retrieve data from the local cache instead of repeatedly sending requests to the server. Not only does this approach significantly reduce the energy consumption associated with data transmission over the network, but it also enhances website performance, providing faster loading times and a better user experience. Implementing caching aligns with sustainable web design practices, as it minimizes server load and optimizes the overall energy efficiency of the platform, contributing to GreenNudge's commitment to reducing its environmental impact.

B. Integrating AI-Based Energy Monitoring Systems

Currently, GreenNudge leverages various tools to assess the platform's environmental impact. While these tools provide valuable insights, future work could focus on integrating AI-based energy monitoring systems to continuously track the platform's energy consumption and carbon emissions in real time. By utilizing machine learning algorithms, these systems can analyze energy usage patterns and predict emissions using both real-time data and historical trends. This AI integration would automate energy efficiency assessments and generate dynamic, accurate carbon footprint reports, enabling continuous evaluation and optimization. Such an approach would not only refine the precision of our evaluations but also help identify optimization opportunities, allowing for proactive adjustments in platform design and deployment. Furthermore, AI-driven monitoring could be seamlessly integrated with existing cloud and middleware solutions, offering a comprehensive and scalable framework for sustainability assessment.

C. Explore Additional AWS Services for Scaling

As we scale the GreenNudge platform, we can explore additional AWS services that further align with green computing principles. For instance, AWS Graviton processors are specifically designed for energy efficiency and offer lower carbon emissions compared to traditional instances, making them a better option for reducing the environmental footprint of our operations. In addition, leveraging EC2 Auto Scaling can dynamically adjust resource allocation based on real-time demand, ensuring that we are only using the necessary computational power when needed. This helps avoid idle energy consumption and optimizes resource efficiency. By integrating these services, we can ensure that the platform scales sustainably, maintaining a strong focus on minimizing energy use while supporting operational growth.

VI. Conclusion

The GreenNudge platform represents a strategic effort to integrate sustainability principles into digital infrastructure for the textile industry, particularly targeting SMEs striving to transition to greener supply chains. By leveraging advanced technologies, such as AWS cloud services and AI-based energy monitoring, the platform aims to minimize its carbon footprint while providing valuable support to businesses committed to environmental sustainability. Through implementing caching techniques, integrating AI for real-time energy monitoring, and exploring scalable and energy-efficient cloud services like AWS Graviton processors, GreenNudge seeks to balance efficiency, performance, and sustainability.

The ongoing and future development of the platform will focus on refining these systems and incorporating user feedback to continuously optimize energy consumption and operational efficiency. This holistic approach not only aligns with global decarbonization goals, such as the 2050 Paris Agreement, but also sets a precedent for sustainable digital platform design. GreenNudge's commitment to reducing its environmental impact while supporting SME growth and decarbonization efforts highlights the potential for technology to drive meaningful change across industries. The insights and methodologies derived from this project can serve as a guide for other developers and sectors seeking to integrate green computing into their digital solutions.

References

[1] World Wide Web Consortium (W3C). *Sustainable Web Design Community Group*. https://w3c.github.io/sustyweb. Accessed: 2024-10-14.

[2] Asli Koprulu and M Murat Albayrakoglu. "Supply chain management in the textile industry: a supplier selection model with the analytical hierarchy process". In: *Proceeding of the international symposium on the analytic hierarchy process*. 2007, pp. 3–6.

[3] Herbert Jodlbauer et al. "Supply chain management: a structured narrative review of current challenges and recommendations for action". In: *Logistics* 7.4 (2023), p. 70.

[4] Rina Diane Caballar. "We Need to Decarbonize Software: The Way We Write Software has Unappreciated Environmental Impacts". In: *IEEE Spectrum* 61.4 (2024), pp. 26–31.

[5] About Amazon Team. *Amazon meets 100% renewable energy goal seven years early*. https://www.aboutamazon.eu/news/sustainability/amazon-meets-100-renewable-energy-goal-seven-years-early. Accessed: 2024-10-11. 2024.

[6] Andrea Bianco, Reza Mashayekhi, and Michela Meo. "On the energy consumption computation in Content Delivery Networks". In: *Sustainable Computing: Informatics and Systems* 16 (2017), pp. 56–65. ISSN: 2210-5379. DOI: https://doi.org/10.1016/j.suscom.2017.08.008. URL: https://www.sciencedirect.com/science/article/pii/S2210537916301196.

[7] Yu-Hua Li et al. "An Agent-Based Framework for Matching Buyers and Sellers". In: *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*. 2008, pp. 1–4. DOI: 10.1109/WiCom.2008.2224.

[8] Gaurav Tewari, Jim Youll, and Pattie Maes. "Personalized location-based brokering using an agent-based intermediary architecture". In: *Decision Support Systems* 34.2 (2003). Agents and E-Commerce Business Models, pp. 127–137. ISSN: 0167-9236. DOI: https://doi.org/10.1016/S0167-9236(02)00076-3. URL: https://www.sciencedirect.com/science/article/pii/S0167923602000763.

[9] Kouider Chadli, Goetz Botterweck, and Takfarinas Saber. "The Environmental Cost of Engineering Machine Learning-Enabled Systems: A Mapping Study". In: *Proceedings of the 4th Workshop on Machine Learning and Systems*. 2024, pp. 200–207.

[10] Eva García-Martín et al. "Estimation of energy consumption in machine learning". In: *Journal of Parallel and Distributed Computing* 134 (2019), pp. 75–88. ISSN: 0743-7315. DOI: https://doi.org/10.1016/j.jpdc.2019.07.007. URL: https://www.sciencedirect.com/science/article/pii/S0743731518308773.

[11] Way-Siang Goh et al. "Evaluation of recommendation models for matchmaking". In: *Intelligent Computing & Optimization*. Lecture notes in networks and systems. Cham: Springer International Publishing, 2023, pp. 843–852.

[12] Umesh Bellur and Roshan Kulkarni. "Improved matchmaking algorithm for semantic web services based on bipartite graph matching". In: *IEEE International Conference on Web Services (ICWS 2007)*. Salt Lake City, UT, USA: IEEE, July 2007.

[13] Yoon-Joo Park. "An Adaptive Match-Making System reflecting the explicit and implicit preferences of users". en. In: *Expert Syst. Appl.* 40.4 (Mar. 2013), pp. 1196–1204.

[14] Avita Katal, Susheela Dahiya, and Tanupriya Choudhury. "Energy efficiency in cloud computing data centers: a survey on software technologies". In: *Cluster Computing* 26.3 (2023), pp. 1845–1875.

[15] Salil Bharany et al. "A systematic survey on energy-efficient techniques in sustainable cloud computing". In: *Sustainability* 14.10 (2022), p. 6256.

[16] Ke Xing, Wei Qian, and Atiq Uz Zaman. "Development of a cloud-based platform for footprint assessment in green supply chain management". In: *Journal of cleaner production* 139 (2016), pp. 191–203.

[17] Kadan Lottick et al. "Energy Usage Reports: Environmental awareness as part of algorithmic accountability". In: *arXiv preprint arXiv:1911.08354* (2019).

[18] İbrahim Şanlıalp, Muhammed Maruf Öztürk, and Tuncay Yiğit. "Energy Efficiency Analysis of Code Refactoring Techniques for Green and Sustainable Software in Portable Devices". In: *Electronics* 11.3 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11030442. URL: https://www.mdpi.com/2079-9292/11/3/442.

[19] Athanasios Kiourtis et al. "UI/UX Sustainable Design: Best Practices for Applications CO2 Emissions Reduction". In: *2024 9th International Conference on Smart and Sustainable Technologies (SpliTech)*. 2024, pp. 01–06. DOI: 10.23919/SpliTech61897.2024.10612495.

[20] Paige Wenbin Tien et al. "Machine Learning and Deep Learning Methods for Enhancing Building Energy Efficiency and Indoor Environmental Quality – A Review".

In: *Energy and AI* 10 (2022), p. 100198. ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2022.100198. URL: https://www.sciencedirect.com/science/article/pii/S2666546822000441.