Design and Carbon Footprint Analysis of a Web-Based Assistive Application

Aboli Sontakke[1], Sayali Suradkar[2], Sanskruti Tirvir[3], Isha Save[4], Prof. Shriya Mundhe[5]

[1234] Student, Dept. of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

[5] Professor, Dept. of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India

---***---

Abstract

The rapid growth of digital technologies has transformed communication, accessibility, and information sharing. Web-based assistive tools—such as image-to-text conversion, speech transcription, text-to-speech systems, and content highlighting—enable individuals with visual, auditory, or cognitive challenges to engage effectively, promoting digital inclusion. While their social benefits are well recognized, the environmental impact of these technologies is often overlooked. Every digital interaction consumes computational resources that require electricity, generating carbon emissions depending on the energy source.

This study evaluates the carbon footprint of web-based assistive applications using a framework that estimates emissions based on the execution time of various assistive modules. Energy consumption is monitored in real time using average power metrics, and carbon emissions are calculated according to standard grid intensity values. A centralized dashboard tracks and visualizes emissions for each module, enhancing transparency and environmental awareness.

Experimental results indicate that computationally intensive tasks, such as optical character recognition and live transcription, produce higher emissions than lighter text processing tasks. A clear correlation is observed between task duration and carbon output, validating the time-based estimation approach. Even brief digital operations generate measurable emissions, which can accumulate significantly when multiple modules run consecutively. This research highlights the importance of incorporating carbon-awareness into software design, supporting environmentally responsible and sustainable development of digital systems.

Key Words: Digital Carbon Footprint, Green Computing, Web-Based Assistive Technologies, Carbon Emission Estimation, Energy Modeling, Sustainable Software Engineering.

1. INTRODUCTION

The rapid growth of digital technologies has changed nearly every part of modern life, including communication, education, healthcare, and accessibility. Web-based assistive technologies like image-to-text converters, text-to-speech systems, live transcription tools, and visual content readers have greatly improved digital inclusivity. These applications help people with visual, auditory, and cognitive challenges participate more effectively in the digital world. As digital

accessibility keeps evolving, more users are depending on these assistive platforms. however, the environmental impact of these digital systems is rarely discussed. Every digital action, whether loading a webpage, processing an image, running a speech model, or converting text, uses computational resources. These tasks require electricity for processing power, memory, storage, and network communication. The electricity that computing devices use comes from energy sources that may release carbon emissions, depending on the local energy mix. Thus, even software-based actions indirectly contribute to global greenhouse gas emissions.

The idea of a digital carbon footprint refers to the total greenhouse gas emissions created by digital activities and online systems. While there has been considerable research on large-scale data centers and cloud systems, less attention has been paid to carbon emissions at the application level. Web- based tools run on personal devices also use significant amounts of energy, especially when they perform demanding tasks like optical character recognition, real-time speech processing, or dynamic content rendering. When considered across thousands or millions of users, these seemingly small emissions can add up to a substantial environmental impact.

2. RELATED WORK

As awareness of climate change and sustainable development increases, researchers are looking into the environmental impact of digital technologies. Several studies have focused on measuring energy use and carbon emissions linked to computing systems, data centers, and internet infrastructure. However, analyzing the carbon footprint at the application level is still a developing area of research.

Early research in green computing mainly focused on energy- efficient hardware and optimizing data centers. Studies on cloud computing environments emphasized the high energy demand of large server infrastructures. They suggested techniques like virtualization and load balancing to lower energy use. This work laid the groundwork for understanding the environmental cost of digital services.

Later research shifted toward software-level energy profiling. Researchers showed that inefficient algorithms, too many background processes, and unoptimized web scripts lead to extra energy use. Energy-aware programming techniques were introduced to reduce processor use and lower power draw. These studies highlighted how software design choices directly affect energy consumption.

Recently, web sustainability has attracted attention as internet use continues to rise globally. Researchers have analyzed the carbon footprint of websites based on page size, data transfer volume, and hosting infrastructure. Studies found that media- heavy websites with large scripts, images, and third-party integrations consume more energy when loading and rendering. Tools and frameworks were created to estimate website carbon emissions by combining data transfer metrics with average grid carbon intensity values.

Other research efforts have focused on estimating carbon emissions using execution-time models. These models calculate energy consumption by multiplying processing time by average device power ratings. Carbon emissions are then derived by applying regional carbon intensity factors. Such methods allow for practical estimation without needing special hardware sensors.

While the existing literature offers valuable insights into digital sustainability, there is limited research on assistive web technologies. Most studies focus on large systems or static website analysis instead of dynamic, computation-intensive assistive modules like optical character recognition, speech processing, and real-time

In recent years, sustainability has become a major focus in engineering. However, sustainable software development is still a developing area of research. Developers often prioritize speed and usability, but environmental efficiency is rarely taken into account during design and implementation. There is a growing need to build carbon awareness directly into digital platforms so that both users and developers can better understand the environmental cost of their activities.

This research aims to fill this gap by proposing a carbon footprint estimation framework for web-based assistive technologies. The proposed model calculates energy consumption based on execution time and estimated system power usage. Carbon emissions are then calculated using standard grid carbon intensity values. The framework works across multiple assistive modules, allowing for both individual emission measurement and overall tracking through a centralized dashboard.

The main goal of this study is to measure the environmental impact of commonly used assistive web applications and raise awareness of sustainable computing practices. By showing that digital accessibility tools also produce measurable carbon emissions, this research encourages the adoption of greener software design principles and responsible digital use.

3. METHODOLOGY

3.1 Carbon Estimation Model

The proposed system estimates carbon emissions generated by web-based assistive technologies using an execution-time-based energy modeling approach. Instead of relying on external hardware sensors or server-level monitoring tools, the framework calculates emissions directly within the application during runtime. For each assistive module (such as image-to-text, speech transcription, or text-to-speech), the execution duration is measured using a high-resolution timer. The start time is recorded before the computational task begins, and the end time is captured after the task completes. The difference between these timestamps represents the processing duration.

Energy consumption is estimated by multiplying the execution duration by the assumed average system power consumption. This provides an approximation of

the electrical energy consumed during task execution.
The carbon emission is then calculated by applying a carbon intensity factor, which represents the amount of carbon dioxide emitted per kilowatt-hour (kWh) of electricity generated. This approach allows lightweight, real-time estimation without complex hardware integration.

3.2 Mathematical Formulation

The carbon estimation process follows the equations below:

1. **Energy Consumption Calculation**

$$Energy(kWh) = \frac{Power(W) \times Time(seconds)}{1000 \times 3600}$$

Where:

- Power is the estimated system power consumption in watts
- Time is the execution duration in seconds

2. **Carbon Emission Calculation**

$$CO_2Emission(grams) = Energy(kWh) \times CarbonIntensity\left(\frac{gCO_2}{kWh}\right)$$

In this research, a standard average carbon intensity value is used to represent grid emissions. The final emission value is displayed in grams for better interpretability at small scales.

3.3 Cumulative Emission Tracking

The total cumulative carbon emission generated during a session is calculated as the summation of emissions from all executed assistive modules:

$$\text{Total CO}_2 = \sum_{i=1}^{n} CO_{2,i}$$

Where
$CO_{2,i}$ represents the carbon emission (in grams) generated by the *i-th* module execution, and
n denotes the total number of module executions during the session.
Since each module emission is derived from its corresponding energy consumption, it can be expressed as:

$$CO_{2,i} = E_i \times CI$$

where
E_i is the energy consumed by the *i-th* module (in kWh), and
CI is the carbon intensity factor (gCO_2/kWh).
Substituting into the cumulative equation:

$$\text{Total CO}_2 = \sum_{i=1}^{n} (E_i \times CI)$$

Further, energy consumption for each module is calculated using:

$$E_i = \frac{P \times t_i}{1000}$$

where
P is the average system power consumption (in Watts), and
t_i is the execution time of the *i-th* module (in hours).
Therefore, the final cumulative carbon emission formula becomes:

$$\text{Total CO}_2 = \sum_{i=1}^{n} \left(\frac{P \times t_i}{1000} \times CI\right)$$

4. SYSTEM ARCHITECTURE

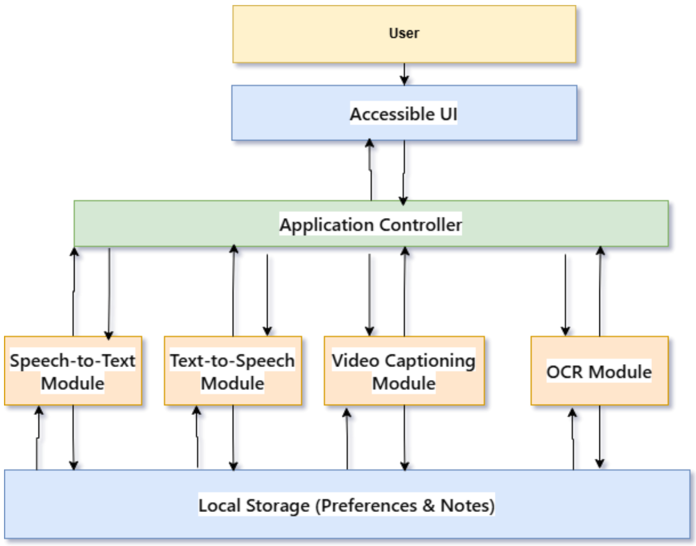


Fig: Architecture Diagram

The system architecture consists of three primary components:

1. **Assistive Modules**
   Independent web-based tools such as image-to-text conversion, speech transcription, and text-to-speech processing.

2. **Carbon Monitoring Service**
   A centralized service that:
   - Receives execution duration from modules
   - Calculates energy consumption

- Computes carbon emissions
- Maintains cumulative totals

3. **Carbon Dashboard**
   A visualization interface that displays:
   - Cumulative CO_2 emissions
   - Total energy consumption
   - Estimated system load

Each module sends its execution duration to the carbon monitoring service, ensuring consistent and centralized emission calculations across the application.

4. **Assumptions**

Since direct hardware-level power monitoring is not implemented, the following assumptions are made:

4.1 The system operates at an average power consumption of approximately 150 W during active processing.

4.2 A constant grid carbon intensity value is used for emission calculations.

4.3 Network-related emissions and server-side cloud processing are not separately measured.

4.4 The estimation focuses on client-side computational energy consumption.

These assumptions provide a simplified yet practical estimation model suitable for application-level carbon tracking.

5. RESULTS AND DISCUSSION

This section presents the emission values obtained from different assistive modules and analyses their environmental impact. The goal is not only to report numerical results but also to interpret their significance in terms of sustainable computing.

5.1 Individual Module Emissions

Each assistive module was executed independently, and carbon emissions were calculated based on execution duration.

5.1.1 Image-to-Text Module

- Energy Used: 0.000015 kWh
- CO_2 Emission: 0.006488 grams

The image-to-text module involves optical character recognition processing, which requires image decoding, text extraction, and rendering. The processing duration directly influenced energy consumption and emission output.
Similarly, other modules such as speech transcription and text-to-speech conversion generated emissions proportional to their execution time and computational complexity.
These results confirm that even short-duration computational tasks generate measurable carbon emissions.

5.1.2 Speech Transcription Module

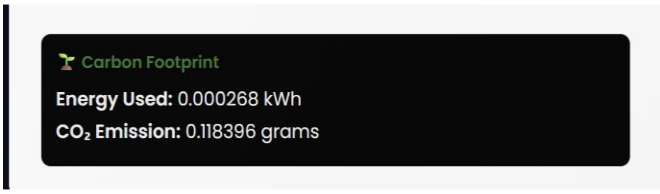
The speech transcription module performs real-time audio capture and converts speech signals into textual data using browser-based recognition engines. This process requires continuous audio sampling, signal processing, and text generation. Compared to static tasks, real-time processing leads to higher processor engagement, resulting in comparatively increased energy consumption and carbon emissions. The emission values observed confirm that continuous processing tasks contribute more significantly to digital carbon footprint.

Furthermore, this framework encourages future improvements such as:

- Dynamic power estimation instead of fixed power assumptions
- Regional carbon intensity adaptation
- Cloud-side emission monitoring

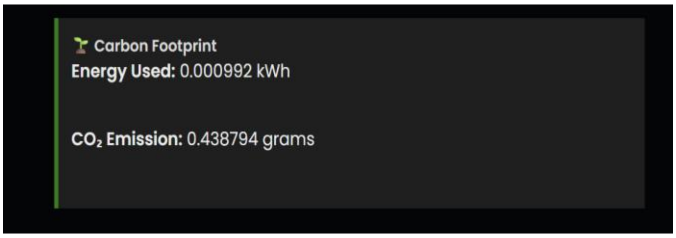
Overall, the findings highlight the importance of embedding sustainability considerations within software development practices.

5.1.3Text-to-Speech Module



The text-to-speech module converts written input into synthesized speech output. This involves text parsing, phoneme generation, and audio waveform synthesis. Although less computation-intensive than real-time transcription, the audio generation process still requires measurable processing power. The recorded emissions indicate moderate energy consumption proportional to the length of the input text and duration of playback.

5.1.4 Highlight Reader Module



The highlight reader module performs lightweight text processing operations such as content selection, rendering, and visual highlighting. Since this task does not involve heavy signal processing or image decoding, the computational load remains relatively low. Consequently, the observed carbon emissions were minimal compared to OCR and speech-based modules. This demonstrates how algorithmic complexity directly influences environmental impact.

5.1.5 Live Speech Transcription Module

The live speech transcription module performs continuous real-time audio capture and converts spoken input into textual output. Unlike static modules that process a fixed input, this feature requires ongoing audio sampling, signal buffering, speech recognition processing, and dynamic text rendering.

Because the module operates in real time, the processor remains actively engaged throughout the duration of speech input. This sustained computational activity leads to higher energy consumption compared to short-duration tasks such as text highlighting or simple rendering operations.

The emission values observed for live transcription were comparatively higher than lightweight modules, primarily due to:

- Continuous microphone input processing
- Real-time speech-to-text conversion
- Background recognition engine activation
- Dynamic text updating

These findings indicate that real-time assistive technologies contribute more significantly to digital carbon footprint than static or single-execution tasks.

5.3 Cumulative Dashboard Results

When multiple assistive modules were executed sequentially, the centralized carbon dashboard displayed the following cumulative session-level values:

- **Total CO_2 Emissions:** 2.401600 g CO_2
- **Total Energy Consumption:** 0.005431 kWh
- **Estimated System Load:** 150 W

The cumulative emission values are noticeably higher than those observed for individual modules. This difference occurs because the dashboard aggregates emissions from all executed modules during the session rather than displaying isolated task-level measurements. Unlike individual module outputs, which represent emissions for a single computational event, the dashboard continuously sums emissions generated across multiple interactions. As a result, the cumulative values provide a more realistic representation of the total environmental impact during practical usage.

It is also important to note that even if a user actively interacts with only one visible module, background processes, previously executed modules, or prolonged session activity may contribute to the overall cumulative total unless the system is explicitly reset. This highlights how digital systems can continue consuming energy beyond immediate visible interactions.

5.2 Analytical Interpretation

The cumulative emission value of 2.401600**g CO_2** demonstrates that while individual module

emissions may appear negligible, repeated or sequential usage significantly increases total environmental impact. This reinforces the importance of session-level monitoring rather than focusing solely on isolated execution events.
Furthermore, the dashboard-based aggregation approach provides:

- Better transparency of total digital energy consumption
- Improved sustainability awareness for users
- A measurable way to evaluate optimization strategies

Overall, cumulative tracking offers a broader sustainability perspective by capturing the combined environmental effect of multiple assistive operations within a single user session.

5.4 Comparative Analysis

Based on the experimental observations, several meaningful insights were derived regarding the environmental behavior of different assistive modules.
First, modules involving continuous or real-time processing, such as live transcription, generated comparatively higher carbon emissions than short-duration static tasks. This is primarily due to sustained processor utilization during continuous audio capture, speech recognition, and dynamic text rendering. In contrast, lightweight operations such as text highlighting or brief content processing required minimal computational engagement and therefore resulted in lower emissions.
Second, a direct proportional relationship was observed between execution time and energy consumption. As execution duration increased, energy usage increased correspondingly, leading to higher calculated CO_2 emissions. This confirms the validity of the execution-time-based estimation model used in this research.
Third, cumulative tracking provided a more comprehensive representation of environmental impact compared to isolated module measurements. While individual emission values appeared relatively small, session-level aggregation revealed a measurable carbon footprint. This demonstrates that sustainability analysis must consider repeated and prolonged interactions rather than single computational events.
Finally, although emission values were recorded in grams and may seem negligible at the individual level, large-scale deployment across thousands or millions of users could significantly amplify the overall environmental footprint. Therefore, even minor efficiency improvements at the application level can have meaningful impact when scaled.

7. DISCUSSION

The results demonstrate that web-based assistive technologies, while socially beneficial and essential for digital inclusivity, are not environmentally neutral. Every computational process—whether image decoding, speech processing, or text rendering—requires electrical energy, which indirectly contributes to carbon emissions depending on the energy source.
However, it is important to contextualize the findings. The measured emissions per session remain relatively small, indicating that assistive technologies do not pose high immediate environmental risk at the individual level. The primary contribution of this study lies not in highlighting excessive emissions, but in revealing their existence and making them measurable.
By integrating carbon estimation directly within assistive applications, this research introduces environmental awareness into the software development lifecycle. Developers are encouraged to consider algorithmic efficiency, execution duration, and processing optimization as sustainability factors alongside performance and usability.
Furthermore, the study demonstrates that transparency plays a key role in sustainable computing. When users can visualize cumulative emissions through a dashboard, they gain a clearer understanding of the environmental impact of their digital interactions. Such awareness-driven systems can promote responsible usage behavior and encourage greener software design practices.
In summary, the findings reinforce the need to incorporate sustainability metrics into application-level design, particularly as digital accessibility tools continue to expand globally.

5. SUSTAINABILITY RELEVANCE

Sustainability in computing has become increasingly important as digital technologies expand globally. While assistive technologies enhance accessibility and social inclusion, it is equally essential to ensure that these digital solutions align with environmental sustainability goals. The proposed carbon footprint monitoring framework contributes to this objective by promoting awareness of energy-efficient digital practices.
This research directly supports the following United Nations Sustainable Development Goals (SDGs):

SDG 7: Affordable and Clean Energy

By estimating the energy consumption of web-based applications, the system highlights the importance of efficient energy usage in digital platforms. Encouraging optimized software design helps reduce unnecessary power consumption, thereby contributing to responsible energy utilization.

SDG 9: Industry, Innovation, and Infrastructure

The integration of carbon monitoring into assistive technologies promotes sustainable innovation in software engineering. Instead of focusing solely on performance and usability, this research introduces environmental impact as an additional design parameter. Such an approach strengthens sustainable digital infrastructure development.

SDG 12: Responsible Consumption and Production

Digital resources, including computational power and electricity, are forms of consumption. By making carbon emissions visible to users, the system fosters responsible digital usage behavior. Awareness-driven systems can reduce redundant processing and encourage efficient application interaction.

SDG 13: Climate Action

Although individual emissions from web applications are relatively small, their cumulative global impact can be significant. Monitoring and minimizing digital carbon footprints support broader climate action initiatives by reducing indirect greenhouse gas emissions associated with electricity generation.
Overall, this research bridges the gap between digital accessibility and environmental sustainability. It demonstrates that socially beneficial technologies can also be designed with ecological responsibility in mins.

8. APPLICATION INTERFACE

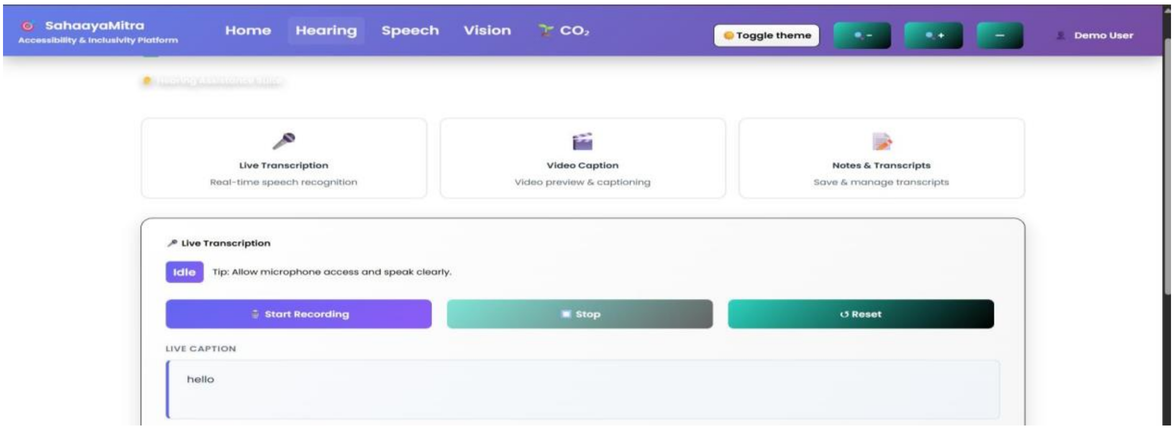


Fig 1: Home Page of Application

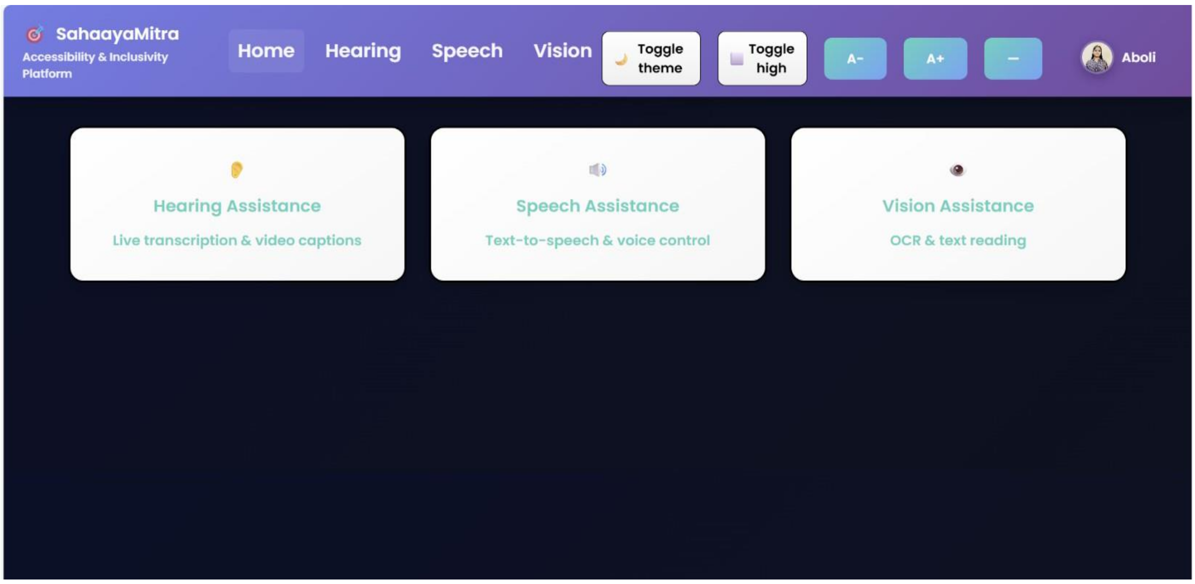


Fig 2: Hearing Module

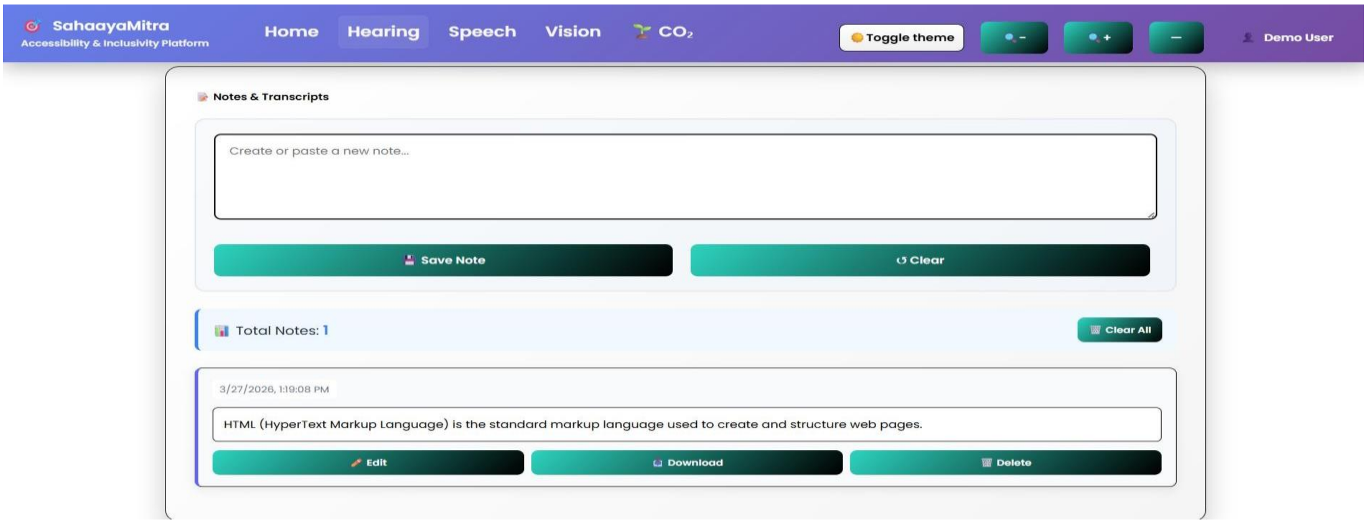


Fig 3: Speech Module

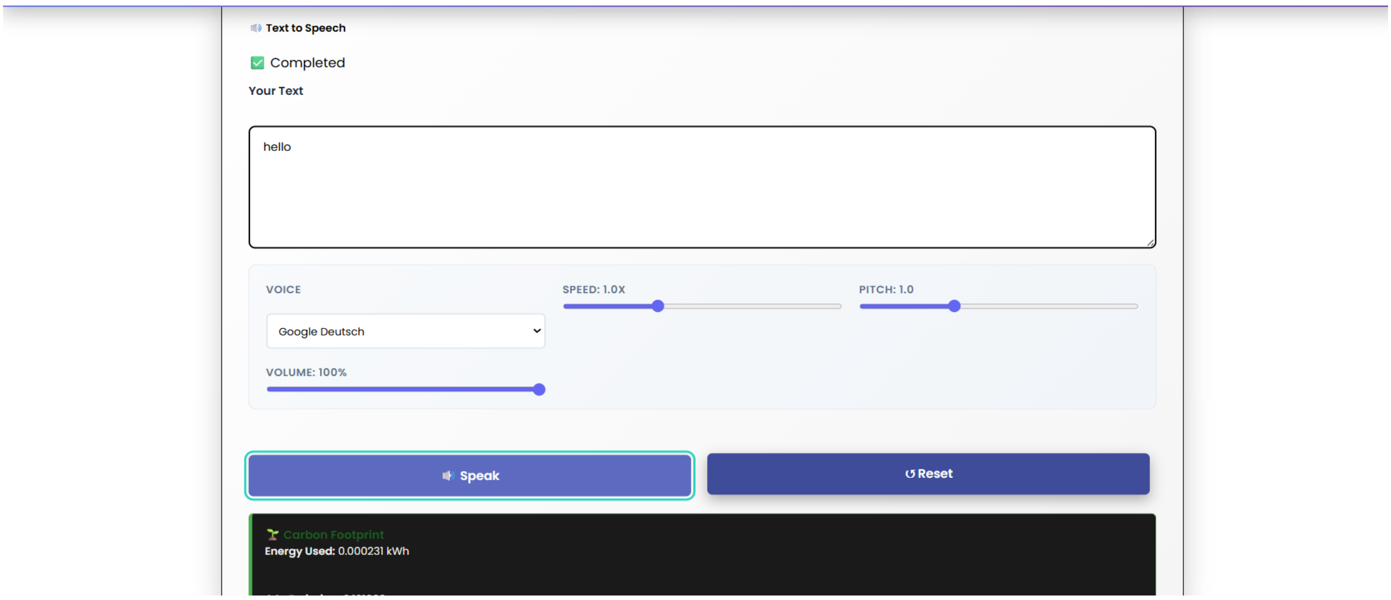


Fig 4: Speech Module

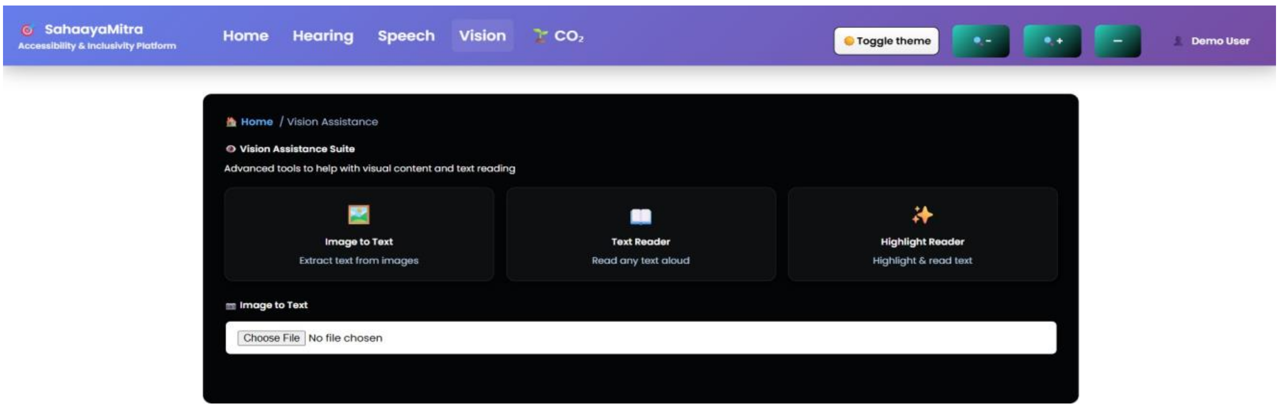


Fig 5: Vision Module

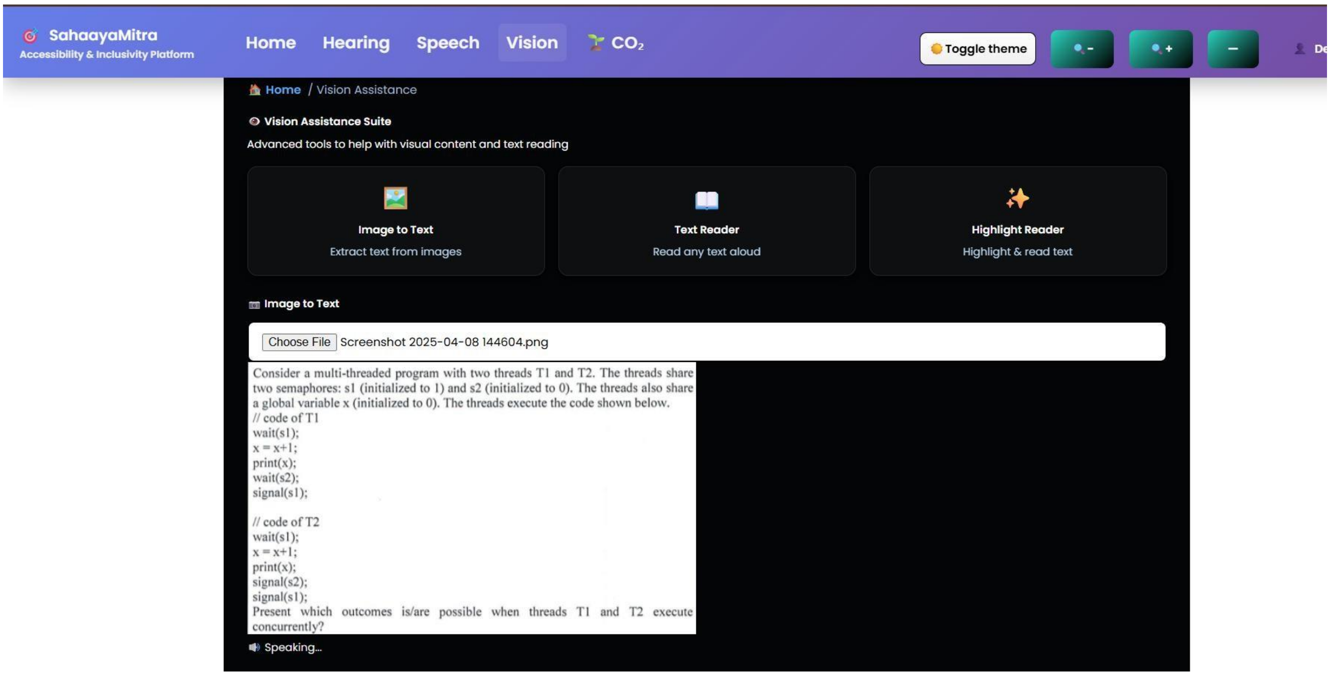


Fig 6: Vision Module

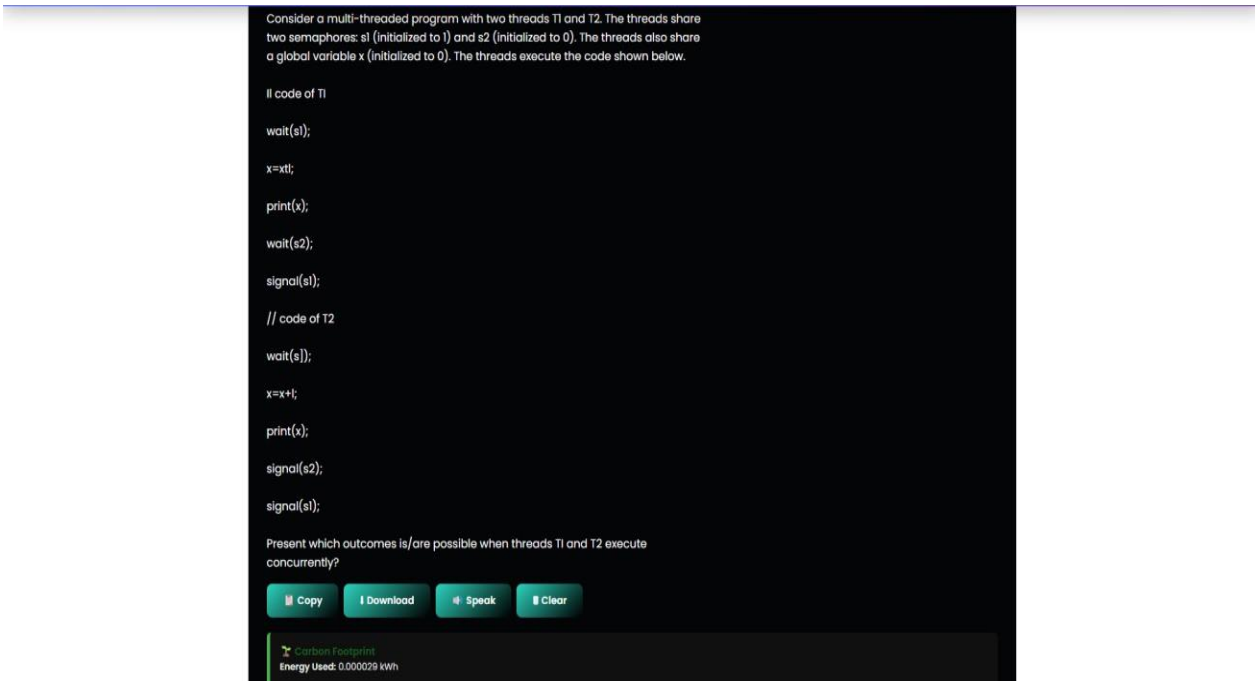


Fig 7: Vision Module

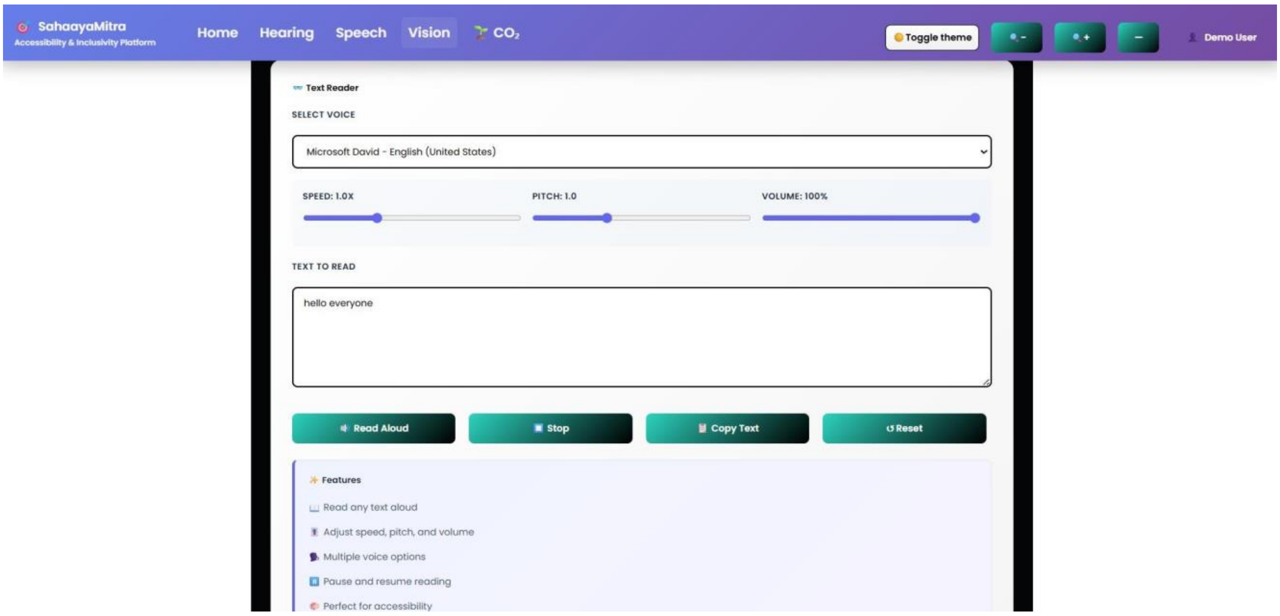


Fig 8: Vision Module

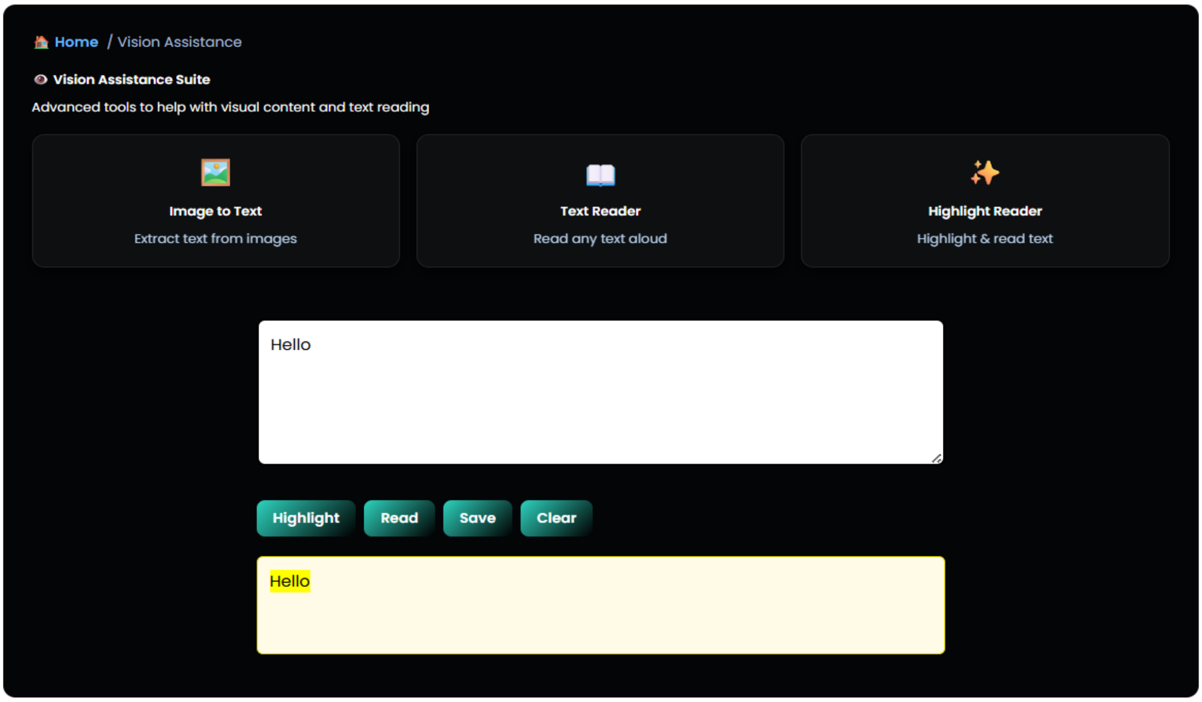


Fig 9: Vision Module

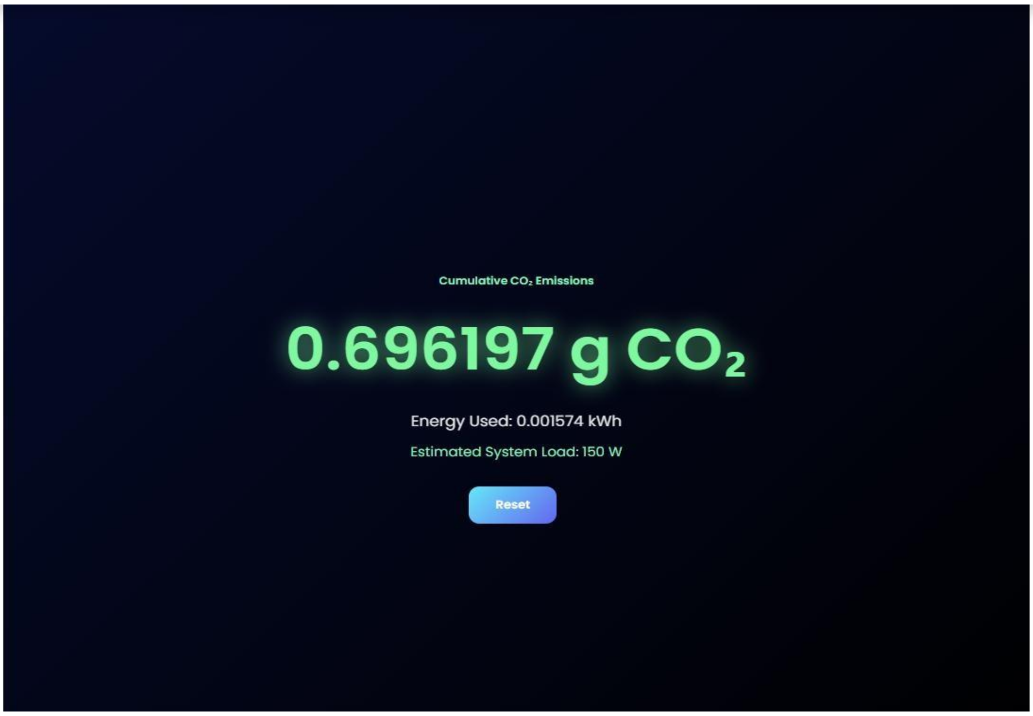


Fig 10: Carbon footprint estimation

9. CONCLUSION

This research presented a digital carbon footprint analysis framework for web-based assistive technologies. While such technologies significantly improve accessibility and inclusivity, their environmental impact is often overlooked. The proposed system integrates a lightweight carbon estimation model directly into the application, enabling real- time monitoring of energy consumption and CO_2 emissions.

By measuring execution time and applying power and carbon intensity assumptions, the framework estimates both individual module emissions and cumulative session-level impact. Experimental results demonstrated that even short computational tasks generate measurable carbon emissions. Although the values appear small at an individual level, large- scale usage across users and platforms can lead to substantial environmental impact.

The study highlights the importance of embedding sustainability metrics within software systems. Rather than treating carbon analysis as an external audit process, integrating emission tracking at the application level encourages responsible software design and usage behavior. The centralized dashboard further enhances transparency by visualizing cumulative emissions.

However, the current model relies on fixed assumptions for system power consumption and carbon intensity. Future work may focus on dynamic power profiling, real-time regional carbon intensity integration, cloud-side emission monitoring, and automated optimization suggestions for reducing digital carbon footprints.

In conclusion, this research demonstrates that assistive technologies and environmental sustainability are not mutually exclusive. By incorporating carbon awareness into digital systems, developers can contribute toward climate- conscious innovation while maintaining social impact.

10. REFERENCES

[1] M. Tauseef, S. G. Naik, and K. S. Harsha, “SpeakEasy Reader: An OCR-Based Text-to-Speech Device for Enhanced Accessibility in Assisting Reading Challenges,” in Proc. ICCCNT, IIT Mandi, India, 2024.

[2] Y. Joh and J. Lee, “Image Accessibility for Screen Reader Users: A Systematic Review and Roadmap,” Electronics, vol. 10, no. 8, pp. 1–22, Apr. 2023.

[3] A. Kerdar, M. Al-Khalifa, and S. Al-Mutairi, “The Accessibility of Digital Technologies for People with Visual Impairment and Blindness: A Scoping Review,” Discover Computing, vol. 2, no. 1, pp. 1–18, 2024.

[4] R. Patel, S. Mehta, and A. Shah, "Screen Reader AI: A Conversational Web-Accessibility Assistant for Blind and Low- Vision Users," Int. J. Eng. Sci. Inf. Technol., vol. 5, no. 2, pp. 45– 52, 2025.

[5] M. Maashi, H. G. Iskandar, and M. Rizwanullah, "IoT-driven smart assistive communication system for the hearing impaired with hybrid deep learning models for sign language recognition," Scientific Reports, 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-89975-1

[6] X. Chen, "Overview of Speech Recognition Algorithms and Their Applications," Applied and Computational Engineering, vol. 183, pp. 7–13, 2025.

[7] H. Lou et al., "SpeechAgent: An End-to-End Mobile Infrastructure for Speech Impairment Assistance," 2025.

[8] K. Revankar et al., "Sanvaad: A Multimodal Accessibility Framework for ISL Recognition and Voice-Based Interaction," 2025.

[9] A. Beloglazov and R. Buyya, "Energy Efficient Allocation of Virtual Machines in Cloud Data Centers," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, 2012.

[10] L. A. Barroso and U. Hölzle, The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, 2nd ed. Morgan & Claypool, 2013.

[11] S. Naumann, M. Dick, E. Kern, and T. Johann, "The GREENSOFT Model: A Reference Model for Green and Sustainable Software Engineering," Sustainable Computing: Informatics and Systems, vol. 1, no. 4, pp. 294–304, 2011.

[12] T. Mytton, "Digital Carbon Footprint: Estimating the Carbon Emissions of Internet Use," Journal of Cleaner Production, vol. 209, pp. 135–144, 2019.

[13] Website Carbon Calculator, "How Website Carbon Emissions are Calculated." [Online]. Available: https://www.websitecarbon.com

[14] A. Hindle, "Green Mining: A Methodology of Relating Software Change to Power Consumption," in Proc. IEEE MSR, 2012, pp. 78–87.