ChatGPT Carbon Tracker

Raghav Talwar[1], *Konrad Barbers*[1], *Peiyan Liu*[1], *Shalakha R S*[1]
[1] TU Delft, Netherlands
rtalwar@tudelft.nl, kbarbers@tudelft.nl, peiyanliu@tudelft.nl, srudrapatnasur@tudelft.nl

1 INTRODUCTION AND PROBLEM STATEMENT

The rapid integration of large scale AI agents such as ChatGPT has increased, reshaping how individuals and businesses interact in communication, automate tasks, and retrieve information. However, this widespread adoption of these systems raises critical sustainability concerns due to the significant energy consumption that is involved in these operations. As of early 2025, ChatGPT alone reports over 400 million weekly active users, handling over 1 billion daily interactions globally . This exponential growth in user activity underscores the increasing dependency on AI models for real-time language processing.

However, this widespread integration raises critical sustainability concerns. Each user query processed by ChatGPT involves resource intensive operations, such as tokenization, model inference, and response generation executed in energy hungry data centers. While such systems deliver immense utility, they also generate substantial energy costs.

Despite growing awareness of AI's ecological footprint, there remains a critical lack of transparency and tooling that allows users and organizations to understand and mitigate their environmental impact. There is currently no standardized methods to assess carbon emissions at the depth of individual AI interactions. This gap not only limits accountability but also hampers the ability of stakeholders to optimize usage for sustainability.

Given the urgency of addressing climate change and the increasing ubiquity of AI, it is imperative to develop real-time, user-facing mechanisms for quantifying and visualizing the carbon footprint of AI chat interactions. Such tools are essential for promoting environmental accountability and enabling informed decision-making for sustainable AI deployment.

This project introduces a lightweight, browser-based tool that estimates the carbon footprint of ChatGPT sessions in real time. It operates entirely on the client side, using token-level heuristics from DOM observation to approximate emissions. Key contributions an integrated real-time interface, and contextual visualizations that translate carbon output into familiar analogies.

2 RELATED WORK

As large language models (LLMs) like ChatGPT become more widely integrated into daily applications, their environmental and societal impacts are receiving increasing attention. Much of the existing literature has focused on the resource cost of training such models, while the sustainability challenges associated with real-time usage and end-user interaction remain less explored. Our work addresses this area by focusing on inference-level emissions and their visibility to users.

Strubell et al. [1] and Henderson et al. [2] were among the first to examine the energy requirements of NLP model development. They asserted for greater transparency in reporting energy use and carbon emissions, especially in academic research. Lacoste et al. [3] provided practical tools for estimating emissions, including variables such as hardware type and data center location. These contributions helped set a foundation for energy aware machine learning, but remained focused on the development and training stages of a model.

Patterson et al. [4] extended this line of work by analyzing the energy usage of large models like GPT-3 and showing how architectural and hardware choices can improve efficiency. However, their study mainly looked at aggregate emissions and did not address how this information might be presented to users or applied in everyday contexts.

Some more recent studies have begun to consider the impact of inference. Vanderbauwhede [5] estimated that search engines enhanced with LLMs can increase the energy cost per query by over **60 times** compared to traditional search, underlyning the need to measure emissions beyond just the training.

Khowaja et al. [6] proposed SPADE, a framework for evaluating AI systems through multiple lenses including sustainability, fairness, and accessibility. Their work highlights that making AI more sustainable is not only a technical challenge but also a matter of individual importance and social responsibility.

Despite these efforts, few tools make the environmental impact of AI visible to end users. Most approaches rely on backend data or infrastructure level metrics, which are not accessible in real time during user interaction. There is a gap in lightweight, user-facing systems that can promote awareness and encourage more responsible usage.

Our work contributes to filling this gap by developing a browser extension that estimates the carbon footprint of ChatGPT sessions based on token usage. Unlike backend-heavy methods, our approach works entirely on the client side, bringing sustainability metrics directly to the interface where users engage with AI. This reflects a shift toward incorporating sustainability not just in how software is built, but also in how it is used.

3 SOLUTION PROPOSAL

To address the lack of transparency surrounding the environmental impact of AI model usage, this work introduces a browser based extension that estimates and displays the carbon footprint of user interactions with large language models such as ChatGPT. The goal is to provide realtime and session based emissions tracking that encourages environmentally conscious usage by making energy consumption and carbon emissions visible and interpretable to users.

The proposed solution is based on per session inference level estimation, an aspect largely overlooked in existing sustainability research. While much of the literature has focused on the energy cost of training large models, the inference stage responsible for

Raghav Talwar[1], Konrad Barbers[1], Peiyan Liu[1], Shalakha R S[1]
[1] TU Delft, Netherlands
rtalwar@tudelft.nl, kbarbers@tudelft.nl, peiyanliu@tudelft.nl, srudrapatnasur@tudelft.nl

serving billions of queries daily remains underexplored despite its substantial overall footprint. This framework targets that gap by observing the interaction locally in the browser, allowing energy and emissions estimates to be derived without requiring backend integration or server-side telemetry.

The methodology relies on estimating the number of tokens generated during each chat session, using word and character-level heuristics. Each token is associated with an empirical energy cost, and total energy consumption is converted into carbon emissions using a fixed carbon intensity factor. These emissions estimates are further contextualized through relatable comparisons, such as the number of equivalent Google searches or minutes of video streaming, which help users interpret abstract environmental data in familiar terms.

To present this information effectively, a lightweight user interface is integrated into the chat environment, presenting metrics such as token count, carbon emissions, and session duration. The design prioritizes minimal disruption while ensuring that sustainability metrics are continuously available throughout the interaction.

By enabling real-time carbon traceability at the level of individual AI interactions, this solution makes three key contributions.

- It introduces a novel approach for modeling inference time emissions without relying on privileged system access.
- It brings environmental accountability directly into the user interface, encouraging reflection and behavior change.
- It lays the foundation for better integration of sustainability metrics into AI assisted workflows, supporting future efforts to align high utility AI tools with long-term environmental responsibility.

4 IMPLEMENTATION

This section details the implementation of the proposed **ChatGPT Carbon Tracker** solution. It begins with an overview of the extension's functionality, followed by a discussion of its architectural design, the computational model for carbon footprint estimation, chat monitoring and user interface integration and performance optimizations.

To encourage transparency and community involvement, we release our implementation as an open-source browser extension. The code and documentation are available at **Github**.

4.1 Functional Overview

The **ChatGPT Carbon Tracker** [15][16] is a browser extension designed to estimate and visualize the carbon footprint of ChatGPT interactions in real time. By monitoring token usage from both user inputs and assistant responses, the extension applies an energy estimation model to quantify environmental impact.

The implementation adopts a modular architecture (illustrated in Figure 4), ensuring scalability and maintainability. The core components include:

- **Content Script**: Captures user interactions, extracts text, and calculates token usage based on an established method [10], where every 4 characters correspond to 1 token. The processed data is then prepared for further computation.
- **Background Script**: Manages lifecycle events such as installation and settings, ensuring consistent extension behavior.
- **Popup Interface**: Provides a real-time dashboard that visualizes energy consumption and carbon emissions in an intuitive format.

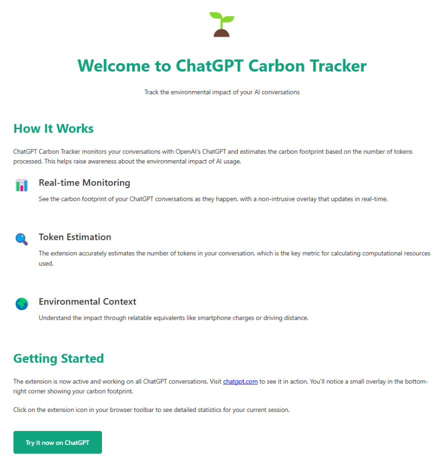


Figure 1: Welcome Screen

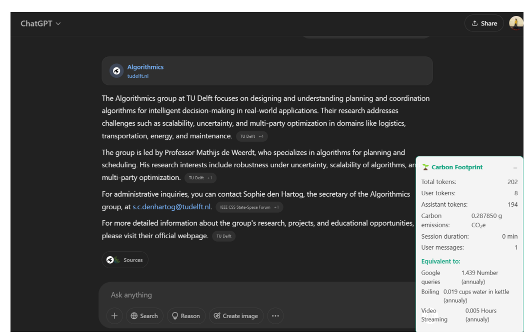


Figure 2: Plugin in action

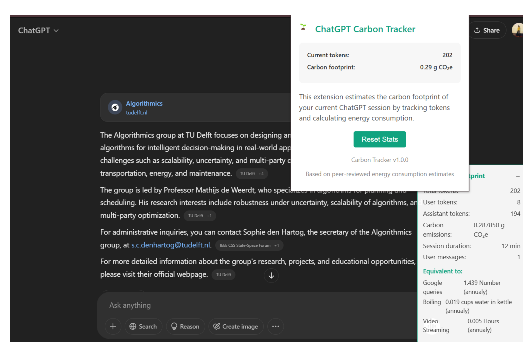


Figure 3: Popup

4.2 Technical Architecture

The extension is designed to balance **performance, accuracy, and maintainability** while operating within browser constraints. The core functionality is structured as follows:

The extension's **manifest.json** file specifies necessary permissions and UI components. It defines the content script, background script, and popup elements while ensuring compliance with browser security policies.

The **content script** dynamically injects monitoring logic into web pages, capturing both user messages and assistant responses. Extracted text is tokenized[10] and processed to estimate energy consumption based on predefined models.

The **background script** applies a mathematical calculation to convert token usage into estimated energy consumption. The estimated energy usage is then mapped to equivalent **carbon emissions**[8] using a standardized emissions factor. Results are stored and managed efficiently to minimize performance overhead.

The **popup interface** loads HTML and JavaScript components that generate a responsive dashboard. Users can view real-time statistics on energy consumption and carbon impact directly within the extension's popup.

This architecture ensures modularity, enabling seamless updates and future enhancements. Additionally, caching and lightweight processing techniques are done to minimize runtime overhead, ensuring a **smooth user experience** without significantly affecting browser performance.

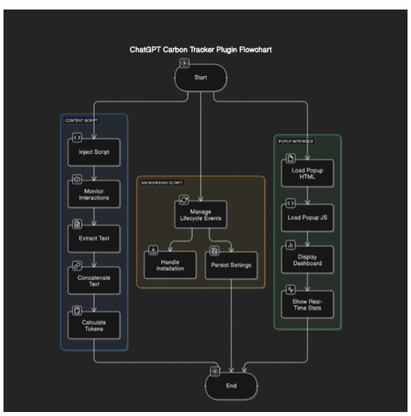


Figure 4: Overview of the carbon estimation process

4.3 Carbon Footprint Estimation Model

The carbon footprint estimation is based on a computational model that correlates token usage with energy consumption. The energy expenditure associated with each interaction is given by:

$$E_{\mathrm{wh}} = T \times W \tag{1}$$

where E_{wh} represents the estimated energy consumption in watt-hours, T is the total number of tokens processed, W denotes the energy consumption per token (0.003 Wh)[Assuming 100 tokens per query.] [7].

The carbon emissions associated with this energy usage are computed as:

$$C_{\mathrm{gCO_2e}} = E_{\mathrm{wh}} \times I \tag{2}$$

where I is the carbon intensity of energy consumption, assumed to be 0.475 kg CO_2e/kWh [11].

To provide users with an intuitive understanding of the carbon impact, the emissions are contextualized using real-world comparisons. The equivalent environmental impact is expressed in terms of the number of cups of water that could be boiled with the same energy [13], the number of trees required to absorb an equivalent amount of CO_2 [14], and the number of google queries it would account for [9]. These equivalencies are derived as follows:

$$\text{Google Queries (Numbers)} = \frac{C_{\mathrm{gCO_2e}}}{0.2} \tag{3}$$

$$\text{Boiling Water Equivalent (cups)} = \frac{C_{\mathrm{gCO_2e}}}{15} \tag{4}$$

$$\text{Video Streaming(Hours)} = \frac{C_{\mathrm{gCO_2e}}}{55} \tag{5}$$

4.4 Real-time Monitoring and User Interface Integration

To ensure accurate real-time tracking of user and assistant messages, the tool needs a process to detect updates dynamically as they appear on the webpage. This is achieved using an **observer pattern**, which allows the extension to monitor changes in the conversation interface efficiently. Specifically, we utilize a **MutationObserver**, a built in web API designed for detecting modifications. By observing a specific HTML element where messages appear, the tracker continuously updates token usage and carbon footprint calculations in synchronization with the conversation flow. The popup interface provides users with a summary of cumulative emissions data and includes a reset function to restart calculations for the current conversation.

The graphical user interface (GUI) which can be referred here [3][**??**][**??**] is designed to be discreet yet informative. The popup interface presents a summary of the ongoing session, including the total tokens processed, estimated energy consumption, and CO_2 emissions. Additionally, a *reset* function is provided, allowing users to clear existing data and start fresh calculations.

4.5 Performance Optimization

To minimize computational overhead while maintaining accuracy, several optimization strategies are employed. The extension applies **debouncing**[12] techniques to avoid redundant recalculations, ensuring that energy usage is computed only after meaningful user interactions. Trivial messages, such as system-generated responses, loading indicators, and model status updates, are systematically excluded from the token count to prevent inflated estimates.

The implementation also incorporates **lazy UI updates**, reducing unnecessary modifications to the Document Object Model (DOM). By updating the carbon footprint display only when significant changes occur, the extension maintains efficiency without compromising usability.

Raghav Talwar[1], Konrad Barbers[1], Peiyan Liu[1], Shalakha R S[1]
[1] TU Delft, Netherlands
rtalwar@tudelft.nl, kbarbers@tudelft.nl, peiyanliu@tudelft.nl, srudrapatnasur@tudelft.nl

5 EVALUATION

To evaluate the **ChatGPT Carbon Tracker**, volunteer participants were asked to use ChatGPT with the plugin installed and fill in a short feedback survey, shown in Figure 7. The participants were predominantly young adult students in the field of computer science at TU Delft, with the rest being working professionals in other fields. This survey was meant to gather first impressions of the participants as well as feedback for further improvements. Overall, they responded positively while still providing insights into areas of improvement for the user experience.

5.1 User Study and Survey Questions

To evaluate the extension, the participants were first asked to use ChatGPT as they usually would with the Carbon Tracker installed in their browser for at least one small conversation. They were not actively monitored during this process, and the choice of the length and content of the conversation were left up to each individual participant.

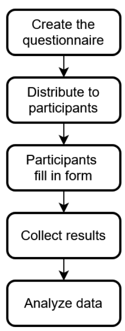


Figure 5: The process used to gather evaluation data.

Following this step, the participants were asked to fill in a short online survey on their experiences of using the extension. The survey consisted of four closed questions and one open question, all of which were required to be filled in to complete it. In the closed questions, participants were asked if they agreed that the tracker made them more aware of the environmental impact of ChatGPT, how intrusive or unintrusive the user interface felt, whether the comparison figures shown were useful to contextualize the emission score, and whether they would want to use the extension in the future; the open question simply asked for further suggestions. No identifying information about the participants was collected. In total eight people submitted a response to the survey.

5.2 Closed Questions

The first question asked participants to rank whether their awareness of ChatGPT's energy usage increased on a five-level Likert scale from "strongly disagree" to "strongly agree". Out of the 8 respondents, 6 indicated that the extension increased their awareness against the environmental impact of ChatGPT. Among these 6, half selected "agree," while the other half selected "strongly agree." These results suggest that the extension was able to effectively raise awareness about the environmental impact of AI.

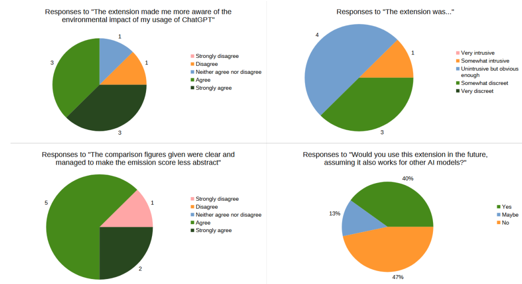


Figure 6: Pie charts of responses to the closed questions.

Asked to classify the extension on a scale from "Very intrusive" to "Very discreet", 4 selected "nintrusive but obvious enough", a further 3 "somewhat discreet", and the last participant selected "somewhat intrusive", implying that while the Carbon Tracker user interface is appealing and useful to users, some would like it to more apparent.

In the third question, respondents were asked to rank the clarity of the comparison emission figures on a scale from "strongly disagree" to "strongly agree". Seven users either agree (5 responses) or strongly agree (2 responses) that the comparisons to the carbon emissions of everyday processes helped make the emissions calculated for the current chat more clear. The last respondent selected the "Strongly disagree" option, further clarifying that the way some of the comparison figures were presented were confusing and worded in an unclear way.

When asked on whether they would use this extension in the future, assuming it also worked for other AI models, six responded with "Yes" and two with "Maybe", implying a mostly positive experience of using the carbon tracker.

5.3 Suggestions

When asked for open feedback, the responses were mostly positive, offering a few suggestions. Three respondents suggested minor UI improvements, two of which also suggested increasing user engagement with more interesting and/or appealing visuals. One respondent, who self-identified as "not a professional in this field", asked for an explanation of the term "token", while another critiqued the use of tokens as the plugin not being sensitive enough to use them for real-time calculations due to time delays.

Overall, the suggestions tended towards minor UI improvements, clarifying of terms and greater user customizability.

6 DISCUSSION

The ChatGPT Carbon Tracker helps address the lack of visibility around the environmental impact of using large language models. While it does not solve the broader issue of carbon emissions from AI, it makes part of the problem visible to users. This aligns with the goal of making software more sustainable not only in how it is built, but also in how it is used.

From our user study, we saw that most people found the tool helpful in raising awareness. However, some users mentioned that technical terms like "token" were unclear. This shows a challenge in designing tools that are understandable to people without a technical background. If sustainability tools are too difficult to interpret, they will not support real individual or social change.

The tool works fully in the browser, which makes it easy to use and deploy. But this also means it uses estimated values rather than actual system data. There is a trade-off between simplicity and accuracy. Even though the estimates are not exact, they are enough to start a conversation and help users think more about their impact.

Overall, the tool does not fully solve the problem of sustainable AI use, but it makes a useful contribution. It offers a simple way for users to become more aware of energy use in real time and demonstrates the potential for developing similar tools in the future.

7 LIMITATIONS

As the design, development and evaluation of this program was limited to only a few weeks, there are multiple aspects of the extension that could be improved or explored further.

Estimation of energy figures. The constants chosen for token-level energy usage and ChatGPT's environmental impact are estimates, especially the energy consumption figure, as no precise data was available. The landing page of the extension cites the sources for these numbers, but they do not account for differences in hardware, data center location, or actual runtime energy usage.

Environmental impact on other GenAIs. The current version of the Carbon Tracker supports only ChatGPT. Although the extension's design allows for reuse, it has not been adapted or tested on other widely used models such as DeepSeek or Midjourney. This limits its generalizability to other generative AI systems.

Choice of carbon emission comparisons. The metrics used for comparison (such as boiling water or video streaming) were selected informally, aiming for relatability and reasonable scale. While most users found these helpful, at least one participant found them confusing or unclear, especially due to missing contextual explanations.

User interface feedback. Several users praised the UI's color scheme and clean design. However, others mentioned minor layout issues such as insufficient padding, unresponsive elements, or the need for more visually engaging interactions. One user also noted that token updates appeared delayed.

Terminology clarity. Some survey responses indicated that users were unfamiliar with terms like "token." Without clearer explanations, non-technical users may not fully understand the reported carbon estimates or how they are calculated.

Evaluation scope. The plugin was evaluated using a short survey with eight participants. While the responses were helpful, the sample was small, and no structured usability testing or behavioral data was collected. This limits the strength of conclusions that can be drawn from the feedback.

8 CONCLUSION

This project demonstrates how sustainability can be meaningfully integrated into user-facing software tools. By developing a browser based extension that estimates and visualizes the carbon footprint of ChatGPT sessions in real time, we address the lack of transparency surrounding the environmental impact of AI usage at the individual level. Unlike prior work that focuses on training efficiency or system-level metrics, our approach makes energy consumption data directly visible and interpretable for users.

The design and implementation of this tool reflect key principles of sustainable software engineering. It supports environmental awareness during the software usage phase, incorporates lightweight monitoring without backend access, and invites users to reflect on the broader implications of their digital behavior. In doing so, it contributes not only to Green software engineering but also to the social and individual dimensions of sustainability, which are increasingly vital in today's software ecosystems.

Through this project, we explored the practical challenges of measuring energy consumption in AI applications and transforming abstract metrics into meaningful insights. While the plugin is currently scoped to ChatGPT, the architecture is generalizable and can be extended to other AI systems. Future improvements could include refining energy estimation models using more accurate and location-aware data, extending the tool to other generative AI platforms, and supporting alternative carbon comparison modes tailored to different user profiles or regions. Conducting large-scale, long-term user studies would also help assess behavioral changes and inform design decisions that make sustainability more actionable.

This project is not only a technical exercise, but also an attempt to help users become more aware of the environmental impact of their AI usage. Looking ahead, future development could include open-sourcing the extension, allowing others to contribute improvements, and exploring how this tool might be used in educational or organizational contexts. By balancing technical design with user-centered sustainability, we hope to support software systems that are both useful and more environmentally responsible.

REFERENCES

[1] Strubell, E., Ganesh, A., & McCallum, A. (2019). Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650. Available at: https://aclanthology.org/P19-1355

[2] Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., & Pineau, J. (2020). Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *Journal of Machine Learning Research*, 21(248), pp. 1–43. Available at: https://arxiv.org/abs/2002.05651

[3] Lacoste, A., Luccioni, A., Schmidt, V., & Dandres, T. (2019). Quantifying the Carbon Emissions of Machine Learning. *arXiv preprint arXiv:1910.09700*. Available at: https://arxiv.org/abs/1910.09700

[4] Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon Emissions and Large Neural Network Training. *arXiv preprint arXiv:2104.10350*. Available at: https://arxiv.org/abs/2104.10350

[5] Vanderbauwhede, W. (2024). Estimating the Increase in Emissions caused by AI-augmented Search. *arXiv preprint arXiv:2407.16894*. Available at: https://arxiv.org/abs/2407.16894

[6] Khowaja, S. A., et al. (2023). ChatGPT Needs SPADE (Sustainability, PrivAcy, Digital divide, and Ethics) Evaluation: A Review. *arXiv preprint arXiv:2305.03123*. Available at: https://arxiv.org/abs/2305.03123

[7] Epoch AI. "How much energy does ChatGPT use?" Available at: https://epoch.ai/gradient-updates/how-much-energy-does-chatgpt-use

[8] Arbor. "AI's Environmental Impact: Calculated and Explained" Available at: https://www.arbor.eco/blog/ai-environmental-impact

[9] Search Engine Land. "Calculating The Carbon Footprint Of A Google Search" Available at: https://searchengineland.com/calculating-the-carbon-footprint-of-a-google-search-16105

[10] OpenAI Help. "What are tokens and how to count them?" Available at: https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them

[11] Arbor. "AI Environmental Impact." Available at: https://www.arbor.eco/blog/ai-environmental-impact

Raghav Talwar[1], *Konrad Barbers*[1], *Peiyan Liu*[1], *Shalakha R S*[1]
[1] TU Delft, Netherlands
rtalwar@tudelft.nl, kbarbers@tudelft.nl, peiyanliu@tudelft.nl, srudrapatnasur@tudelft.nl

[12] GeeksforGeeks "Debouncing in JavaScript" Available at: https://www.geeksforgeeks.org/debouncing-in-javascript/

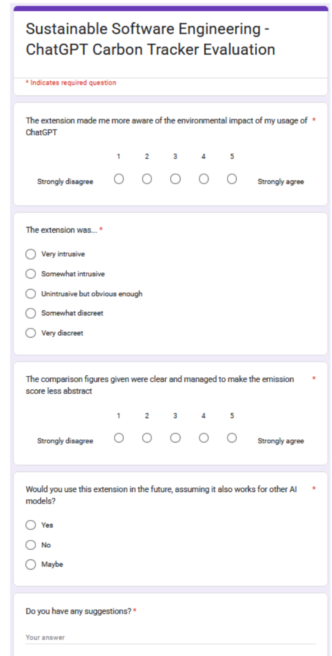
[13] The Guardian. "Energy Consumption of Boiling a Kettle." Available at: https://www.theguardian.com/environment/ethicallivingblog/2008/mar/07/keepyourkettleincheck

[14] Greenly "The Carbon Cost of Streaming" Available at: https://greenly.earth/en-gb/leaf-media/data-stories/the-carbon-cost-of-streaming

[15] https://chromewebstore.google.com/detail/chatgpt-carbon-tracker/cclkbhgeajbcehglecbnhpchnfhpakdk

[16] https://addons.mozilla.org/en-US/firefox/addon/chatgpt-carbon-tracker/

9 APPENDIX


Sustainable Software Engineering - ChatGPT Carbon Tracker Evaluation

* Indicates required question

The extension made me more aware of the environmental impact of my usage of ChatGPT *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

The extension was... *

- ○ Very intrusive
- ○ Somewhat intrusive
- ○ Unintrusive but obvious enough
- ○ Somewhat discreet
- ○ Very discreet

The comparison figures given were clear and managed to make the emission score less abstract *

	1	2	3	4	5	
Strongly disagree	○	○	○	○	○	Strongly agree

Would you use this extension in the future, assuming it also works for other AI models? *

- ○ Yes
- ○ No
- ○ Maybe

Do you have any suggestions? *

Your answer

Figure 7: The full survey presented to testers