International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Eco Friendly Checker: Leveraging Multimodal Data Fusion and Reinforcement-Based Metrics for Sustainable Web Browsing

Ramya B N[1], Aryan M[2], K Vihar[3], Mohith Kumar N[4], Nitish G S[5] *

[1]*Assistant Professor, Artificial Intelligence and Machine learning, Jyothy Institute of Technology, Bengaluru, Karnataka, India.*
[2,3,4,5] *Student, Artificial Intelligence and Machine learning, Jyothy Institute of Technology, Bengaluru, Karnataka, India.*

A B S T R A C T

In recent years, increasing environmental awareness has led consumers to seek more sustainable and eco-friendly products. However, verifying the ecological impact of a product while shopping online remains a challenge. This paper presents a browser extension tool, "Eco Friendly Product Checker," designed to assist users in evaluating the sustainability of products listed on e-commerce websites. The system integrates front-end JavaScript logic to extract product details such as name, price, and description from web pages. These details are then sent to a Node.js-based backend server, which analyzes the information and returns a sustainability rating along with environmentally-friendly product alternatives. The extension seamlessly integrates into the user's browser, offering real-time feedback and recommendations without interrupting the shopping experience. This tool aims to empower consumers with greater transparency, promote responsible purchasing habits, and contribute to more sustainable e-commerce practices.

Keywords: Sustainability, Chrome Extension, Product Rating, Green Alternatives, Eco-Friendly Shopping

1. INTRODUCTION

The rise of e-commerce has significantly transformed the way consumers purchase goods, offering convenience, variety, and competitive pricing. However, this rapid digitalization of shopping has also led to increased consumption and limited visibility into the environmental impact of purchased products. While consumers are becoming more conscious of sustainability, most online platforms do not provide adequate tools to assess the eco-friendliness of items.

To bridge this gap, we present the "Eco-Friendly Product Checker," a Chrome browser extension that allows users to evaluate the sustainability of products they browse online. This tool extracts product information directly from e-commerce websites and interacts with a backend system to generate a sustainability rating. The extension also provides users with eco-friendly alternative suggestions, helping them make more informed and environmentally responsible decisions.

Our goal is to create a seamless and accessible solution that enhances user awareness and promotes sustainable consumption habits in the digital shopping space.

2. SYSTEM ARCHITECTURE

The Eco Friendly Checker extension is built using a modular client-server architecture that enables seamless interaction between a user's browser and a backend sustainability evaluation engine. The system is composed of three major components: the Chrome extension frontend, the content extraction module, and the Node.js-based backend server.

2.1 Chrome Extension Interface

The frontend of the system is a lightweight Chrome extension consisting of a popup UI and a background script. Upon user interaction, it triggers a workflow that extracts product information from the current web page and communicates with the backend server for eco-evaluation.

Key modules include:

- Popup Interface: Displays a minimal UI with a "Check Product" button.
- Popup Script (popup.js): Manages UI behavior, sends requests to the content script, and handles backend responses.

- Background Script (background.js): Logs the extension installation and maintains extension-level lifecycle control.

2.2 Content Extraction Script

The **content script (content.js)** is injected into shopping websites and is responsible for scraping structured data such as:

- Product Name
- Description
- Price
- URL

It uses DOM selectors to extract this information and supports fallback logic to work on pages with different structures.

The extracted data is then made accessible via Chrome's messaging API when a request is sent from the popup.

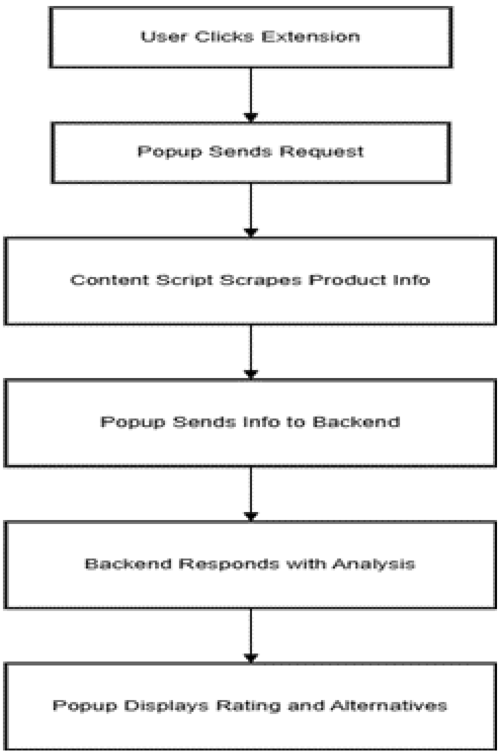
2.3 Backend Server (Node.js)

The backend is built using **Express.js**, running locally on http://localhost:3000. When the extension sends product data, the backend processes the request and returns:

- A dummy **sustainability rating** (e.g., 8.4/10)
- A short **AI-generated analysis** text
- A set of **eco-friendly alternative product links**

The backend uses CORS and JSON middleware to handle requests effectively and supports future integration with AI/ML models or sustainability databases.

2.4 Data Flow Diagram



3. Working & User Interface

The Eco Friendly Checker Chrome extension provides a smooth and intuitive user experience for evaluating the environmental impact of products while browsing e-commerce websites. The workflow is designed for minimal user interaction, requiring only a single click to initiate the analysis process.

3.1 Working Procedure

The extension operates through a coordinated interaction between its popup interface, content script, and backend server. The sequence of actions is as follows:

1. **User Navigation**: The user visits a product page on an e-commerce site (e.g., Amazon).
2. **Extension Trigger**: The user clicks the Eco Friendly Checker extension icon in the browser toolbar.
3. **UI Display**: A popup appears with a button labeled **"Check Product"**.
4. **Content Extraction**: On button click, the popup sends a message to the content script which extracts product data such as name, description, and URL from the current page.
5. **Backend Communication**: The extracted data is sent to a backend API (/rate) running on a local Express.js server.
6. **Eco Evaluation**: The backend responds with a dummy sustainability rating, an AI-generated analysis summary, and links to eco-friendly alternatives.
7. **Result Display**: The popup UI dynamically updates to display the rating, analysis, and alternative suggestions.

This approach ensures real-time feedback with no page reloads or manual data input required by the user.

3.2 User Interface Design

The UI of the extension focuses on simplicity and clarity. It consists of:

- **Initial View**: A clean popup with the title "Eco Friendly Checker", a brief subtitle encouraging sustainable shopping, and a green "Check Product" button.
- **Result View**: After processing, the popup expands to show:
  - A **Sustainability Rating** (e.g., "8.4/10")
  - A short **analysis** based on the product content
  - Two hyperlinked **eco-friendly alternatives**

The design uses a green color scheme to reinforce the environmental theme and a minimal layout to maintain focus on key information.

3.3 Visual Demonstration

Two key screenshots demonstrate the extension's function:

- **Before Analysis**: Displays the popup with the "Check Product" button.
- **After Analysis**: Shows the sustainability rating and links to alternative products after backend processing.

These screenshots validate the smooth integration and user-friendly experience offered by the extension during live product browsing.

Fig. 1 - Initial view of the extension popup showing the "Check Product" button.

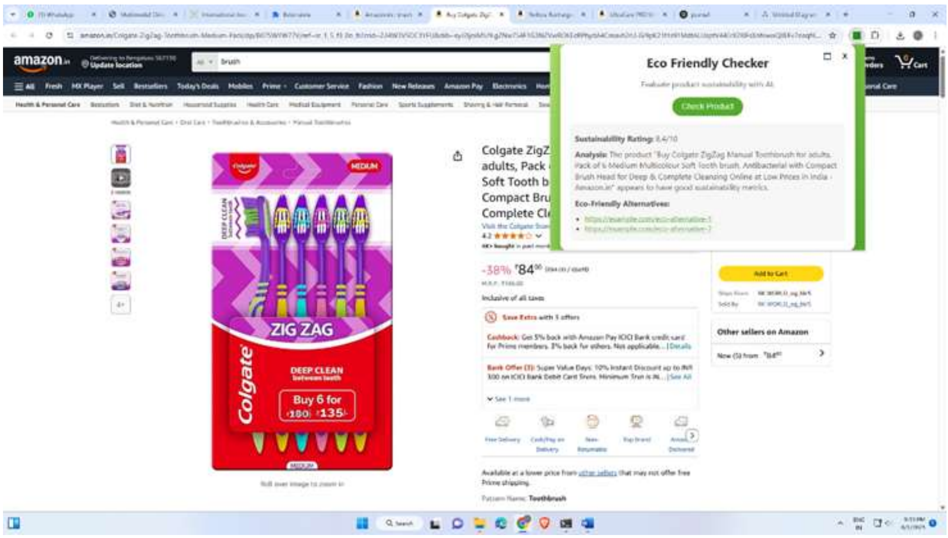


Fig. 2 - Popup interface showing the sustainability rating and eco-friendly alternatives.

4. Technologies Used

The development of the Eco Friendly Checker Chrome extension leverages a combination of **frontend web technologies**, **browser APIs**, and a **lightweight backend server**. The following technologies were utilized across the system's components:

4.1 Frontend Technologies

- **HTML5**: Used to structure the popup interface of the extension, ensuring semantic clarity and responsive layout.
- **CSS3**: Applied for styling the popup UI, with a focus on a clean, minimal, and eco-themed design.
- **JavaScript (ES6+)**: Serves as the core scripting language to implement interaction logic in both the popup and content scripts.
- **Chrome Extension APIs**: Specifically, the messaging API is used for communication between the popup, background script, and content script. The extension also uses manifest.json (Manifest v3) for configuration and permissions.

4.2 Backend Technologies

- **Node.js**: A runtime environment used to build the backend server due to its asynchronous, event-driven nature, which is well-suited for handling multiple API requests.
- **Express.js**: A minimal web framework for Node.js, used to define and manage the /rate endpoint that processes product data and returns sustainability results.
- **CORS Middleware**: Used to enable cross-origin communication between the local frontend (Chrome extension) and backend server.

4.3 Development Tools

- **Visual Studio Code (VS Code)**: The primary code editor used for developing and debugging both frontend and backend code.
- **Google Chrome**: Used as the testbed browser for loading and interacting with the extension.

5. Results and Future Work

5.1 Current Results

The Eco Friendly Checker Chrome extension was successfully developed, tested, and deployed in a local environment. It functions as intended across major e-commerce websites such as Amazon and Flipkart, where it can:

- Extract relevant product information (name, description, and URL) from dynamically generated pages.
- Communicate with a local backend server using a structured API request.
- Display eco-friendliness ratings and suggest environmentally conscious alternatives through a user-friendly popup interface.

The backend server responds with dummy but realistic sustainability data, and the extension successfully updates the UI to reflect this information in real time. The entire system demonstrates a complete, responsive, and modular product analysis pipeline from browser to backend.

Screenshots were captured to validate the working of the system and demonstrate its usability.

5.2 Limitations

- The eco-rating logic is currently **dummy** and does not yet use a real sustainability scoring algorithm.
- The extension works best on product pages with identifiable structure; some websites with non-standard layouts may not yield accurate results.
- Real-time data sources or machine learning models have not yet been integrated for deeper environmental analysis.

5.3 Future Enhancements

To enhance the system and extend its practical applicability, the following improvements are proposed:

- **Integration with sustainability databases** such as EcoLabel Index or product life-cycle APIs to provide authentic ratings.
- **Machine Learning model** to classify product sustainability based on features such as material composition, manufacturer credibility, and user reviews.
- **Browser storage** to track past evaluations and allow historical comparisons.
- **User feedback mechanism** to continuously refine and personalize product evaluations.
- **Deployment on a live web server** to enable real-time, online usage without requiring local backend setup.

All authors are required to complete the Procedia exclusive license transfer agreement before the article can be published, which they can do online. This transfer agreement enables Elsevier to protect the copyrighted material for the authors, but does not relinquish the authors' proprietary rights. The copyright transfer covers the exclusive rights to reproduce and distribute the article, including reprints, photographic reproductions, microfilm or any other reproductions of similar nature and translations. Authors are responsible for obtaining from the copyright holder, the permission to reproduce any figures for which copyright exists.

6. Acknowledgements

The authors would like to express their sincere gratitude to the Department of Artificial Intelligence and Machine Learning at Jyothy Institute of Technology, Bengaluru, for providing the necessary infrastructure and support to carry out this project. We extend special thanks to our mentor, **Ms.**

Ramya B N, for her continuous guidance, insightful suggestions, and encouragement throughout the development of the "Eco Friendly Checker" Chrome extension.

We also thank our peers and testers who contributed valuable feedback during the extension’s evaluation phase, which helped us enhance its usability and functionality.

6. References

Koo, C., Yu, C. P., & Lee, H. (2020). ***Developing a green shopping decision support system for consumers*****.** ***Journal of Cleaner Production, 258*****, 120726.** **https://doi.org/10.1016/j.jclepro.2020.120726**

Lee, S. M., & Trimi, S. (2018). ***Innovation for creating a smart future*****.** ***Journal of Innovation & Knowledge, 3*****(1), 1–8.** **https://doi.org/10.1016/j.jik.2016.11.001**

Cervantes, H., & Kazman, R. (2016). ***Designing Software Architectures: A Practical Approach*****. Addison-Wesley.**

Niemelä-Nyrhinen, J., & Uusitalo, O. (2013). ***Identifying green consumers – A segmentation study*****.** ***Journal of Consumer Policy, 36*****, 229–245.** **https://doi.org/10.1007/s10603-013-9224-1**

Lutfi, A., Alrahhal, M. M., & Alshira’h, A. F. (2022). ***Green IT adoption and sustainability: A study of technology adoption models in SMEs*****.** ***Journal of Cleaner Production, 368*****, 133142.** **https://doi.org/10.1016/j.jclepro.2022.133142**

Node.js Foundation. (2023). ***Node.js Documentation*****. Retrieved from https://nodejs.org/en/docs**

EcoLabel Index. (2023). ***Global directory of ecolabels*****. Retrieved from http://www.ecolabelindex.com/**