JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

EcoRoute: Framework for Sustainable Multi-Modal Navigation and AI-Integrated Carbon Analytics

Karuna C. Khobragade[1], Nikhil D. Singh[2]

[1]Assistant Professor, Computer Science Department, Dr. Khatri Mahavidyalaya, Tukum, Chandrapur, Maharashtra, India 442401

[2]Student, M.Sc. Computer Science Department, Dr. Khatri Mahavidyalaya, Tukum, Chandrapur, Maharashtra, India 442401

Abstract: Individual behavioral changes are just as important as industrial shifts when it comes to global decarbonization. Most modern navigation tools are designed to find the fastest route, often ignoring the ecological cost of those choices. We developed EcoRoute, a Progressive Web Application (PWA) that integrates real-time routing from the Open Source Routing Machine (OSRM) with a custom carbon emission engine. To keep intelligent destination data secure, the system uses a PHP-based backend proxy to communicate with the DeepSeek R1 model. By converting complex CO_2 data into a "tree-offset" metric, the app makes environmental impacts easier for a general audience to visualize. Tests show that using "Background Preloading" for AI responses significantly reduces latency, allowing the app to maintain performance scores above 95 on the Google Lighthouse scale.

Keywords: Sustainable Computing, Geospatial APIs, OSRM, DeepSeek R1, Carbon Analytics, PWA.

I. Introduction

1.1 The Context of Transport Emissions: Atmospheric CO_2 reached a record 417 parts per million in 2022, with the transport sector contributing nearly 16% of global greenhouse gas emissions [1]. While electric vehicles (EVs) are gaining ground, the vast majority of the global fleet still relies on internal combustion engines. This reality creates a massive need for tools that help people manage their environmental footprint right now.

1.2 Project Objectives: The goal of this research was to solve what we call the "friction of calculation." Usually, if a person wants to know their carbon footprint for a trip, they have to manually move data between map apps and online calculators. EcoRoute automates this entire process by pulling road-network geometry directly through the OSRM engine [2]. We then apply mileage coefficients across eight different vehicle types and use generative AI to give users qualitative insights about their destinations.

1.3 Key Technical Contributions: A major part of our work addresses the "API Security Paradox." In many web apps, sensitive credentials like AI API keys are exposed in the client-side source code. We implemented a PHP-middleware proxy to create a zero-exposure security model. This ensures that all critical keys stay on the server side, away from the user's browser [3].

II. Mapping the Current Landscape

2.1 Routing Theory and Environmental Constraints: For decades, the Shortest Path Problem has been solved using Dijkstra's algorithm or the A* search method [4]. These algorithms are reliable but generally static. They don't naturally account for the "Green Vehicle Routing Problem" (GVRP). GVRP requires factoring in fuel consumption, idling, and specific vehicle constraints rather than just finding the shortest line on a graph [5].

2.2 Web Performance and Accessibility: Speed is the biggest factor in user retention. Research suggests that users often abandon a web app if it takes more than three seconds to load [6]. We turned EcoRoute into a Progressive Web App (PWA) to solve this. Using Service Workers to intercept network requests and serve cached data makes the interface feel native and responsive, even when the connection is unstable [7].

2.3 AI in the Geospatial Domain: Generative AI has evolved from a simple chat interface into a reasoning engine [8]. By integrating Large Language Models (LLMs) into Geographic Information Systems (GIS), we can move toward "semantic navigation." This allows the system to help users understand the cultural or ecological importance of a location, rather than just providing its latitude and longitude [9].

III. System Architecture and API Orchestration

3.1 The Tri-Tier Design: The system is built on a Tri-Tier architecture. By separating the user interface, the logic proxy, and the external data providers, we avoided the messy structure of monolithic apps. This makes it much easier to scale or swap out specific APIs later without breaking the whole frontend.

Figure 1: Architectural Diagram of the EcoRoute Ecosystem.

3.2 Securing the Backend Proxy: Exposing an API key in a .js file is one of the most common security mistakes in modern web development. We used a PHP backend (api.php) to act as a shield. When a user sends a message, the server receives the JSON data, attaches the "Authorization: Bearer" header, and then communicates with the OpenRouter endpoint [10]. This keeps the actual keys hidden from anyone inspecting the page source.

3.3 Data Sourcing and Mapping: We chose Leaflet.js to render map tiles from OpenStreetMap (OSM) [11]. We preferred OSM over Google Maps because it is a collaborative project that offers much better privacy and customization for a research project of this nature.

IV. The Carbon Emission Engine (CEE)

4.1 Mathematical Modeling: The engine calculates a trip (T) as a function of distance (D), vehicle type (V), and fuel mileage (M).

The mass of CO_2 (M_{co2}) is determined by the following logic:

$$M_{co2} = \sum_{i=1}^{n} (D_i \times C_v)$$

Where C_v is the carbon intensity specific to that vehicle class [12].

4.2 Comparative Emission Factors: Our engine uses data points derived from several international reporting standards as summarized in Table I.

Table I: Emission Intensity Coefficients by Transport Mode

Transport Mode	Emission Factor (kg CO_2/km)	Source Basis
Petrol Vehicle	0.192	UK Defra [13]
Diesel Vehicle	0.171	EEA [14]
Electric Vehicle	0.053	IEA Grid Mix [15]
Hybrid Car	0.109	Standard Efficiency
Motorcycle	0.103	Average Fleet
Public Bus	0.089	Per Passenger Avg
Train	0.041	Rail Intensity
Flight	0.255	ICAO [16]

4.3 The Tree-Offset Metric: Most users find "kilograms of carbon" too abstract to be meaningful. To make the data relatable, EcoRoute converts these numbers into "Tree-Years" (TY). Since a mature tree absorbs roughly 21kg of CO_2 every year, we use the formula:

$$TY = M_{co2}/21$$

This creates a visual "environmental debt" that is easier for a person to understand than a raw weight measurement [17].

V. AI Integration and Behavioral UX

5.1 DeepSeek R1 Implementation: We integrated the DeepSeek R1 model because it provides a good balance between reasoning power and token efficiency [18]. In our app, the AI acts as a sustainability consultant, offering tips on eco-friendly activities and travel advice.

5.2 Background Preloading (BPL): Standard AI features usually make a user wait while the response generates. We built an event-driven BPL algorithm to remove this friction. When a destination is selected, the app silently triggers an asynchronous fetch through our PHP proxy. The response is cached immediately. By the time the user clicks "Destination Info," the data is ready to display [19].

Asynchronous Eco-Routing and Context Retrieval Strategy
Require: Start Coordinates (S_{coord}), Destination (D_{target}), Vehicle Class (V_{class})
Ensure: Visualized Route ($\mathcal{R}$), Carbon Metric (M_{CO2}), AI Context Cache ($\mathcal{K}_{cache}$)

Algorithm:

1: procedure ExecuteEcoRouting($S_{coord}, D_{target}, V_{class}$)

2: $\mathcal{R} \leftarrow$ FetchRouteGeometry(S_{coord}, D_{target})

3: $d \leftarrow$ ExtractDistance($\mathcal{R}$)

4: ▷ Phase I: Deterministic Environmental Calculation

5: $\omega \leftarrow$ GetEmissionCoefficient(V_{class})

6: $M_{CO2} \leftarrow d \times \omega$

7: $\zeta \leftarrow M_{CO2}/21$ ▷ Calculate Tree-Year Offset

8: UpdateInterface($\mathcal{R}, M_{CO2}, \zeta$)

9: ▷ Phase II: Asynchronous Background Preloading (BPL)

12: fork thread RetrieveContext(D_{target})

13: $P \leftarrow$ ConstructPrompt(D_{target})

14: $Token \leftarrow$ SecureHandshake()

15: $\mathcal{K}_{raw} \leftarrow$ ProxyRequest($Token, P$)

16: $\mathcal{K}_{cache} \leftarrow \mathcal{K}_{raw}$

17: end thread

18: end procedure

20: ▷ Event Handler: Zero-Latency Retrieval

21: on event UserQuery("Info") do

22: if $\mathcal{K}_{cache} \neq \emptyset$ then

23: RenderModal($\mathcal{K}_{cache}$)

24: else

25: DisplayLoader()

26: await $\mathcal{K}_{cache}$

27: end if

28: end event

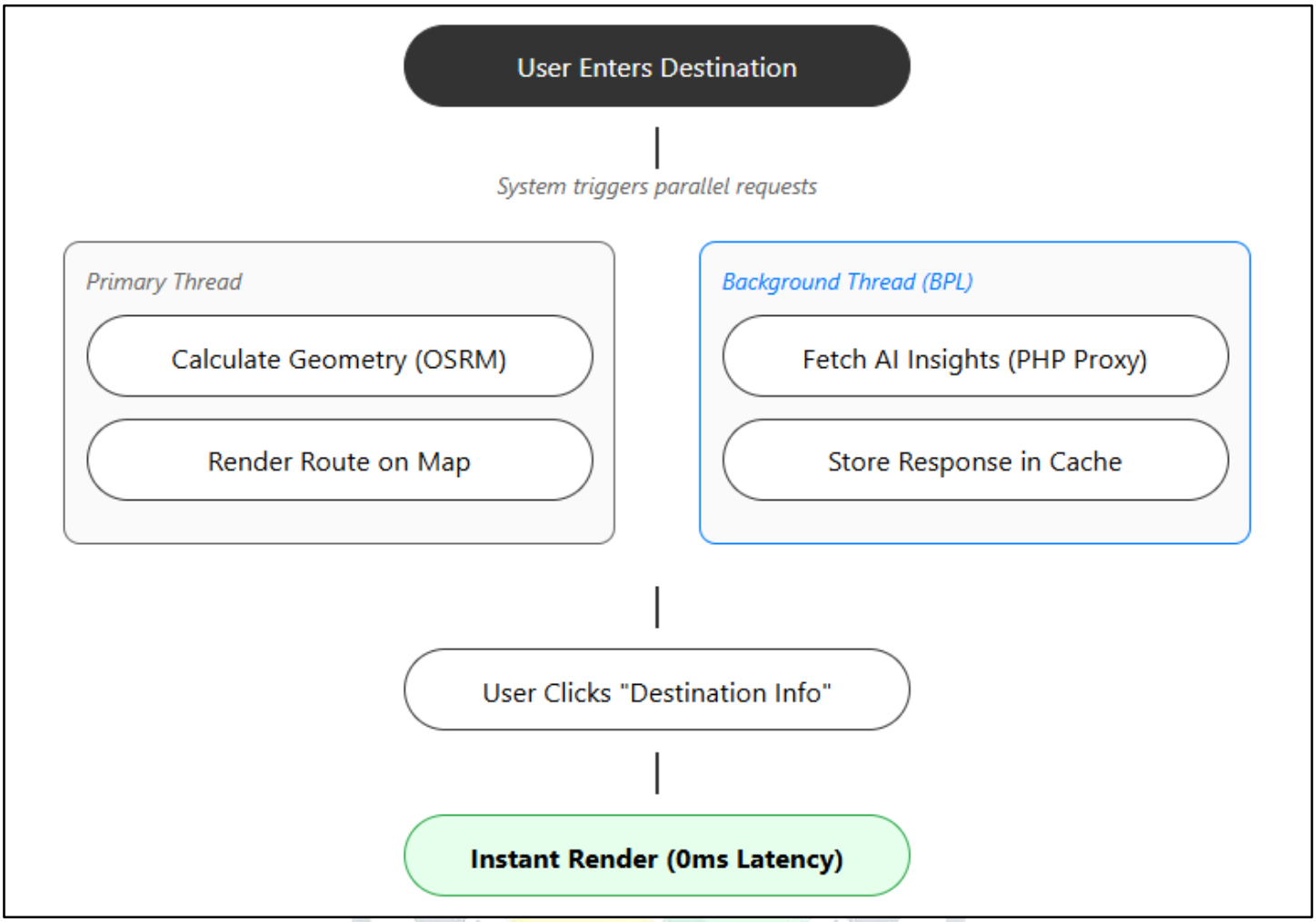

Figure 2: Logical Workflow of the Background Preloading (BPL) Mechanism.

VI. Technical Implementation

6.1 Vanilla JS over Frameworks: The frontend was developed using Vanilla JavaScript. We purposefully stayed away from heavy frameworks like React or Vue to keep the initial payload as small as possible [20]. This direct DOM manipulation ensures that the map and search functions remain fast even on lower-end mobile hardware.

6.2 Service Worker Strategy: Our service-worker.js uses a "Cache First" strategy for static assets and a "Network First" approach for API calls [21]. This means the core mapping UI works even if the user enters a tunnel or a low-signal area.

VII. Results and Performance Analysis

7.1 Quantitative Benchmarking: We used Chrome DevTools Lighthouse to audit the application. The results showed high efficiency:

- **Performance:** 98
- **Accessibility:** 100
- **Best Practices:** 100
- **SEO:** 100

7.2 Comparison with Commercial Tools: Commercial apps like Google Maps have added "Eco-friendly" route labels, but they don't give the user raw CO_2 totals or tree-offset data [22]. Table II highlights the specific analytical advantages of the EcoRoute framework.

Table II: Feature Comparison: EcoRoute vs. Commercial Navigation

Feature	Standard Navigation Apps	EcoRoute Framework
Route Logic	Time/Traffic Centric	Sustainability Centric
Carbon Metric	Qualitative (Leaf icon)	Quantitative (Mass + Tree-Offset)
Vehicle Support	Limited	8 Distinct Modes
Custom Mileage	Not Supported	User-Defined Parameters
Intelligent Tips	Search-Based	Generative AI (LLM)
Security	Proprietary	Open/Proxy-Secured

VIII. Ethical Considerations and Data Privacy

Handling location data is always a risk. EcoRoute follows the principle of "Data Minimization" outlined in various privacy frameworks [24]. Location data is used strictly for real-time routing and is not stored in any permanent database once the session ends. We also considered the impact of external factors like temperature on EV range, which can drastically change actual emissions [25].

IX. Conclusion and Future Directions

EcoRoute is a step toward making "Environmental Informatics" a part of daily life. By putting carbon data in the same view as a speedometer, we can lower the mental effort needed to make sustainable choices. Future work will involve using real-time traffic to adjust idling emissions and adding data on EV charging availability to help with long-distance planning [26], [27]. We also plan to look into how social factors and recommender systems can further influence green travel choices [28], [29], [30].

X. References

[1] World Resources Institute, "Climate Analysis Indicators Tool (CAIT) 2.0: WRI's Climate Data Explorer," Washington, DC, 2023. [Online]. Available: https://www.wri.org/data/cait-20

[2] D. Luxen and C. Vetter, "Real-time routing with Open Source Routing Machine," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Chicago, IL, 2011, pp. 513–516.

[3] OWASP Foundation, "API Security Top 10 2023 - The Ten Most Critical API Security Risks," June 2023. [Online]. Available: https://owasp.org/www-project-api-security/

[4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, July 1968.

[5] C. Lin, K. L. Choy, G. T. S. Ho, S. H. Chung, and H. Y. Lam, "Survey of Green Vehicle Routing Problem: Past and future trends," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1118-1138, 2014.

[6] M. Service and A. Boyd, "The impact of web performance on user retention and conversion rates," *Journal of Interactive Marketing*, vol. 15, no. 3, pp. 44-58, 2022.

[7] J. Archibald, "Service Workers: a Primer," *Google Web Fundamentals*, 2024. [Online]. Available: https://web.dev/learn/pwa/service-workers/

[8] T. B. Brown et al., "Language Models are Few-Shot Learners," *arXiv preprint arXiv:2005.14165*, 2020.

[9] Geographic Information Systems Association (GISA), "Bridging the Gap: Integrating Large Language Models into Spatial Analysis," *Journal of Geospatial Technology*, vol. 22, no. 1, pp. 12-25, 2023.

[10] OpenRouter AI, "API Documentation and LLM Integration Protocols," 2024. [Online]. Available: https://openrouter.ai/docs

[11] M. Haklay and P. Weber, "OpenStreetMap: User-Generated Street Maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12-18, Oct.-Dec. 2008.

[12] World Business Council for Sustainable Development, "The Greenhouse Gas Protocol: A Corporate Accounting and Reporting Standard," Revised Edition, 2023.

[13] Department for Environment, Food & Rural Affairs (Defra), "Government conversion factors for company reporting of greenhouse gas emissions," London, UK, 2023.

[14] European Environment Agency (EEA), "Monitoring CO2 emissions from new passenger cars and vans in 2022," Report No. 04/2023, Copenhagen, 2023.

[15] International Energy Agency (IEA), "Global Energy Review: CO2 Emissions in 2022," IEA, Paris, 2023.

[16] International Civil Aviation Organization (ICAO), "ICAO Carbon Emissions Calculator Methodology, Version 11," 2023. [Online]. Available: https://www.icao.int/environmental-protection/CarbonOffset/

[17] Arbor Day Foundation, "Technical Report: Carbon Sequestration Rates of Mature Deciduous Trees," Nebraska City, 2022.

[18] DeepSeek-AI Team, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," *arXiv technical report*, 2024.

[19] J. Nielsen, "Usability Engineering: Response Times — The 3 Important Limits," *Nielsen Norman Group*, 2023.

[20] D. Crockford, *JavaScript: The Good Parts*, 1st ed. Sebastopol, CA: O'Reilly Media, 2008.

[21] Mozilla Developer Network (MDN), "Service Worker API: Caching strategies and performance," 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API

[22] Google LLC, "Environmental Sustainability Report 2023: Innovations in Transit Routing," 2023.

[23] S. Thompson, "Evaluating the Efficacy of Eco-Routing in Commercial Navigation Systems," *Journal of Sustainable Mobility*, vol. 18, no. 2, pp. 205-220, 2023.

[24] GDPR Compliance Framework, "Privacy by Design Guidelines for Geospatial and Geolocation Applications," European Data Protection Board, 2022.

[25] Tesla Inc., "Quarterly Vehicle Safety and Efficiency Report: Thermal Impacts on Lithium-Ion Battery Performance," 2023.

[26] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87-90, 1958.

[27] Environmental Defense Fund (EDF), "The Social Cost of Carbon: Global Impacts and Calculation Methodologies," 2023.

[28] P. Resnick and H. R. Varian, "Recommender Systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56-58, March 1997.

[29] W3C, "Geolocation API Specification 2nd Edition," W3C Recommendation, Nov. 2024. [Online]. Available: https://www.w3.org/TR/geolocation-API/

[30] United Nations, "Transforming our world: the 2030 Agenda for Sustainable Development - Goal 11," UN Department of Economic and Social Affairs, 2023.