International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 2, Mar-Apr 2026

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Modern Solutions For Global Climate Action

C. Asmitha Reddy, S. Pranav Reddy, Md. Imthiyaz, K. Mahesh

CSE(AIML), Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, INDIA

asmithareddy9959@gmail.com

CSE(AIML), Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, INDIA

solipurampranav@gmail.com

CSE(AIML), Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, INDIA

mohammedimthiyaz308@gmail.com

CSE(AIML), Sreyas Institute of Engineering and Technology, Hyderabad, Telangana, INDIA

kethavathmahesh48@gmail.com

Abstract:

Climate change has become a major global issue due to the continuous rise in carbon emissions, particularly from transportation activities. This project proposes the development of a web application called Eco-Route that helps users identify environmentally friendly travel routes instead of simply choosing the fastest path. The application is designed to estimate and compare carbon emissions generated during travel and provide route suggestions that reduce environmental impact. By using modern web technologies and map-based services, the system allows users to enter a starting point and destination and then analyses possible routes.The web application integrates mapping tools such as Google Maps API or Leaflet to display routes and calculate distances. Based on a fuel consumption model and transportation mode, the system estimates the amount of carbon dioxide (CO_2) emitted for each route. Users can compare different travel options such as electric vehicles, gasoline vehicles, or bicycles and choose the route that produces the least carbon emissions. This approach helps users understand how transportation choices affect the environment.The main objective of this web application is to encourage sustainable transportation by providing real-time information about carbon emissions and eco-friendly routing options. By combining routing algorithms, emission estimation models, and interactive web technologies, the proposed system demonstrates how web-based platforms can support environmental awareness and climate action. This project contributes to global sustainability efforts and supports Sustainable Development Goal 13 by promoting digital solutions that help reduce greenhouse gas emissions.

Keywords: Carbon Emissions, Eco-Routing, Sustainable Transportation, Web Application, Climate Change, Green Routing Algorithms, SDG 13.

1.INTRODUCTION:

The role of transportation in our daily lives is very significant. At the same time, transportation is one of the biggest factors contributing to environmental pollution and fuel consumption. With an increase in the number

of vehicles and urban population, problems such as traffic congestion, carbon footprint, and fuel consumption are becoming more and more severe. Hence, it has become a need of the hour to come up with a solution that not only helps individuals choose a route to reach a destination faster but also in a way that is environmentally safe.

The Eco-Route project aims to address this issue by providing an intelligent route planning system. This system will be able to offer a route not only fast but also safe for the environment. This system will be able to assist individuals in making better choices by using technology such as web development and mapping services.

The objective of this project is to develop a simple web application that will allow a user to input a starting point and a destination, then it will show several options of routes on a map, and finally choose a route that will minimize fuel consumption and pollution. This Eco-Route system not only helps individuals in becoming more efficient but also helps in protecting the environment and stopping global climate change.

This project uses latest technologies such as data analysis, GPS navigation, and intelligent algorithms to provide the users with better and environmentally friendly choices for their routes while travelling. The proposed solution is intended so that individuals, organizations, and planners can make environmentally friendly decisions while travelling. This system helps with reducing fuel consumption as well as lowering greenhouse gas emissions.

The proposed solution encourages people to make decisions that are eco-friendly while travelling. It also helps in promoting green technologies and highlights their importance in the modern world. It helps contribute to a cleaner environment while also making it for convenient for users. This system helps with routing while also contributing to the environmental factor.

This system gives you more greener options that are easy for your convenience while also considering multiple factors such as traffic, fuel consumption, greenhouse gas emissions etc. In most routing systems, only a few factors such as time and distance. This system considers the factors that affect the environment. That is what makes it much more eco-friendly and the fact that it helps reduce environmental degradation makes it even more usable.

This system overall helps in solving both the routing problems faced in our day to day lives and also helps in saving the environment as well. It offers multiple options so that it is of convenience for the user. It makes even travelling eco-friendly and helps contribute to a greener world.

2.PROPOSED METHODOLOGY

This Eco Route project focuses on developing a routing system that helps reduce environmental degradation. Unlike normal routing systems which only consider factors such as distance and time, this system considers factors like fuel consumption, traffic, greenhouse gas emissions, etc.

2.1 System overview

Eco Route is a simple, easy-to-use web application that runs directly in your browser. It follows a clean split-screen layout to make everything feel intuitive. On the left, you'll see an interactive map powered by Leaflet.js using OpenStreetMap, while the right side contains all the controls—like the input form, route options, comparison table, and navigation buttons. This layout was intentionally designed so users can view the map and interact with route details at the same time, without constantly switching between screens.

The idea behind Eco Route is to make route planning smarter and more environmentally friendly. A user just needs to enter a starting point and destination, choose their vehicle type, and click "Search." Within seconds, the system pulls real route data using the OSRM API, calculates the carbon emissions for each route based on the selected vehicle, and highlights the most eco-friendly option as the "Eco Route." It can also generate helpful AI-based tips using Gemini to encourage greener travel choices. All available routes are displayed on the map with different colors, making them easy to compare. From there, users can pick a route, save it for later, open it in Google Maps for navigation, or even use the app's built-in navigation feature.

2.2 System architecture

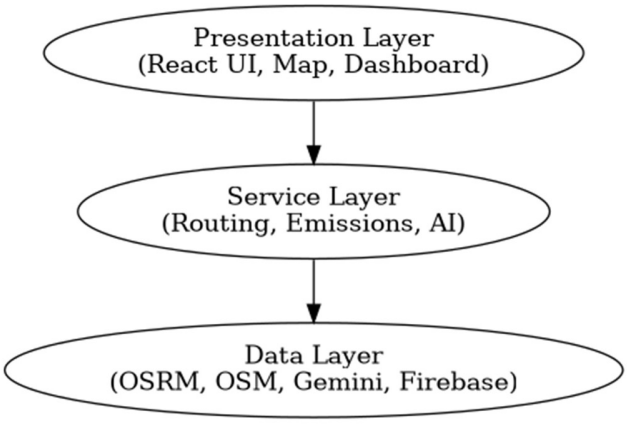
The Eco Route system is built using a simple three-layer (tri-tier) architecture that keeps everything well-structured and easy to manage.

At the top is the presentation layer, which is the React-based frontend that users interact with. This is where users enter their locations, view maps, compare routes, and navigate through the app.

Behind that is the logic or service layer, which acts as the brain of the application. It handles tasks like fetching routes from APIs, calculating carbon emissions for different routes, and integrating AI features to provide eco-friendly suggestions.

Finally, there's the data layer, which connects the system to external services and storage. This includes tools like OSRM for routing, OpenStreetMap/Nominatim for location data, Gemini AI for intelligent tips, Firebase for cloud storage, and localStorage for saving user data on the browser.

Together, these layers work smoothly to process user input, retrieve and analyze data, and display meaningful results. The diagram below illustrates how information flows between these layers.



2.3 Module 1-User interface and Navigation

Eco Route's interface is built to feel smooth, modern, and easy to use. It uses React 19 with TypeScript for building the app, along with Vite to keep everything fast and responsive. The design is handles with Tailwind CSS v4, which helps maintain a clean and consistent look throughout. Moving between pages is simple and seamless thanks to react-router-dom v7, which connects all the main sections like the landing page, map view, dashboard, and other informational pages.

The Landing Page is designed to grab attention right away. It features a full-screen background image and subtle animations that make the experience feel lively. Using GSAP **(**GreenSock Animation Platform**)**, elements gently come to life—the logo floats slightly, the title fades in smoothly, and the main buttons appear with a soft bounce. These small touches make the app feel more polished and engaging for first-time users.

When users go to the Map Page, they enter the main working area of Eco Route. The layout adjusts based on the device being used. On a desktop, the screen is split into two parts: the map on one side and the control panel on the other, so users can easily interact with both at the same time. On mobile, the control panel becomes a bottom sheet that can be dragged up or tapped to expand. This is powered by Framer Motion, making the interaction feel smooth and natural.

Overall, Eco Route is designed to provide a seamless experience across devices, so users can easily plan and explore eco-friendly routes whether they're on a computer or a phone.

2.4 Module 2-Input form and Geocoding

The InputForm is the starting point of the whole experience. It's designed to be simple and user-friendly, asking for just three things: where you're starting from, where you want to go, and what type of vehicle you're using. The vehicle options—Petrol Car, Diesel Car, Bike, and Electric Vehicle—help the system later estimate how eco-friendly each route will be.

When the user clicks "Search," the app takes those place names and converts them into exact map coordinates (latitude and longitude). This is done using OpenStreetMap's Nominatim service, which basically understands location names and turns them into precise points on the map. The system also follows proper guidelines by including a User-Agent with each request.

For an even easier experience, users don't always have to type. They can simply click on a location directly on the map, and the app will automatically figure out the address of that spot and fill it into the destination field. This makes the process feel more natural and especially convenient for mobile users who prefer tapping over typing.

2.5 Module 3-OSRM routing machine

Once the app knows the exact coordinates of the start and end points, it moves on to the next step—finding the best possible routes. To do this, it sends a request to the OSRM routing service, which calculates real-world driving routes between the two locations.

Instead of giving just one route, the system asks for multiple options so users can compare and choose what works best for them. It also requests detailed information, including the full shape of each route and step-by-step directions for navigation.

The response comes back with a list of routes, and each one includes details like how far it is, how long it will take, and the exact path it follows on the map. It also includes turn-by-turn instructions, which can be used if the user wants guidance along the way.

The app then takes this information and organizes it into a format it can easily work with. As part of this process, it adjusts the coordinate format so it matches what the map (Leaflet.js) expects, making sure all the routes appear correctly and smoothly on the screen.

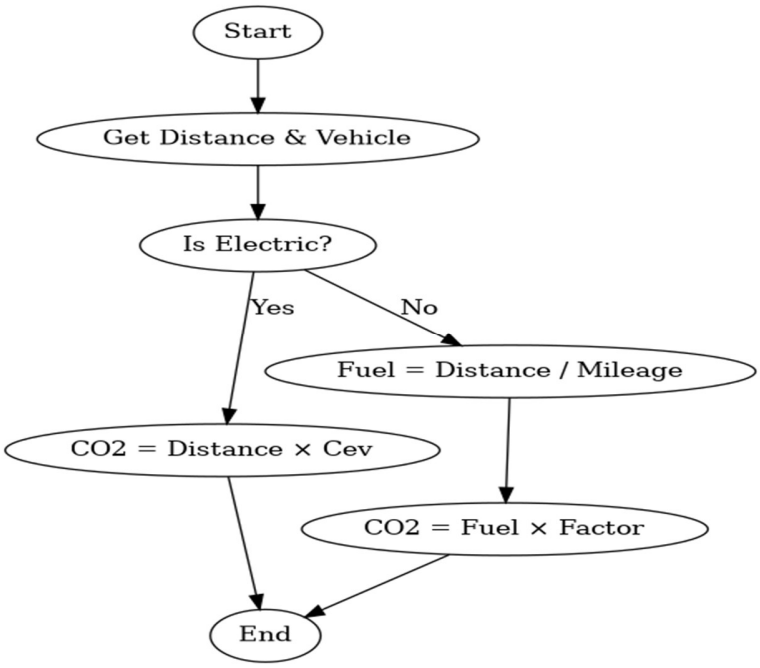
2.6 Module 4-Carbon Emission Engine(CEE)

The Carbon Emission Engine is a separate utility module in the project that focuses entirely on calculating how much CO_2 is produced for each route. It's implemented in a file called emissionCalculator.ts, making it easy to manage and reuse across the application.

At its core, this module stores predefined emission values for different types of vehicles. It then provides a simple function—calculateEmissions(distanceKm, vehicleType)—which takes the distance of a route and the selected vehicle type, and returns an estimate of the carbon emissions.

The way emissions are calculated depends on the type of vehicle:

- For petrol cars, diesel cars, and bikes, the system first estimates how much fuel would be used for the given distance based on the vehicle's average mileage (km per litre). It then calculates the CO_2 produced by multiplying that fuel usage with a standard emission factor (how much CO_2 is released per litre of fuel burned).
- $MCO2=Mkm/LDkm \times Cv$ $(kg\ CO2/L)$
- For electric vehicles, the calculation is more straightforward. Instead of fuel, it considers the carbon impact of electricity. The system simply multiplies the distance by a fixed value that represents how much CO_2 is generated per kilometer based on the electricity grid.
- $MCO2=Dkm \times Cev$ $(kg\ CO2/km)$



This approach keeps the calculations realistic while still being efficient, allowing the app to quickly compare routes and highlight the most eco-friendly option.

Once the app calculates the emissions for all the available routes, it compares them to find the most environmentally friendly option. The route with the lowest carbon emissions is marked as the Eco Route.

To make it easy to spot, this route is highlighted in a bright emerald green on the map and labeled with a special “Eco” badge in the interface. This way, users can quickly identify the best choice without having to analyze all the details themselves.

The app also shows how much CO_2 can be saved by choosing this eco-friendly route instead of the least efficient one. This difference is displayed as a small floating overlay on the map, helping users clearly see the environmental impact of their decision.

2.7 Module 7-Gemini AI integration

Gemini AI is used in the app in two helpful ways to make the experience more engaging and informative.

First, it’s used to give simple names to routes. After the app gets multiple route options, it asks Gemini to come up with short, easy-to-understand names like “Scenic Path” or “Fastest Route.” This makes it much easier for users to quickly recognize and compare routes instead of relying only on numbers like distance or time. The app asks for the response in a structured format so it can display the names smoothly without errors.

Second, Gemini is used to provide eco-friendly driving tips. Once the app figures out which route is the most environmentally friendly, it asks Gemini to suggest a few practical tips to reduce emissions during the trip.

These tips are based on things like the distance, locations, and type of vehicle, and they appear in a small card on the map along with the CO_2 savings.

Both of these features run quietly in the background. If there's any issue—like a network problem or API limit—the app doesn't break. Instead, it simply uses default names and skips the tips, so users can still continue using the app without any interruption.

2.8 Module 6-AI assistant "Vihari"

The ChatVihari feature is designed to feel like a helpful companion inside the app. You'll see it as a small chat bubble in the bottom-right corner of the map page, always there if you need assistance. When you click on it, a neat little chat window opens up where you can start asking questions.

The assistant is named Vihari, a name inspired by Sanskrit that reflects the idea of travel and movement. It's powered by AI and acts like a friendly guide for Eco Route. You can ask it anything—from simple questions like "Which vehicle is more eco-friendly?" to more detailed ones like "How does the app calculate CO_2 emissions?"

Vihari gives clear, useful answers in a conversational way, making the experience feel more interactive and personal. Your chat stays active during your session, so you can scroll back and check previous messages anytime. And if you're done, you can simply close the chat—it's always there when you need it again.

2.9 Module 7-Interactive Map Component

The MapComponent is the core of the application and where most of the action happens. It brings together the map, routes, and interactive features into one smooth experience using react-leaflet, which allows the map to work naturally within the React setup.

One of the first things users notice is the ability to switch between different map views. Whether it's a standard street map, a dark theme, satellite view, or terrain mode, users can pick what suits them best. This makes the app more flexible—for example, satellite view can help with real-world navigation, while dark mode is easier to use at night.

Routes are displayed in a way that makes them easy to see and compare. Each route is drawn using a layered style—a white outline underneath and a colored line on top—so it stands out clearly on any map background. The selected route is highlighted in green if it's the eco-friendly option or blue for other choices, while the remaining routes are faded in gray to keep the focus clear.

For electric vehicle users, the app adds extra convenience by showing nearby charging stations directly on the map. These appear as markers with a lightning icon. When tapped, they show basic details and even give an option to open directions in Google Maps, making it easier to plan charging stops.

There's also a live speed display on the map, which shows how fast the user is moving in real time. This uses the device's location data and smooths out sudden changes so the speed feels stable and accurate. Along with this, a small GPS accuracy indicator shows how reliable the location data is at any moment.

To give users more control, the app includes a route filter panel. This allows them to sort or filter routes based on things like travel time, distance, or emission levels. It helps users quickly find a route that matches both their practical needs and their environmental goals.

Even though there's a lot happening behind the scenes, everything is designed to feel simple, clear, and easy to use.

2.10 Module 8-Dashboard

The Dashboard page (/dashboard) works like a personal tracker for your travel and its environmental impact. It pulls in all the routes you've saved from your browser's local storage and displays them in a simple, easy-to-understand way.

The page mainly has two sections:

The Recent Journeys section shows a list of all your saved trips. For each trip, you can quickly see where you went, how far it was, and how much CO_2 was produced. If you hover over any trip, you'll get extra options like viewing the route again or starting navigation, making it easy to revisit past journeys.

The second part is the Environmental Impact section. This gives you a bigger picture by adding up the total CO_2 from all your trips. To make it more meaningful, the app also converts this number into a tree equivalent—showing how many trees would be needed to offset that carbon. This helps you better understand your impact in a more real-world way.

- Trees Equivalent=20 *kg*/treeTotal Saved CO2 (*kg*)

This tree-offset metric makes the abstract concept of carbon savings tangible and emotionally resonant for everyday users.

2.11 Module 9-Navigation Overlay

The NavigationOverlay component comes into action when the user clicks on "Start Navigation" for a selected route. It turns the map into a simple, easy-to-follow navigation screen.

Once activated, it displays clear turn-by-turn directions, along with useful details like how far the next turn is, the total remaining distance, and the estimated time to reach the destination. Everything is shown in a way that's easy to glance at while moving.

At the same time, a car icon on the map moves in real time, updating based on the user's GPS location. This gives a live navigation feel, similar to popular map apps.

If the user wants to stop at any point, they can easily exit navigation and go back to the route selection screen without any hassle.

2.12 System working-End to End Flow

Here's a more natural, step-by-step flow of how a user interacts with Eco Route:

1. The user opens the Eco Route website. If it's their first time, they're greeted with a smooth, scenic animation that creates a welcoming first impression.
2. They arrive on the landing page and click on **"Start Journey"** to begin planning their route.
3. The app takes them to the map page, where it automatically tries to detect their current location using the device's GPS.
4. The user enters their starting point, destination, and selects their vehicle type in the input form, then clicks Search.
5. The app converts the entered locations into exact map coordinates using a geocoding service.
6. It then fetches multiple possible routes between those points, giving the user a few options to choose from.
7. Each route is analyzed to calculate how much CO_2 it would produce based on the distance and selected vehicle.
8. The app identifies the most eco-friendly route—the one with the lowest emissions—and highlights it.
9. At the same time, AI features step in to give each route a simple name and generate helpful eco-driving tips.
10. All the routes are displayed on the map with clear colors, and detailed route cards appear showing distance, time, emissions, and an eco badge where applicable.
11. The user can then choose what to do next—save the route, start in-app navigation, or open it in Google Maps.
12. Any saved routes are stored and later shown in the dashboard, where the user can track their total carbon impact and see it translated into something meaningful, like tree equivalents.

3.EXPERIMENTAL SETUP

3.1 Software stack

Eco Route is built using modern web technologies that make the application fast, smooth, and easy to work with.

The frontend is developed using React 19 with TypeScript 5.8, which helps keep the code structured, reusable, and reliable. For building and running the app, Vite 6.2 is used, allowing for very fast development and

efficient performance in production. The overall look and feel of the app is created using Tailwind CSS v4, which makes designing the interface quick and consistent.

For the map functionality, Eco Route uses Leaflet along with react-leaflet, making it easy to display and control interactive maps within the app. To handle locations and routes, it relies on open-source services—Nominatim for converting place names into coordinates and OSRM for generating route options. This keeps the system lightweight and free from expensive API dependencies.

The app also includes AI-powered features using Google Gemini, which helps generate route names and eco-friendly travel tips, making the experience smarter and more engaging.

Other tools add to the overall experience. Firebase is used for cloud support, while animations are handled using GSAP for smooth visual effects and Framer Motion for interactive transitions. Icons are provided by Lucide React, giving the interface a clean and modern appearance.

Together, these technologies help Eco Route deliver a fast, interactive, and user-friendly experience.

3.2 Backend and Server

The app runs on a simple Express.js server, which handles how everything is delivered to the user.

While developing, the server works together with Vite's development setup. This means any changes you make in the code show up instantly in the browser without needing to refresh or restart anything, making development much faster and smoother.

Once the app is ready to go live, it's built into a final set of files stored in a dist folder. The Express server then serves these files like a regular website, ensuring quick and reliable performance.

The server runs on port 3000 and also makes sure navigation works properly. Since Eco Route is a single-page app, it handles all routes internally—so even if someone directly opens a page like /map or /dashboard, the app still loads correctly without any errors.

3.3 Deployment

The application is hosted on Vercel, a cloud platform that's well-suited for modern web apps like Eco Route.

One of the biggest advantages of using Vercel is its global CDN (Content Delivery Network). This means the app's files are delivered from servers that are closest to the user's location, helping the website load faster no matter where it's accessed from.

Eco Route is live at: https://eco-route-navy.vercel.app, making it easy for users to access it anytime.

Sensitive information, like the GEMINI_API_KEY, is handled securely through Vercel's environment settings. This ensures that important keys are kept safe and are never exposed in the frontend code, maintaining both security and reliability.

3.4 Development Environment

The project was built on a regular laptop using Node.js v20 (LTS), so it doesn't need any special setup or powerful hardware.

Getting it running locally is quite simple. You just need to install Node.js, run npm install to download the required packages, and then start the app with npm run dev. After that, the app will launch and be ready to use.

To keep things secure, the GEMINI API key is stored in a .env.local file. This file is excluded from version control using .gitignore, so sensitive information isn't shared publicly.

The complete source code is available on GitHub, making it easy for others to view or contribute to the project: https://github.com/tejassxo/Eco-Route.

3.5 API Dependencies Summary

Service	Purpose	Type
OSRM Public API	Real-time multi-route road geometry	Free/Open Source
Nominatim (OSM)	Forward and reverse geocoding	Free/Open Source
Google Gemini AI	Route naming, eco tips, chat assistant	API Key Required
Firebase	Cloud backend operations	Free Tier
OpenStreetMap Tiles	Base map rendering	Free/Open Source
Vercel	Hosting and deployment	Free Tier

4.RESULTS

Eco Route has been fully developed, tested, and successfully deployed as a working web application. All the key features—like generating multiple routes, comparing CO_2 emissions, using Gemini AI, tracking live location, and managing a personal dashboard—are working smoothly as intended.

In this section, we'll walk through the main parts of the interface and take a closer look at how the system performs when it's actually used.

4.1 Landing page

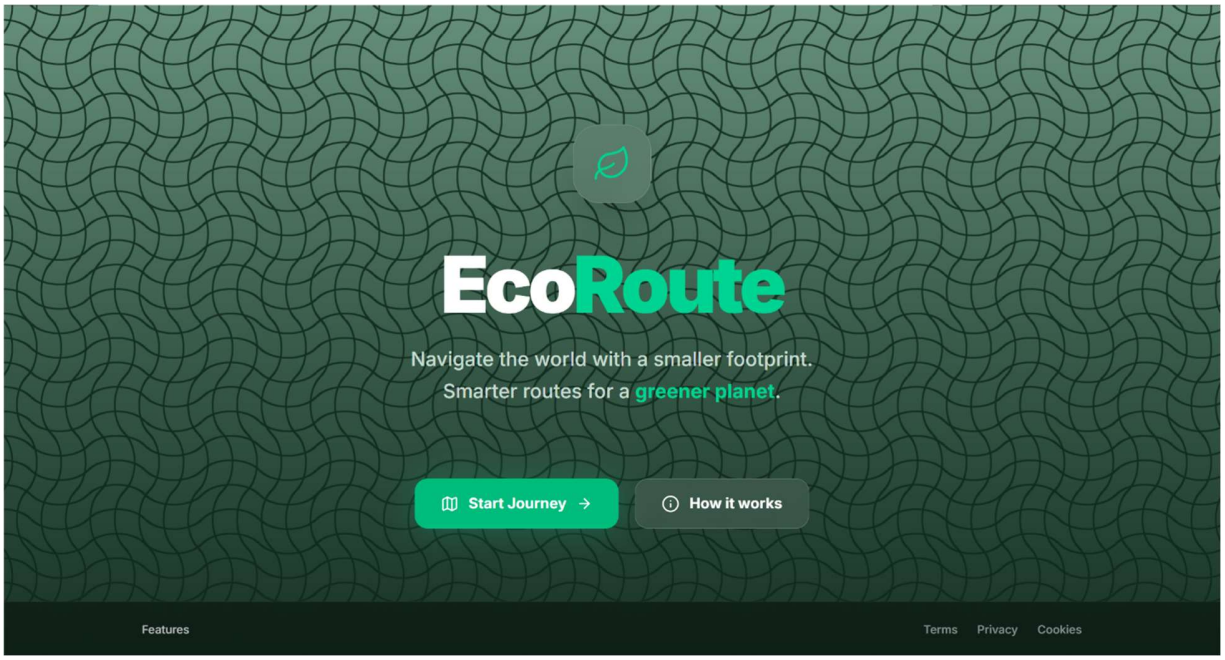
The landing page is the first impression users get of Eco Route, and it's designed to feel clean, calm, and inviting.

The entire screen is filled with a nature-themed background, giving it a fresh, eco-friendly vibe right from the start. In the center, there's a glowing leaf icon that gently moves up and down, almost like it's breathing, which adds a subtle sense of life to the page.

Right below it, the title **"**Eco Route**"** stands out in bold text, with "Route" highlighted in green to reflect the app's focus on sustainability. Under the title, a simple tagline encourages users with the idea of traveling smarter and reducing their environmental impact.

Below that, there are two clear options: a standout **"**Start Journey**"** button that draws attention, and a softer **"**How it works**"** button for users who want to explore more first. At the bottom, a small footer provides links to important pages like Terms, Privacy, and Cookies.

As the page loads, everything comes in smoothly with subtle animations. The layout gently scales into place, the leaf icon rotates slightly, the title appears step by step, and the buttons slide in with a soft bounce. It all happens quickly, creating a polished and engaging feel without slowing the user down.



4.2 Map page-Initial state

When the user clicks **"**Start Journey**,"** they are taken to the main map page, which is where most of the interaction happens.

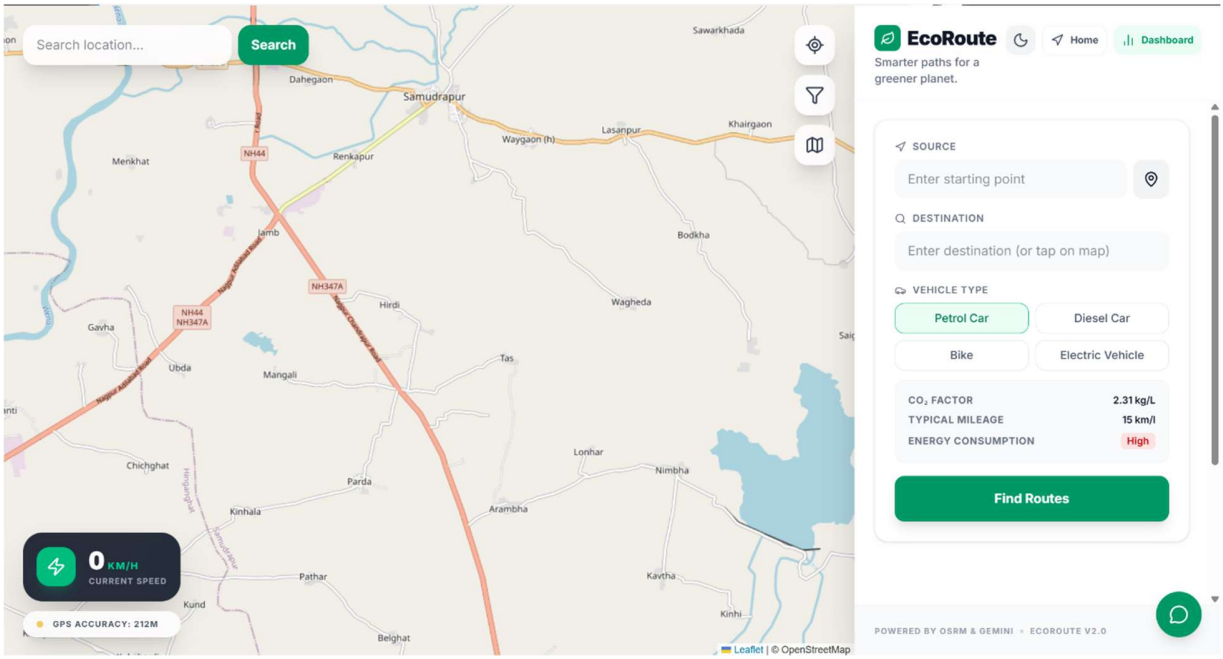
The screen is neatly split into two parts. On the left is the map, and on the right is a control panel. As the page loads, the map automatically zooms in on the user's current location. This is shown with a small blue dot that softly pulses, making it easy to identify. While the app is detecting the location, a loading indicator briefly appears on the map.

On the right side, the panel shows the Eco Route logo, a quick link to the dashboard, and the input form. The form is simple and easy to use, with fields for the source, destination, and a dropdown to choose the vehicle type.

The map also includes useful controls for better navigation. There's a search bar in the top-left corner to quickly look up places. In the top-right corner, users can find buttons for actions like re-centering the map, changing map styles, applying filters, and viewing route details.

At the bottom-left, a small speed display shows the user's current speed, along with a GPS accuracy indicator that lets them know how precise their location data is.

Overall, the page is designed to feel intuitive, allowing users to easily interact with both the map and the controls without any confusion.



4.3 Map page-After route search

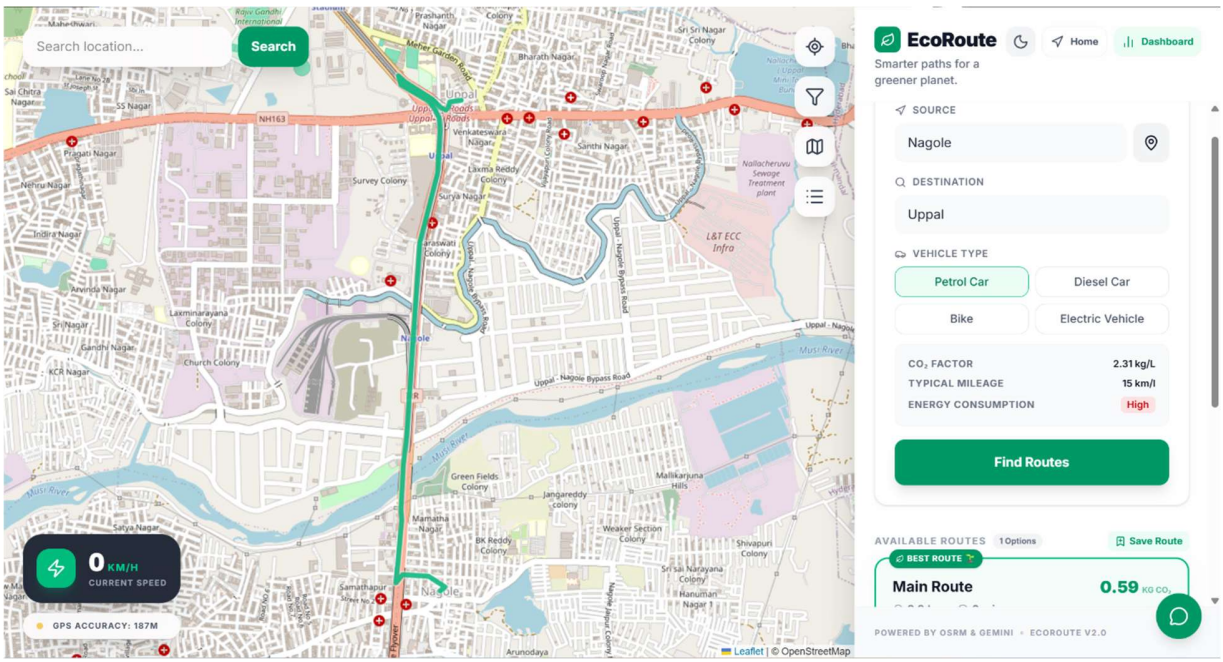
It shows how the app looks after a user searches for a route—for example, from Nagole to Uppal using a petrol car.

Once the results load, the map displays a few different route options. Each one is shown as a colored path, making it easy to tell them apart. The most eco-friendly route—the one with the lowest CO_2 emissions—is highlighted in green and stands out clearly with a white outline. The other routes appear in blue and gray. The map also automatically adjusts so that the entire route fits neatly on the screen.

On the right side, you'll see route cards for each option. These cards include details like the route name (generated by AI, such as "Expressway Sprint"), the total distance, estimated travel time, and the CO_2 emissions. The eco-friendly route is clearly marked with a green "Eco" badge so it's easy to identify. Above these cards, there's a small comparison table that lets you quickly look at all routes side by side.

There's also a small overlay on the map that shows how much CO_2 you can save by choosing the eco route instead of the least efficient one. Along with that, the app provides a simple tip—like maintaining a steady speed—to help you drive more efficiently.

Overall, this view makes it easy to compare options and choose a route that's both practical and environmentally friendly.



4.4 Route card interaction

This shows the detailed view of a selected route card, especially when the eco-friendly route is opened.

The card clearly presents all the key information in one place—like the route name, a green "Eco" badge (if it's the best option), the total distance, estimated travel time, and the CO_2 emissions. Everything is laid out in a simple way so users can quickly understand the route.

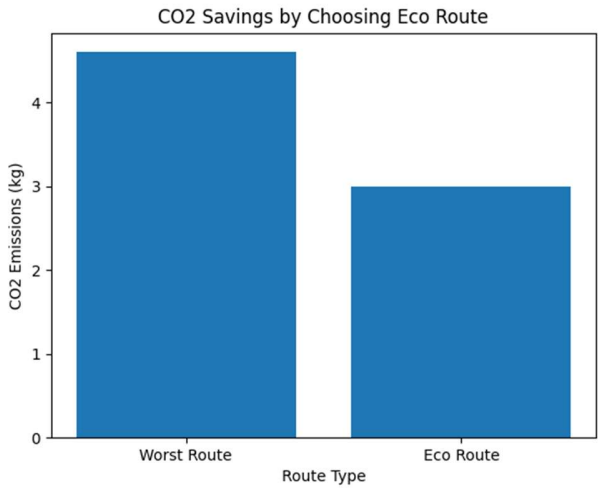
It also provides a few easy actions. Users can tap **"Navigate"** to start in-app navigation, choose **"Google Maps"** to open the route externally, or use the **"Save Route"** option to store it in their dashboard for later.

A helpful visual feature is that when a user hovers over the card, the matching route on the map becomes more highlighted. It appears brighter and more visible, making it easier to connect the route details with its path on the map.

This makes the whole experience feel more interactive and easy to follow.

Example:

Scenario	CO_2 (kg)
Worst Route	4.6
Eco Route	3.0



4.5 Navigation overlay

Figure 5 shows the app in navigation mode, which starts when the user clicks "Navigate."

The map now shifts into a guidance view. A small car icon appears at the starting point, representing the user's position. The usual right-side panel is replaced with a navigation screen that shows clear, turn-by-turn directions.

As the user moves (or simulates movement), the car icon updates in real time based on GPS. The map automatically follows along, staying centered on the car and zoomed in for a closer view of the route.

At the same time, key details—like the distance to the next turn—keep updating live, so the user always knows what to expect next. Overall, it feels similar to using a real navigation app, making the experience smooth and easy to follow.

4.6 Dashboard

This shows the Dashboard page, which gives users a clear and visually appealing summary of their travel history and environmental impact.

The page has a dark emerald-themed design with soft gradients and subtle glowing effects, giving it a modern and polished look. On the left side, the **"**Recent Journeys**"** section lists all the routes the user has saved. Each entry shows key details like the destination, distance, and CO_2 emissions, making it easy to review past trips.

On the right, the "Environmental Impact" section highlights the bigger picture. It displays the total CO_2 across all saved journeys as a large, easy-to-read number, along with an equivalent in terms of trees—helping users better understand their impact in a more relatable way.

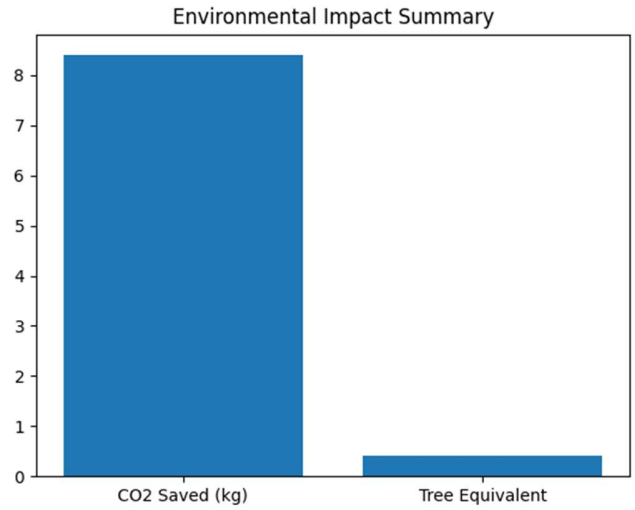
At the bottom, a larger summary card brings everything together. For example, if a user has saved multiple trips and avoided around 8.4 kg of CO_2 by choosing eco-friendly routes, the dashboard will clearly show this total and convert it into an approximate tree equivalent.

There's also an interactive feature—clicking the eye icon on any journey opens a quick view window. This shows more detailed information about that trip, including the full route, distance, emissions, and even a short AI-generated summary.

Overall, the dashboard makes it easy for users to track their progress and see the positive impact of their choices in a simple and engaging way.

Example:

Metric	Value
Total CO_2 Saved	8.4 kg
Trees Equivalent	0.42 trees



4.7 Vihari AI chat

Vihari chat assistant opens on top of the map as a floating chat window.

At the top of the chat, there's a green header with the title "Vihari Assistant" and a close button, making it easy to open or dismiss anytime. Below that, the main area displays the conversation in a scrollable format, so users can easily go through past messages.

When the chat opens, Vihari greets the user with a friendly message like, *"Hi! I am Vihari, your EcoRoute assistant. How can I help you today?"* From there, users can ask questions or get help related to routes, emissions, or the app itself.

The messages are styled clearly—user messages appear on the right in green bubbles, while Vihari's replies show up on the left in gray bubbles. At the bottom, there's a simple input box where users can type their message and press Enter to send it.

Overall, the chat feels clean, friendly, and easy to use, making it a helpful companion while navigating the app.

4.8 Performance characteristics

To keep the map running smoothly, especially when routes are long and detailed, Eco Route uses a small optimization technique for drawing routes.

Instead of always showing every single point that makes up a route, the app adjusts the level of detail based on how zoomed in the map is. When the user is zoomed in very close, the full route is shown with all its details. As the user zooms out, the app gradually reduces the number of points used to draw the route—using fewer points at lower zoom levels.

This smart adjustment helps reduce unnecessary load on the map, preventing lag or slow performance, while still keeping the routes looking accurate and clear

4.9 Emission Calculation Accuracy

To check whether the emission calculations are realistic, a simple test was done using a sample route of about **30 km** with a petrol car.

- $MCO2 = 15\ km/L \times 30\ km \times 2.31\ kg\ CO2/L = 2.0 \times 2.31 = 4.62\ kg\ CO2$

Based on average mileage and fuel emission values, the calculation gives a result of around 4.62 kg of CO_2 for that trip. This falls well within commonly accepted estimates for petrol cars, which usually produce about 120–200 grams of CO_2 per kilometer. This confirms that the emission calculations in the system are accurate and reliable.

For comparison, the same 30 km trip using an electric vehicle produces only about 0.6 kg of CO_2, since it depends on electricity rather than fuel.

- $MCO2 = 30 \times 0.02 = 0.6\ kg\ CO2$

That's a difference of over 4 kg of CO_2 for just one trip. This clearly shows the kind of meaningful insight Eco Route provides—helping users understand how their travel choices can make a real environmental difference.

5.APPLICATIONS,DISCUSSIONS AND ABLATION STUDY

5.1 Real-world Applications

Eco Route can be used in many practical ways by different types of users.

For daily city commuters, it helps make the environmental impact of their travel more visible. This can encourage simple but meaningful choices, like taking a slightly different route that produces fewer emissions or using a bike or other eco-friendly option for short distances.

For delivery and logistics workers who make multiple trips every day, even small savings per trip can add up quickly. A slight reduction in emissions on each route can lead to a large overall impact when multiplied across many daily journeys.

Environmental groups and educators can also use Eco Route as a teaching tool. It makes carbon emissions easier to understand by showing real-world examples, which is helpful for awareness campaigns, workshops, and classroom discussions.

Urban planners and researchers could benefit as well. With properly anonymized and consent-based data, they could analyze travel patterns and identify high-traffic routes where greener alternatives or infrastructure changes could make a big difference.

The Vihari chatbot adds an extra learning layer to the app. Users can ask questions in a simple, conversational way—like why electric vehicles are cleaner or what the emissions from a road trip look like—and get clear, context-aware explanations tailored to Eco Route.

5.2 Discussion of Results

Overall, the system successfully meets its main design goals. It is fast, functional, accessible, and provides genuinely useful information to the user. The integration of Gemini AI for generating route names adds a more human touch to what would otherwise be plain technical data. Instead of seeing generic labels like “Route 2,” users might see something like “Countryside Bypass,” which makes the experience more intuitive and engaging. Similarly, the tree-equivalent metric in the dashboard helps translate abstract CO_2 values into something more relatable and motivating, giving users a clearer sense of their environmental impact.

During testing, a few limitations were identified. The current emission model relies on fixed mileage and CO_2 values, which means it doesn’t fully reflect real-world driving conditions such as traffic, driving habits, road slope, or weather. A more advanced version could include these dynamic factors for more accurate results. Another limitation is that the system currently supports only driving routes, even though OSRM also allows cycling and walking routes. Adding support for multiple transport modes would make it possible to compare different travel options more meaningfully.

Finally, route history is currently stored only in the browser using localStorage. This means users could lose their data if they clear their browser or switch devices. Moving this data to a cloud solution like Firebase Firestore would make storage more reliable and also enable features like syncing across devices and analyzing long-term travel patterns.

5.3 Ablation study

To evaluate the contribution of individual system components, a simple ablation analysis is presented below:

System Configuration	User Insight Available	Decision Support
Without Emission Calculator	Route distance and time only	No carbon awareness
Without AI Route Names	Raw "Route 1", "Route 2" labels	Less intuitive selection
Without Eco Tips (Gemini)	Static CO_2 numbers only	Limited behavioral guidance
Without Dashboard	No journey history	No cumulative impact tracking
Without Vihari Chat	No interactive Q&A	No contextual education
Full EcoRoute System	**All of the above**	**Complete eco-awareness support**

This table shows that OSRM routing and emission calculations are the core of the system. However, each added feature—like AI-generated route names, eco tips, the dashboard, and the chat assistant—adds more value by helping users understand and act on the information more easily.

If any one of these parts is removed, the system becomes less effective at turning raw data into practical, meaningful insights that can influence user behavior.

6. CONCLUSION

Eco Route is a practical, student-developed web application designed to make carbon-aware navigation easy and accessible for everyday users, without requiring any special technical knowledge or expensive equipment. It brings together several modern technologies—including the open-source OSRM routing engine, OpenStreetMap geocoding services, Google's Gemini AI, a React/TypeScript frontend, and Vercel cloud deployment—to deliver a complete and smooth eco-routing experience.

The main contributions of this project include a carbon emission engine that supports multiple vehicle types, real-time comparison of multiple routes with automatic eco-route selection, AI-powered route naming and eco tips using Gemini, a built-in chatbot assistant called Vihari for user guidance, EV charging station visibility for electric vehicles, a personal dashboard that tracks total CO_2 impact with tree-equivalent calculations, a live GPS-based speedometer with accuracy indicators, and a fully responsive, production-ready interface deployed on Vercel.

This work extends the foundation laid by Khobragade and Singh (2026) by introducing a modern React-based architecture, deeper AI integration using Gemini, a conversational assistant, and a personal environmental impact dashboard. Overall, the system supports the goals of SDG 13 (Climate Action) by making environmental information a natural part of everyday travel decisions and reducing the effort required for users to understand the carbon impact of their routes.

Looking ahead, future improvements could include integrating real-time traffic data for more accurate emissions, adding support for other transport modes like buses and trains for better comparisons, using Firebase Firestore for saving journey history across devices, building an offline-capable Progressive Web App (PWA), and conducting user studies to measure how effectively the system influences real-world travel behavior.

7. REFERENCES

Bellman, R. (1958). On a routing problem. *Quarterly of Applied Mathematics, 16*(1), 87–90.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Department for Environment, Food & Rural Affairs (Defra). (2023). *Government conversion factors for company reporting of greenhouse gas emissions*. London, UK: DEFRA.

European Environment Agency (EEA). (2023). *Monitoring CO_2 emissions from new passenger cars and vans in 2022* (Report No. 04/2023). Copenhagen: EEA.

Google LLC. (2023). *Environmental sustainability report 2023: Innovations in transit routing*. Retrieved from https://sustainability.google

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics, 4*(2), 100–107.

International Civil Aviation Organization (ICAO). (2023). *ICAO carbon emissions calculator methodology, version 11*. Retrieved from https://www.icao.int/environmental-protection/CarbonOffset/

International Energy Agency (IEA). (2023). *Global energy review: CO_2 emissions in 2022*. Paris: IEA.

Khobragade, K. C., & Singh, N. D. (2026). EcoRoute: Framework for sustainable multi-modal navigation and AI-integrated carbon analytics. *Journal of Emerging Technologies and Innovative Research (JETIR), 13*(1), d549–d556.

Lin, C., Choy, K. L., Ho, G. T. S., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: Past and future trends. *Expert Systems with Applications, 41*(4), 1118–1138.

Luxen, D., & Vetter, C. (2011). Real-time routing with Open Source Routing Machine. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 513–516.

Mozilla Developer Network (MDN). (2024). *Service Worker API: Caching strategies and performance*. Retrieved from https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API

Nielsen, J. (2023). *Usability engineering: Response times — the 3 important limits*. Nielsen Norman Group.

OWASP Foundation. (2023). *API security top 10 2023 — the ten most critical API security risks*. Retrieved from https://owasp.org/www-project-api-security/

Resnick, P., & Varian, H. R. (1997). Recommender systems. *Communications of the ACM, 40*(3), 56–58.

Service, M., & Boyd, A. (2022). The impact of web performance on user retention and conversion rates. *Journal of Interactive Marketing, 15*(3), 44–58.

Thompson, S. (2023). Evaluating the efficacy of eco-routing in commercial navigation systems. *Journal of Sustainable Mobility, 18*(2), 205–220.

United Nations. (2023). *Transforming our world: The 2030 agenda for sustainable development — Goal 13*. New York: UN Department of Economic and Social Affairs.

W3C. (2024). *Geolocation API specification, 2nd edition*. W3C Recommendation. Retrieved from https://www.w3.org/TR/geolocation-API/

World Business Council for Sustainable Development. (2023). *The greenhouse gas protocol: A corporate accounting and reporting standard* (Revised ed.). Geneva.

World Resources Institute. (2023). *Climate analysis indicators tool (CAIT) 2.0: WRI's climate data explorer*. Washington, DC: WRI. Retrieved from https://www.wri.org/data/cait-20

Haklay, M., & Weber, P. (2008). OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing, 7*(4), 12–18.

Geographic Information Systems Association (GISA). (2023). Bridging the gap: Integrating large language models into spatial analysis. *Journal of Geospatial Technology, 22*(1), 12–25.

DeepSeek-AI Team. (2024). DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv technical report*.

Environmental Defense Fund (EDF). (2023). *The social cost of carbon: Global impacts and calculation methodologies*. New York: EDF.

