



Developer Guide: Understanding & Extending the Carbon Footprint Calculator

This guide is designed to help **new contributors**, especially **beginners**, understand the structure and logic of this **front-end-only web app**—and how to modify or extend it.



Project Structure (Frontend Only)

This app is built using **vanilla HTML, CSS, and JavaScript**. No backend, frameworks, or databases are involved.

```
└─ project-root/
├─ index.html           ← UI layout (main structure)
├─ style.css            ← App styling and layout rules
├─ script.js           ← Core logic (suggestions, calculator, map)
├─ leaflet.js, leaflet.css ← Leaflet map library (via CDN in HTML)
└─ └─ assets/          ← (Optional) Icons, images, or assets
```

🔧 Core Components Overview

1. index.html

Contains:

- ◆ Input fields (product & locality)
 - ◆ Autocomplete suggestions dropdown
 - ◆ "Get Result" button
 - ◆ Result section (footprint + eco-tip)
 - ◆ Map container (Leaflet-powered)
-

2. style.css

Handles:

- 🎨 Dark-themed UI color palette
- 📐 Responsive layout using **Flexbox**
- 🖼️ Panel layout:
 - Left: Inputs & results
 - Right: Interactive map

3. script.js

Contains all app logic and interactivity.

Major Functions:

Function	Purpose
<code>autoSuggest()</code>	Shows dynamic product suggestions
<code>autoSuggestLocality()</code>	Suggests locations based on input
<code>getCombinedFootprint()</code>	Computes footprint based on product \times location
<code>showMap()</code>	Renders map marker using Leaflet
<code>getRandomTip()</code>	Displays a random eco-tip



Footprint Formula (Simplified Logic)

 This is a **placeholder approximation**, not based on real-world data.


```
footprint = baseFootprint[product]  $\times$  (localityMultiplier[location] / 10)
```




- All values are stored in two JavaScript objects:

```
const baseFootprints = { ... }  
const localityMultipliers = { ... }
```

- Found at the top of `script.js`.
-

Map Features (Leaflet.js)

- Uses **OpenStreetMap + Leaflet.js** to render map of **Bengaluru localities**.
-  When user clicks "Get Result":
 - Map centers on selected location.
 - Color-coded marker appears based on emission level:

Color	Meaning
 Red	High emissions
 Orange	Moderate emissions
 Green	Low emissions








How to Run the App

1. No build tools needed.
 2. Just **open** `index.html` in any browser (Chrome, Edge, Firefox).
 3. Ensure:
 - JavaScript is enabled.
 - Browser allows API/CDN requests.
-

+ Beginner-Friendly Contribution Ideas




Here are some small, fun tasks perfect for new developers:

-  Add more products or locations in the data objects
 -  Enhance styling with transitions or gradient effects
 -  Expand map logic to include cities like Mumbai, Delhi
 -  Improve suggestions using trie/fuzzy search algorithms
 -  Add “Save Result” or “Share Report” functionality
-



Troubleshooting Tips

If things break:

-  Clear inputs and retry.
 -  Open browser dev tools (F12 → Console tab) to check for errors.
 -  Use `console.log()` in `script.js` to debug.
-



Final Notes

- This is a **Minimum Viable Product (MVP)** — simple, readable code.
- Great for learning **data + map + UI interactions**.
- Not intended for production or climate science use.
- Perfect for hackathons, education, and prototyping.