I A R E

EDUCATION FOR LIBERATION

INSTITUTE OF AERONAUTICAL ENGINEERING

(AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

Complex Problem-Solving Self-Assessment Form

1	Name of the Student	**BHANU RISHIKESH REDDY PUCHALAPALLI**	
2	Roll Number	**25951A6632**	
3	Branch and Section	CSE-(AI&ML) - B	
4	Program	B. Tech	
5	Course Name	FRONT END WEB DEVELOPMENT LABORATORY	
6	Course Code	ACSE04	
7	Please tick (✓) relevant Engineering Competency (ECs) Profiles		
	EC	**Profiles**	**(✓)**
	EC 1	Ensures that all aspects of an engineering activity are soundly based on fundamental principles - by diagnosing, and taking appropriate action with data, calculations, results, proposals, processes, practices, and documented information that may be ill-founded, illogical, erroneous, unreliable or unrealistic requirements applicable to the engineering discipline	✓
	EC 2	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
	EC 3	Support sustainable development solutions by ensuring functional requirements, minimize environmental impact and optimize resource utilization throughout the life cycle, while balancing performance and cost effectiveness.	
	EC 4	Competently addresses complex engineering problems which involve uncertainty, ambiguity, imprecise information and wide-ranging or conflicting technical, engineering and other issues.	✓
	EC 5	Conceptualises alternative engineering approaches and evaluates potential outcomes against appropriate criteria to justify an optimal solution choice.	✓
	EC 6	Identifies, quantifies, mitigates and manages technical, health, environmental, safety, economic and other contextual risks associated to seek achievable sustainable outcomes with engineering application in the designated engineering discipline.	
	EC 7	Involve the coordination of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies) in the timely delivery of outcomes	
	EC 8	Design and develop solution to complex engineering problem considering a very perspective and taking account of stakeholder views with widely varying needs.	✓
	EC 9	Meet all level, legal, regulatory, relevant standards and codes of practice, protect public health and safety in the course of all engineering activities.	

	EC 10	High level problems including many component parts or sub-problems, partitions problems, processes or systems into manageable elements for the purposes of analysis, modelling or design and then re-combines to form a whole, with the integrity and performance of the overall system as the top consideration.	✓
	EC 11	Undertake CPD activities to maintain and extend competences and enhance the ability to adapt to emerging technologies and the ever-changing nature of work.	✓
	EC 12	Recognize complexity and assess alternatives in light of competing requirements and incomplete knowledge. Require judgement in decision making in the course of all complex engineering activities.	✓
8	Please tick (✓) relevant Course Outcomes (COs) Covered		
	CO	**Course Outcomes**	**(✓)**
	CO 1	Describe language basics like alphabet, strings, grammars, productions, derivations, and Chomsky hierarchy, construct DFA, NFA, and conversion of NFA to DFA, Moore and Mealy machines and interpret differences between them.	✓
	CO 2	Recognize regular expressions, formulate, and build equivalent finite automata for various languages.	✓
	CO 3	Identify closure, and decision properties of the languages and prove the membership.	✓
	CO4	Demonstrate context-free grammars, check the ambiguity of the grammar, and design equivalent PDA to accept the context-free languages.	
	CO 5	Uses mathematical tools and abstract machine models to solve complex problems.	✓
	CO 6	Analyze and distinguish between decidable and undecidable problems.	✓
9	Course ELRVVideo Lectures Viewed	**Number of Videos**	**Viewing time in Hours**
		-	-
10	Justify your understanding of WK1	-	
11	Justify your understanding of WK2 – WK9	-	
12	How many WKs from WK2 to WK9 were implanted?	-	
	Mention them	-	

Date: 27-11-2025

BHANU RISHIKESH

Signature of the Student

INSTITUTE OF AERONAUTICAL ENGINEERING
(AUTONOMOUS)
Dundigal - 500 043, Hyderabad, Telangana

TASK ASSESSMENT

ON

ECO-TRACK

RISHI

25951A6632

IARE

INSTITUTE OF AERONAUTICAL ENGINEERING

(AUTONOMOUS)

Dundigal - 500 043, Hyderabad, Telangana

EDUCATION FOR LIBERATION

ECO – TRACK

A project report submitted in partial fulfillments
of requirements for the award of the
degree of Bachelor of Technology in

CSE (Artificial Intelligence & Machine Learning)

By

BHANU RISHIKESH REDDY PUCHALAPALLI

25951A6632

DECLARATION

I certify that

a. The work contained in this report is original and has been done by me under the guidance of my supervisor (s).
b. The work has not been submitted to any other Institute for any degree or diploma.
c. I have followed the guidelines provided by the Institute for preparing the report.
d. I have conformed to the norms and guidelines given in the Code of Conduct of the Institute.
e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

RISHI

Place: Hyderabad **Signature of the Student**

Date: 11-12-2025

CERTIFICATE

This is to certify that the project report entitled **ECO-TRACK** submitted by **BHANU RISHIKESH REDDY PUCHALAPALLI** to the Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **CSE**
- (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING) is a Bonafide record of work carried out by his guidance and supervision. The Contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Supervisor **Head of the Department**

Date: 11-12-2025

Principal

APPROVAL SHEET

This project report entitled **ECO-TRACK** submitted by **BHANU RISHIKESH REDDY PUCHALAPALLI** is approved for the award of the Degree Bachelor of Technology in Branch **CSE (Artificial Intelligence & Machine Learning).**

Examiner **Supervisor(s)**

Principal

Date: 11-12-2025

Place: Hyderabad

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

I am extremely grateful and express my profound gratitude and indebtedness to my project guide **Dr .M NAGARAJU, Assistant Professor and Deputy Head of Department,** Department of **CSE (Artificial Intelligence & Machine Learning),** for his kind help and for giving me the necessary guidance and valuable suggestions for this project work.

I am grateful to **Dr. M. PURUSHOTHAM REDDY**, **Professor** and **Head of the Department**, Department of **CSE (Artificial Intelligence & Machine Learning),** for extending his support to carry on this project work. I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

I express my sincere gratitude to **Dr. L. V. NARASIMHA PRASAD, Professor** and **Principal**
who has been a great source of information for my work.

I thank our college management and respected **Sri M. RAJASHEKAR REDDY**, **Chairman**, **IARE, Dundigal** for providing me with the necessary infrastructure to conduct the project work.

I take this opportunity to express my deepest gratitude to one and all who directly or indirectly helped me in bringing this effort to present form.

ABSTRACT

Climate change has emerged as one of the most pressing global issues, largely driven by increased carbon emissions resulting from human activities. Despite the availability of scientific data and environmental studies, most individuals lack awareness of how their daily habits contribute to carbon emissions. ECOTRACK is a web-based personal carbon footprint calculator designed to bridge this awareness gap by providing users with a simple, interactive, and visually engaging platform.

The primary aim of ECOTRACK is to calculate an individual's carbon footprint based on inputs such as transportation methods, electricity consumption, food habits, and lifestyle choices. The system processes this information using predefined emission factors and presents the results in both numerical and graphical formats. Charts and visual indicators make the data easier to understand and interpret, allowing users to identify major sources of emissions in their daily lives.

The project not only demonstrates technical skills in HTML, CSS, and JavaScript but also highlights the role of technology in promoting sustainability. ECOTRACK ultimately aims to educate users, encourage environmentally responsible behavior, and contribute to a more sustainable future through informed decision-making.

.

CONTENTS

CHAPTER 1: INTRODUCTION

1.1 PROBLEM STATEMENT

Although climate change awareness is increasing, most individuals do not have a clear understanding of their personal contribution to carbon emissions. Existing tools for carbon footprint calculation are often complex, lack proper visualization, or are not user-friendly. Many people find it difficult to interpret raw numerical data related to emissions, which reduces the effectiveness of such tools.

The problem addressed by ECOTRACK is the lack of a simple, interactive, and visually appealing platform that allows users to calculate and understand their carbon footprint easily. Without proper awareness and visualization, individuals are less likely to change their habits or adopt sustainable practices. There is a need for a front-end web solution that bridges the gap between environmental data and user understanding.

ECOTRACK aims to solve this problem by providing a clean user interface, easy data input forms, accurate calculations, and graphical representations of results. The application focuses on educating users and motivating them to make environmentally responsible choices.

1.2 INTRODUCTION

Sustainability has become a key focus area in modern technological development due to increasing environmental degradation and climate change. Carbon footprint refers to the total amount of greenhouse gases emitted directly or indirectly by human activities, usually measured in terms of carbon dioxide equivalents.

Activities such as driving vehicles, using electricity generated from fossil fuels, consuming processed food, and excessive use of electronic devices significantly contribute to carbon emissions.

In today’s digital era, web-based applications play an important role in spreading awareness and providing solutions to global issues. ECOTRACK is a sustainability-oriented front-end web application that helps individuals calculate their personal carbon footprint.

The application not only provides numerical results but also represents the data visually using charts, making it easier for users to understand their environmental impact.

ECOTRACK combines environmental science concepts with modern web technologies to deliver an informative and interactive platform. The project emphasizes simplicity, accessibility, and usability, ensuring that users from different backgrounds can use the application effectively.

1.3 REQUIREMENTS

# Task Manager

- The interface must allow users to add tasks with categories, due dates, priority levels.
- Users must be able to edit, update, and delete tasks.
- Tasks must visually indicate status (pending, completed, overdue).
- FR4: Client-side storage (Local Storage/Session Storage) must retain tasks across sessions.

# Goal Tracker

- FR5: Users must be able to create short-term and long-term goals.
- FR6: The dashboard should visually show progress (bars, percentages, or icons).
- FR7: Users must be able to update milestones dynamically.

# Motivational Quote Engine

- FR8: The dashboard must display a new quote dynamically each day or on refresh.
- FR9: Users must be allowed to save favorite quotes for later viewing.

# Dashboard Overview

- FR10: The main screen must summarize tasks due, goals progress, and the daily quote.
- FR11: The interface must be fully interactive, with instant UI updates using JS

1.4 PRE-REQUISITES

1. Technical Pre-requisites
- HTML5 for structure and UI layout
- CSS3 for responsive design, themes, and styling
- JavaScript for dashboard logic, dynamic content updates, and storage handling
- Knowledge of DOM manipulation
- Familiarity with responsive frameworks (optional)

2. Tool Pre-requisites
- Code editor (VS Code, Sublime, Atom)
- Modern browsers (Chrome, Edge, Firefox)
- Version control (Git/GitHub)
- UI planning tools like Figma (optional)

3. User Pre-requisites
- Basic understanding of creating tasks/goals
- Familiarity with reading dashboards and progress indicators
- Ability to use simple web interfaces

1.5 TECHNOLOGIES USED

The core structure of the application is created using **HTML (HyperText Markup Language)**. HTML is responsible for defining the layout of the web pages, including headings, forms, input fields, buttons, and containers. It provides the basic skeleton of the application and ensures proper organization of content.

CSS (Cascading Style Sheets) is used to enhance the visual appearance of the application. CSS plays a crucial role in designing a clean and user-friendly interface by controlling colors, fonts, spacing, alignment, and responsiveness. Media queries are used to ensure that the application works smoothly on different screen sizes such as desktops, tablets, and mobile devices.

JavaScript is the core technology used to implement the functionality of ECOTRACK. It handles user input, validates form data, performs carbon footprint calculations, and dynamically updates results without refreshing the page. JavaScript also enables interaction between different components of the application.

For data visualization, JavaScript chart libraries such as **Chart.js** are used. These libraries help display carbon footprint data in the form of charts and graphs, making the output more understandable and visually engaging for users.

CHAPTER 2 - REVIEW OF RELEVANT LITERATURE

LITERATURE REVIEW / EXISTING SYSTEM

Several carbon footprint calculators already exist, such as those provided by government agencies and environmental organizations. These tools often provide accurate results but may lack user-friendly interfaces or engaging visualizations. Some calculators require extensive data input, which discourages users from completing the process.

Most existing systems focus on accuracy rather than usability. Many tools display results only in numerical form, making it difficult for users to interpret the data. Additionally, some platforms are not optimized for mobile devices, limiting accessibility.

ECOTRACK improves upon these limitations by offering a simplified interface, minimal data input, and attractive charts. The use of front-end technologies ensures better performance and accessibility, making the application suitable for a wider audience.

PROPOSED SYSTEM

The proposed system, ECOTRACK, is a front-end web application that calculates personal carbon footprint using user input and predefined emission factors. The system focuses on simplicity, interactivity, and visualization. Users enter data through structured forms, and the application processes the data using JavaScript logic.

The calculated carbon footprint is displayed using charts such as bar charts or pie charts. This visual representation helps users quickly understand which activities contribute most to their emissions. The system also provides brief suggestions for reducing carbon footprint based on the results.

SYSTEM ARCHITECTURE

ECOTRACK follows a simple client-side architecture. The application consists of three main layers: the presentation layer, the logic layer, and the visualization layer. The presentation layer is responsible for displaying the user interface using HTML and CSS. The logic layer processes user input and performs carbon footprint calculations using JavaScript. The visualization layer generates charts and graphs to represent the results.

Since the project is limited to front-end development, all operations are performed on the client side without server interaction. This architecture ensures fast response time and ease of deployment.

CHAPTER 3: METHODOLGY

The development of ECOTRACK follows a systematic and structured methodology to ensure accuracy, usability, and effectiveness. The first step involved understanding the problem and identifying the requirements of the system. This included analyzing sustainability concepts, carbon footprint calculation methods, and user expectations. Based on this analysis, the functional and non-functional requirements of the application were defined.

The next phase focused on designing the user interface. Wireframes and layout ideas were created to ensure a simple and intuitive user experience. The goal was to minimize user effort while entering data and maximize clarity in displaying results. HTML and CSS were then used to implement the interface design.

After the UI design, JavaScript logic was developed to handle user inputs and perform carbon footprint calculations using predefined emission factors. Input validation was implemented to avoid incorrect or missing data. Once calculations were complete, chart libraries were integrated to visualize the results.

The final phase involved testing the application for functionality, responsiveness, and usability across different browsers and devices. Any errors identified during testing were fixed to improve reliability. This methodology ensured a smooth development process and a functional final product.

3.1 OBJECTIVES

The primary objective of the ECOTRACK project is to design and develop a front-end web application that calculates an individual’s personal carbon footprint. The project aims to create awareness about carbon emissions and encourage users to adopt sustainable habits in their daily lives. By providing a clear breakdown of emissions from various activities, ECOTRACK helps users identify areas where they can reduce their environmental impact.

Another important objective is to apply front-end web development concepts such as responsive design, client-side scripting, and data visualization in a practical scenario. The project enables students to gain hands-on experience with HTML, CSS, and JavaScript while addressing a real-world problem. It also focuses on improving user experience through simple navigation, clean layouts, and interactive elements.

Additionally, ECOTRACK aims to present carbon footprint data in an understandable and engaging manner using charts and graphs. Visualization plays a key role in helping users interpret data effectively. Overall, the project seeks to combine technical learning with social responsibility, demonstrating how web technologies can contribute to environmental sustainability.

3.2 FEATURES

ECOTRACK offers several features that make it an effective and user-friendly carbon footprint calculator. One of the key features is its simple and intuitive user interface, which allows users to enter their daily activity data easily without technical complexity. Clear labels and structured forms guide users through the input process.

Another important feature is real-time carbon footprint calculation. As soon as the user submits the data, the application processes the input and displays the results instantly. This quick response enhances user engagement and satisfaction.

The application also includes interactive data visualization using charts and graphs. These visual elements help users clearly understand how different activities contribute to their overall carbon footprint. The graphical representation makes complex data easier to interpret.

ECOTRACK is designed to be responsive, meaning it works effectively on desktops, laptops, tablets, and mobile devices. The application is also lightweight and does not require any backend server or database, ensuring fast performance.

Additionally, ECOTRACK serves an educational purpose by increasing awareness about sustainability and environmental responsibility. These features together make ECOTRACK a valuable front-end web development project with real-world relevance.

3.3 ADVANTAGES

ECOTRACK provides several advantages that make it an effective sustainability-focused web application. One of the major advantages is its **ease of use**. The application is designed for users with minimal technical knowledge, allowing anyone to calculate their carbon footprint without difficulty.

Another key advantage is that ECOTRACK is a **client-side application**, meaning it does not require backend servers or databases. This reduces complexity, improves performance, and makes deployment easier. The application runs entirely in the user’s browser, ensuring faster response times.

The use of **visual charts and graphs** is another significant advantage. Visualization helps users quickly identify the major contributors to their carbon emissions, making the results more meaningful and actionable. Compared to plain numerical data, graphical representation improves understanding and engagement.

ECOTRACK also promotes **environmental awareness** by educating users about the impact of their daily habits. It encourages sustainable decision-making and responsible behavior. From an academic perspective, the project helps students gain practical experience in front-end web development while addressing an important global issue.

3.4 DISADVANTAGES / DRAWBACKS

Despite its advantages, ECOTRACK has certain limitations. One major drawback is that the carbon footprint calculations are approximate. The results depend on predefined emission factors and user-provided data, which may not always be accurate. Small errors in input can lead to variations in output.

Another limitation is the absence of data storage. Since ECOTRACK is a front-end-only application, user data is not saved for future reference. Users cannot track changes in their carbon footprint over time or compare historical data.

The application also has a limited scope of emission categories. Only selected activities such as transportation, electricity usage, and food consumption are considered. Other factors like waste management, water usage, and industrial consumption are not included.

Additionally, ECOTRACK does not provide personalized recommendations based on user behavior. Suggestions for reducing carbon footprint are general in nature. These drawbacks highlight areas where the application can be improved in future versions.

3.5 APPLICATIONS

ECOTRACK has several practical applications in both educational and awareness-based environments. In educational institutions, the application can be used as a learning tool to teach students about sustainability, climate change, and the environmental impact of human activities. It can also serve as a demonstration project in web development courses.

Environmental organizations and sustainability campaigns can use ECOTRACK to spread awareness among the public. The application can be used during workshops, seminars, and awareness programs to help individuals understand their carbon footprint.

ECOTRACK can also be used by individuals who want to monitor and reduce their environmental impact. By understanding emission sources, users can make informed lifestyle choices. These applications highlight the relevance and usefulness of ECOTRACK in real-world scenarios.

3.4 CODE

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>EcoTrack — Personal Carbon Tracker</title>

  <!-- Chart.js -->
  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

  <style>
    /* ---------- Light pastel theme ---------- */
    :root{
      --bg: #fbfbfd;
      --panel: #ffffff;
      --muted: #64748b;
      --accent: #7cb4f5;    /* sky */
      --accent-2: #a6e6c3;  /* mint */
      --danger: #ff7b7b;
      --soft: #f3f6ff;
      --radius: 12px;
      --shadow: 0 6px 18px rgba(16,24,40,0.06);
      --maxw: 1100px;
      font-family: Inter, system-ui, -apple-system, "Segoe UI", Roboto,
   "Helvetica Neue", Arial;
    }
    *{box-sizing:border-box}
    body{
      margin:0;
      padding:28px;
      background:linear-gradient(180deg,var(--bg), #f7fbff 160%);
      color:#0f1724;
      -webkit-font-smoothing:antialiased;
      -moz-osx-font-smoothing:grayscale;
      min-height:100vh;
      display:flex;
  justify-content:center;
  align-items:flex-start;
 }
 .wrap{
  width:100%;
  max-width:var(--maxw);
  display:grid;
  grid-template-columns:360px 1fr;
  gap:20px;
 }

 /* left panel (controls) */
 .panel{
  background:var(--panel);
  border-radius:var(--radius);
  box-shadow:var(--shadow);
  padding:18px;
 }
 .brand{
  display:flex;gap:12px;align-items:center;margin-bottom:12px;
 }
 .logo{
  width:48px;height:48px;border-radius:10px;background:linear-gradient(135deg,var(--accent),var(--accent-2));
  display:flex;align-items:center;justify-content:center;color:white;font-weight:700;font-size:18px;
 }
 h1{font-size:18px;margin:0}
 p.lead{margin:6px 0 12px;color:var(--muted);font-size:13px}

 label{display:block;font-size:13px;color:#0f1724;margin-bottom:6px;font-weight:600}
 .field{margin-bottom:12px}
 input[type="number"], input[type="date"], select{
  width:100%;padding:10px;border-radius:10px;border:1px solid rgba(15,23,42,0.06);background:#fff;font-size:14px;
 }
 .row{display:flex;gap:8px}
```

```
.row .half{flex:1}
.muted{color:var(--muted);font-size:13px}

.btn{
  background:linear-gradient(90deg,var(--accent),var(--accent-2));
  color:white;border:0;padding:10px 12px;border-radius:10px;font-weight:700;cursor:pointer;
}
.btn.ghost{background:transparent;border:1px solid rgba(15,23,42,0.06);color:var(--muted);font-weight:600}

/* right area */
.main{
  display:flex;flex-direction:column;gap:16px;
}
.card{
  background:var(--panel);border-radius:12px;padding:16px;box-shadow:var(--shadow);
}
.stats{display:flex;gap:14px;flex-wrap:wrap}
.stat{flex:1;min-width:160px;background:linear-gradient(180deg,#fff,#fbfdff);padding:12px;border-radius:10px;text-align:left}
.stat .num{font-weight:800;font-size:20px}
.charts{display:grid;grid-template-columns:1fr 420px;gap:16px}
canvas{max-width:100%;height:auto}

table{width:100%;border-collapse:collapse;font-size:14px}
th,td{padding:8px;text-align:left;border-top:1px solid rgba(15,23,42,0.04)}
thead th{font-size:13px;color:var(--muted);font-weight:600}

.small{font-size:13px;color:var(--muted)}
.tips{display:flex;flex-direction:column;gap:8px;margin-top:8px}
.tip{background:#f7fff6;border-radius:8px;padding:10px;color:#0f5132;font-weight:600}

/* responsive */
@media (max-width:1000px){
      .wrap{grid-template-columns:1fr; padding:12px}
      .charts{grid-template-columns:1fr}
    }
  </style>
</head>
<body>
  <div class="wrap">
    <!-- LEFT: Controls -->
    <div class="panel" id="controls">
      <div class="brand">
        <div class="logo">ET</div>
        <div>
          <h1>EcoTrack</h1>
          <div class="small">Estimate & visualize your carbon footprint</div>
        </div>
      </div>

      <p class="lead">Enter today's activities (values are optional — leave 0 if not applicable). Emissions are estimates.</p>

      <!-- Emission factors reference -->
      <div style="margin-bottom:12px">
        <div class="small" style="font-weight:700;margin-bottom:6px">Emission factors (estimates)</div>
        <div class="small">Car: <b>0.192 kg CO₂e/km</b> · Bus: <b>0.089 kg CO₂e/km</b> · Train: <b>0.041 kg CO₂e/km</b></div>
        <div class="small">Electricity: <b>0.6 kg CO₂e/kWh</b> · Gas: <b>2.3 kg CO₂e/liter</b></div>
        <div class="small">Flight: <b>90 kg CO₂e/hour</b> · Meat meal: <b>2.5 kg CO₂e/meal</b></div>
      </div>

      <div class="field">
        <label for="entry-date">Date</label>
        <input type="date" id="entry-date" />
      </div>

      <div class="field">
      <label>Transport (km)</label>
      <div class="row">
        <input type="number" id="car-km" class="half" placeholder="Car km" min="0" step="0.1" />
        <input type="number" id="bus-km" class="half" placeholder="Bus km" min="0" step="0.1" />
      </div>
      <div style="margin-top:8px">
        <input type="number" id="train-km" placeholder="Train km" min="0" step="0.1" />
      </div>
    </div>

    <div class="field row">
      <div class="half">
        <label for="elec-kwh">Electricity (kWh)</label>
        <input type="number" id="elec-kwh" min="0" step="0.1" />
      </div>
      <div class="half">
        <label for="gas-liters">Gas (L)</label>
        <input type="number" id="gas-liters" min="0" step="0.1" />
      </div>
    </div>

    <div class="field row">
      <div class="half">
        <label for="flight-hrs">Flight (hrs)</label>
        <input type="number" id="flight-hrs" min="0" step="0.1" />
      </div>
      <div class="half">
        <label for="meat-meals">Meat meals (#)</label>
        <input type="number" id="meat-meals" min="0" step="1" />
      </div>
    </div>

    <div style="display:flex;gap:8px;margin-top:8px">
      <button class="btn" id="add-entry">Add entry</button>
      <button class="btn ghost" id="clear-form">Clear</button>
    </div>

    <div style="margin-top:14px;display:flex;gap:8px">
      <button class="btn ghost" id="export-csv">Export CSV</button>
      <button class="btn ghost" id="import-sample">Seed sample</button>
    </div>

    <div style="margin-top:14px" class="small">
      <b>Storage:</b> Data is stored in your browser (LocalStorage). No
server.
    </div>
  </div>

  <!-- RIGHT: Dashboard -->
  <div class="main">
    <div class="card">
      <div style="display:flex;justify-content:space-between;align-
items:center">
        <div>
          <div style="font-size:16px;font-weight:800">Overview</div>
          <div class="small">Daily entries, category breakdown and recent
activity</div>
        </div>
        <div class="muted small" id="last-updated">—</div>
      </div>

      <div style="margin-top:12px" class="stats">
        <div class="stat">
          <div class="small">Total CO₂e (all time)</div>
          <div class="num" id="total-co2">0 kg</div>
          <div class="small muted" id="avg-daily">Avg/day: —</div>
        </div>
        <div class="stat">
          <div class="small">This day's total</div>
          <div class="num" id="today-co2">0 kg</div>
          <div class="small muted">Tip below suggests reductions</div>
        </div>
        <div class="stat">
        <div class="small">Entries</div>
        <div class="num" id="entry-count">0</div>
        <div class="small muted">Edit or delete entries in history</div>
      </div>
    </div>
  </div>

  <div class="card charts">
    <div>
      <div style="display:flex;justify-content:space-between;align-items:center">
        <div style="font-weight:700">Emissions by category</div>
        <div class="small muted">Hover or tap legend to toggle</div>
      </div>
      <canvas id="doughnutChart" style="margin-top:8px;max-height:320px"></canvas>
    </div>

    <div>
      <div style="font-weight:700">Last 30 days</div>
      <canvas id="lineChart" style="margin-top:8px;max-height:220px"></canvas>

      <div class="small muted" style="margin-top:10px">Suggestions</div>
      <div class="tips" id="tips-area" style="margin-top:8px">
        <!-- tips inserted here -->
      </div>
    </div>
  </div>

  <div class="card">
    <div style="display:flex;justify-content:space-between;align-items:center">
      <div style="font-weight:700">Recent entries</div>
      <div class="small muted">Manage history</div>
    </div>
    <div style="margin-top:8px;overflow:auto">
      <table>
```

```
        <thead><tr><th>Date</th><th>Category</th><th>Total kg
  CO₂e</th><th></th></tr></thead>
        <tbody id="entries-tbody"></tbody>
       </table>
      </div>
     </div>

    </div>
   </div>

<script>
/* ----------------- Data model & emission factors ----------------- */
/* Emission factors are simplified estimates. Adjust as needed. */
const EF = {
  car: 0.192,    // kg CO2e per km
  bus: 0.089,
  train: 0.041,
  electricity: 0.6, // kg CO2e per kWh
  gas: 2.3,         // kg CO2e per liter
  flight: 90,       // kg CO2e per flight hour (short/medium avg)
  meatMeal: 2.5     // kg CO2e per meat-based meal
};

const LS_KEY = 'ecotrack-data-v1';
let state = { entries: [] };

/* ---------- Utilities ---------- */
function formatKg(n){ return
   `${Number(n).toLocaleString(undefined,{maximumFractionDigits:1})} kg`;
   }
function todayISO(){ return new Date().toISOString().slice(0,10); }
function uid(){ return 'e_'+Math.random().toString(36).slice(2,9); }

/* load/save */
function load(){
  try{
    const raw = localStorage.getItem(LS_KEY);
    if(raw) state = JSON.parse(raw);
  }catch(e){ console.warn(e); state = {entries:[]}; }
}
function save(){
  localStorage.setItem(LS_KEY, JSON.stringify(state));
}

/* compute CO2 for an entry */
function computeEntry(entry){
  // entry fields: date, carKm, busKm, trainKm, elecKwh, gasL, flightHrs,
   meatMeals
  const by = {};
  by.transport = (entry.carKm||0)*EF.car + (entry.busKm||0)*EF.bus +
   (entry.trainKm||0)*EF.train;
  by.energy = (entry.elecKwh||0)*EF.electricity + (entry.gasL||0)*EF.gas;
  by.flights = (entry.flightHrs||0)*EF.flight;
  by.food = (entry.meatMeals||0)*EF.meatMeal;
  by.other = 0;
  const total = Object.values(by).reduce((s,v)=>s+v,0);
  return {by, total};
}

/* ---------- Rendering & charts ---------- */
let doughnutChart = null;
let lineChart = null;

function renderAll(){
  // aggregates
  const entries = state.entries.slice().sort((a,b)=>
   b.date.localeCompare(a.date));
  const totals = entries.reduce((agg,e)=>{
    const c = computeEntry(e);
    agg.totalAll += c.total;
    agg.by.transport += c.by.transport;
    agg.by.energy += c.by.energy;
    agg.by.flights += c.by.flights;
    agg.by.food += c.by.food;
    agg.by.other += c.by.other;
    return agg;
  }, {totalAll:0, by:{transport:0,energy:0,flights:0,food:0,other:0}});

  document.getElementById('total-co2').textContent =
   formatKg(totals.totalAll);
  document.getElementById('entry-count').textContent = entries.length;
  document.getElementById('last-updated').textContent = entries.length ?
   `Last: ${entries[0].date}` : '—';

  // average per day (over last 30 days if entries present)
  const last30 = getTotalsForLastNDays(30);
  const sum30 = last30.reduce((s,x)=>s+x.amount,0);
  const avg = last30.length ? sum30 / Math.max(1,last30.length) : 0;
  document.getElementById('avg-daily').textContent = `Avg/day:
   ${formatKg(avg)}`;
  // today's total
  const today = entries.find(e => e.date === todayISO());
  document.getElementById('today-co2').textContent = formatKg(today ?
   computeEntry(today).total : 0);

  // doughnut chart (by category)
  const catLabels = ['Transport','Home energy','Flights','Food','Other'];
  const catValues = [
    totals.by.transport,
    totals.by.energy,
    totals.by.flights,
    totals.by.food,
    totals.by.other
  ];
  drawDoughnut(catLabels, catValues);

  // line chart (last 30 days)
  drawLine(last30);

  // render entries list (first 12)
  renderEntriesTable(entries.slice(0,12));

  // tips
  renderTips(totals.by);
```

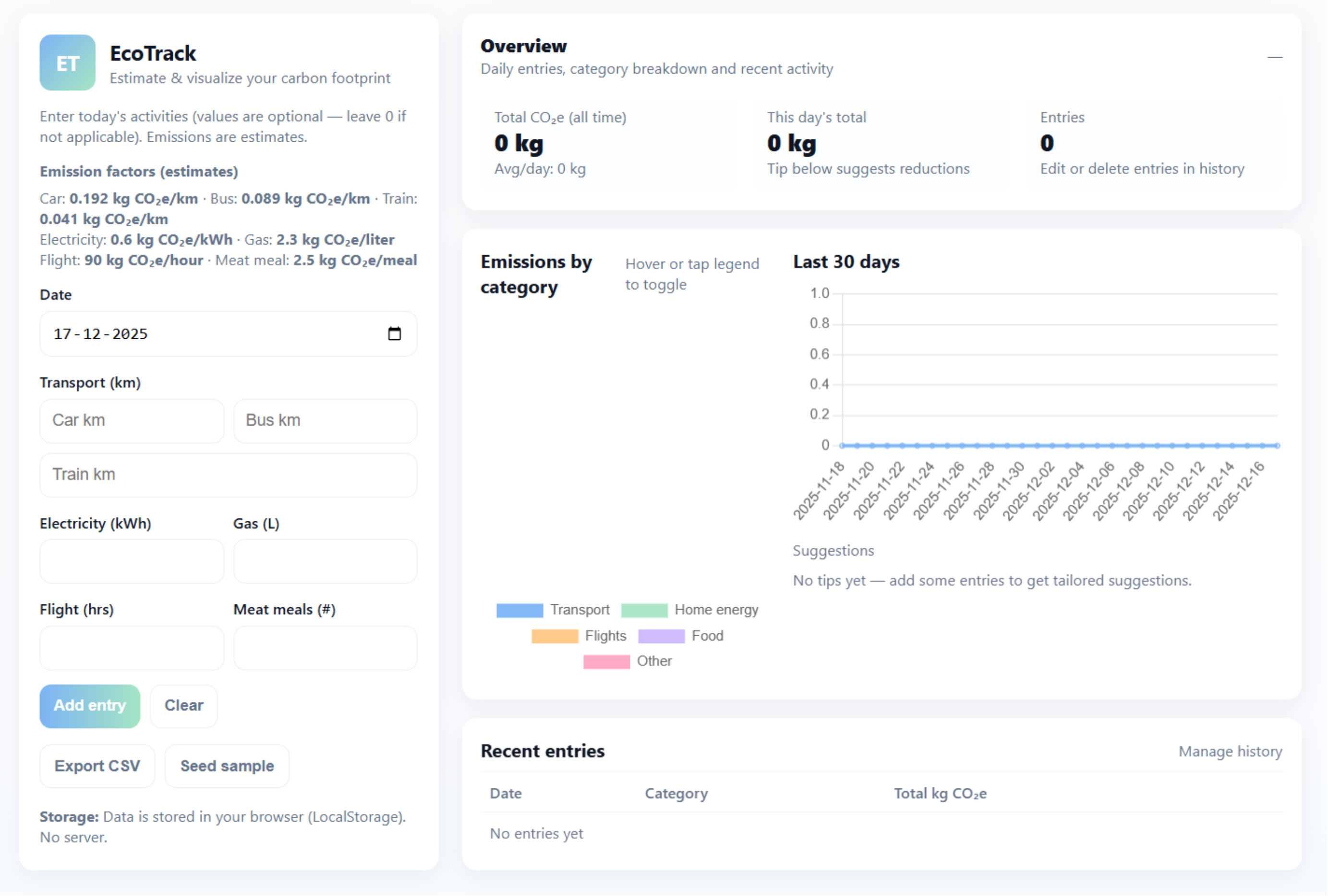
```
}

/* get totals per day for last N days (array of {date, amount}) */
function getTotalsForLastNDays(n){
  const today = new Date();
  const arr = [];
  for(let i = n-1; i>=0; i--){
    const d = new Date(); d.setDate(today.getDate()-i);
    arr.push({date:d.toISOString().slice(0,10), amount: 0});
  }
  // sum amounts from entries
  state.entries.forEach(e=>{
    const comp = computeEntry(e).total;
    const item = arr.find(a=>a.date===e.date);
    if(item) item.amount += comp;
  });
  return arr;
}

/* draw doughnut */
function drawDoughnut(labels, data){
  const ctx = document.getElementById('doughnutChart').getContext('2d');
  const bg = generateColors(labels.length);
  if(doughnutChart) {
    doughnutChart.data.labels = labels;
    doughnutChart.data.datasets[0].data = data;
    doughnutChart.data.datasets[0].backgroundColor = bg;
    doughnutChart.update();
    return;
  }
  doughnutChart = new Chart(ctx, {
    type:'doughnut',
    data:{labels, datasets:[{data, backgroundColor:bg, borderWidth:0}]},
    options:{
      plugins:{legend:{position:'bottom'}},
      maintainAspectRatio:false
    }
  });
```

```
}

/* draw line */
function drawLine(dataArr){
  const ctx = document.getElementById('lineChart').getContext('2d');
  const labels = dataArr.map(d=>d.date);
  const values = dataArr.map(d=>Number(d.amount.toFixed(2)));
  if(lineChart){
    lineChart.data.labels = labels;
    lineChart.data.datasets[0].data = values;
    lineChart.update();
    return;
  }
  lineChart = new Chart(ctx, {
    type:'line',
    data:{
      labels,
      datasets:[{
        label:'Daily CO₂e (kg)',
        data: values,
        tension:0.25,
        fill:true,
        backgroundColor:'rgba(124,180,245,0.12)',
        borderColor:'rgba(124,180,245,0.95)',
        pointRadius:2
      }]
    },
    options:{
      plugins:{legend:{display:false}},
      scales:{x:{grid:{display:false}}, y:{beginAtZero:true}}
    }
  });
}

/* render recent entries table */
function renderEntriesTable(entries){
  const tbody = document.getElementById('entries-tbody');
  tbody.innerHTML = '';
  if(entries.length===0){
    tbody.innerHTML = '<tr><td colspan="4" class="small muted">No entries
   yet</td></tr>';
    return;
  }
  entries.forEach(e=>{
    const comp = computeEntry(e);
    const tr = document.createElement('tr');
    tr.innerHTML = `<td>${e.date}</td>
                    <td class="small">${categorySummary(e)}</td>
                    <td>${formatKg(comp.total)}</td>
                    <td style="text-align:right">
                      <button class="btn ghost" data-id="${e.id}"
   onclick="editEntry('${e.id}')">Edit</button>
                      <button class="btn ghost" data-id="${e.id}"
   onclick="deleteEntry('${e.id}')">Delete</button>
                    </td>`;
    tbody.appendChild(tr);
  });
}

/* category summary small text */
function categorySummary(e){
  const parts = [];
  if(e.carKm) parts.push(`Car ${e.carKm}km`);
  if(e.busKm) parts.push(`Bus ${e.busKm}km`);
  if(e.trainKm) parts.push(`Train ${e.trainKm}km`);
  if(e.elecKwh) parts.push(`Elec ${e.elecKwh}kWh`);
  if(e.gasL) parts.push(`Gas ${e.gasL}L`);
  if(e.flightHrs) parts.push(`Flight ${e.flightHrs}h`);
  if(e.meatMeals) parts.push(`${e.meatMeals} meat`);
  return parts.join(' · ') || 'No data';
}

/* render tips based on major categories */
function renderTips(by){
  const tipsArea = document.getElementById('tips-area');
  tipsArea.innerHTML = '';
  // create array of categories ordered by value desc
  const arr = [
    {k:'transport', v:by.transport, tip:'Consider carpooling, using public transport, or biking for short trips.'},
    {k:'energy', v:by.energy, tip:'Lower thermostat, switch to LED bulbs, and reduce idle appliance use.'},
    {k:'flights', v:by.flights, tip:'Reduce short-haul flights or choose lower-emission routing where possible.'},
    {k:'food', v:by.food, tip:'Try one or two plant-based meals per week; it reduces food-related emissions.'},
    {k:'other', v:by.other, tip:'Check for waste reduction and choose low-impact products.'}
  ];
  arr.sort((a,b)=>b.v-a.v);
  arr.forEach(it=>{
    if(it.v > 0){
      const div = document.createElement('div');
      div.className = 'tip';
      div.textContent = it.tip;
      tipsArea.appendChild(div);
    }
  });
  if(tipsArea.children.length===0){
    tipsArea.innerHTML = '<div class="small muted">No tips yet — add some entries to get tailored suggestions.</div>';
  }
}

/* generate pastel colors */
function generateColors(n){
  const palette = [
    'rgba(124,180,245,0.95)',
    'rgba(166,230,195,0.95)',
    'rgba(255,199,132,0.95)',
    'rgba(204,183,255,0.95)',
    'rgba(255,166,196,0.95)',
    'rgba(200,200,200,0.95)'
  ];
  return Array.from({length:n},(_,i)=>palette[i % palette.length]);
}

/* ---------- CRUD for entries ---------- */
function addEntryFromForm(){
  const date = document.getElementById('entry-date').value || todayISO();
  const e = {
    id: uid(),
    date,
    carKm: Number(document.getElementById('car-km').value) || 0,
    busKm: Number(document.getElementById('bus-km').value) || 0,
    trainKm: Number(document.getElementById('train-km').value) || 0,
    elecKwh: Number(document.getElementById('elec-kwh').value) || 0,
    gasL: Number(document.getElementById('gas-liters').value) || 0,
    flightHrs: Number(document.getElementById('flight-hrs').value) || 0,
    meatMeals: Number(document.getElementById('meat-meals').value) || 0
  };
  state.entries.push(e);
  save();
  renderAll();
  clearForm();
}

/* edit entry (simple prompt-based edit for demo) */
function editEntry(id){
  const entry = state.entries.find(x=>x.id===id);
  if(!entry) return alert('Entry not found');
  // populate form and remove original, then user will click Add
  document.getElementById('entry-date').value = entry.date;
  document.getElementById('car-km').value = entry.carKm;
  document.getElementById('bus-km').value = entry.busKm;
  document.getElementById('train-km').value = entry.trainKm;
  document.getElementById('elec-kwh').value = entry.elecKwh;
  document.getElementById('gas-liters').value = entry.gasL;
  document.getElementById('flight-hrs').value = entry.flightHrs;
  document.getElementById('meat-meals').value = entry.meatMeals;
  // remove the old entry (so Add will reinsert it)
  state.entries = state.entries.filter(x=>x.id!==id);
  save();
  renderAll();
  window.scrollTo({top:0,behavior:'smooth'});
}

function deleteEntry(id){
  if(!confirm('Delete this entry?')) return;
  state.entries = state.entries.filter(x=>x.id!==id);
  save();
  renderAll();
}

/* clears form */
function clearForm(){
  document.getElementById('entry-date').value = todayISO();
  ['car-km','bus-km','train-km','elec-kwh','gas-liters','flight-hrs','meat-
   meals'].forEach(id=>document.getElementById(id).value='');
}

/* export CSV */
function exportCSV(){
  if(!state.entries.length) return alert('No entries to export');
  const rows =
   [['id','date','car_km','bus_km','train_km','elec_kwh','gas_l','flight_hrs','meat_
   meals','total_kg']];
  state.entries.forEach(e=>{
    const comp = computeEntry(e).total;
    rows.push([e.id,e.date,e.carKm,e.busKm,e.trainKm,e.elecKwh,e.gasL,e.flig
   htHrs,e.meatMeals,comp.toFixed(2)]);
  });
  const csv =
   rows.map(r=>r.map(c=>`"${String(c).replace(/"/g,'""')}"`).join(',')).join('\n');
  const blob = new Blob([csv], {type:'text/csv'});
  const url = URL.createObjectURL(blob);
  const a = document.createElement('a');
  a.href = url; a.download = 'ecotrack_entries.csv'; a.click();
  URL.revokeObjectURL(url);
}
```

```
/* seed sample data */
function seedSample(){
  const now = new Date();
  for(let i=0;i<8;i++){
    const d = new Date(now); d.setDate(now.getDate()-i);
    state.entries.push({
      id: uid(),
      date: d.toISOString().slice(0,10),
      carKm: Math.round(Math.random()*40),
      busKm: Math.round(Math.random()*10),
      trainKm: Math.round(Math.random()*20),
      elecKwh: Math.round(Math.random()*12),
      gasL: Math.round(Math.random()*2),
      flightHrs: Math.random()<0.15 ? Math.round(Math.random()*4)+1 : 0,
      meatMeals: Math.round(Math.random()*2)
    });
  }
  save();
  renderAll();
}

/* ---------- Init and event wiring ---------- */
document.addEventListener('DOMContentLoaded', ()=>{
  load();
  if(!document.getElementById('entry-date').value)
   document.getElementById('entry-date').value = todayISO();
  document.getElementById('add-entry').addEventListener('click',
   addEntryFromForm);
  document.getElementById('clear-form').addEventListener('click',
   clearForm);
  document.getElementById('export-csv').addEventListener('click',
   exportCSV);
  document.getElementById('import-sample').addEventListener('click',
   seedSample);

  renderAll();
});
```

```
/* expose edit/delete to global scope for inline handlers */
window.editEntry = editEntry;
window.deleteEntry = deleteEntry;

</script>
</body>
</html>
```

OUTPUT



EcoTrack

Estimate & visualize your carbon footprint

Enter today's activities (values are optional — leave 0 if not applicable). Emissions are estimates.

Emission factors (estimates)

Car: **0.192 kg CO₂e/km** · Bus: **0.089 kg CO₂e/km** · Train: **0.041 kg CO₂e/km**
Electricity: **0.6 kg CO₂e/kWh** · Gas: **2.3 kg CO₂e/liter**
Flight: **90 kg CO₂e/hour** · Meat meal: **2.5 kg CO₂e/meal**

Date: 17 - 12 - 2025

Storage: Data is stored in your browser (LocalStorage). No server.

Overview

Daily entries, category breakdown and recent activity

Total CO₂e (all time): **0 kg** — Avg/day: 0 kg

This day's total: **0 kg** — Tip below suggests reductions

Entries: **0** — Edit or delete entries in history

Emissions by category — Hover or tap legend to toggle

Last 30 days

Suggestions

No tips yet — add some entries to get tailored suggestions.

Recent entries — Manage history

Date	Category	Total kg CO₂e
No entries yet		

CHAPTER 4 - RESULTS AND DISCUSSIONS

The result of the ECOTRACK application is the calculated personal carbon footprint of the user, displayed in both numerical and graphical formats. After entering the required data, users receive an estimate of their total carbon emissions. This output provides a clear overview of their environmental impact.

The graphical representation of results is one of the most important outcomes of the project. Charts such as bar graphs or pie charts visually display the contribution of different activities to the total carbon footprint. This helps users easily identify high-emission areas in their lifestyle.

The output is generated instantly using JavaScript, ensuring quick response without page reloads. The results are displayed in a structured and visually appealing format, enhancing readability and user understanding. Overall, the output of ECOTRACK effectively meets the project's objective of increasing awareness and promoting sustainable behavior.

CHAPTER 5 - CONCLUSION AND FUTURE SCOPE

5.1 Conclusions

ECOTRACK successfully demonstrates the application of front-end web development technologies in addressing a real-world environmental problem. The project combines sustainability concepts with modern web design to create an informative and interactive carbon footprint calculator.

Through simple input forms, accurate calculations, and visual charts, ECOTRACK helps users understand their personal contribution to carbon emissions. The project highlights the importance of awareness and individual responsibility in combating climate change.

From an academic perspective, ECOTRACK provides valuable hands-on experience in HTML, CSS, JavaScript, and data visualization. Overall, the project proves that technology can play a vital role in promoting sustainability and environmental responsibility.

5.2 FUTURE SCOPE

The ECOTRACK project has significant potential for future enhancements. One major improvement would be the integration of a backend system to store user data securely. This would allow users to track their carbon footprint over time and analyze trends.

Another possible enhancement is the inclusion of additional emission categories such as waste management, water usage, and consumer goods. Integration with external APIs can provide real-time data and improve calculation accuracy.

Future versions can also include personalized recommendations using artificial intelligence or machine learning techniques. A mobile application version of ECOTRACK can further increase accessibility and user engagement. These enhancements can transform ECOTRACK into a comprehensive sustainability platform.