CARBON FOOTPRINT TRACKER

Empowering you to reduce your environmental impact

Debleena Shree

2341621
4 BCA B

CONTENT

1. INTRODUCTION

The Carbon Footprint Tracker is a web application designed to help individuals and organizations measure, track, and reduce their carbon emissions. By providing a user-friendly interface for inputting energy usage, transportation, waste, food, and shopping data, the app calculates the user's carbon footprint and offers actionable recommendations to reduce it. The solution aims to promote environmental awareness and encourage sustainable practices.

Purpose

- Environmental Awareness: Educate users about their carbon emissions and environmental impact.
- Actionable Insights: Provide personalized recommendations to reduce carbon footprints.
- Progress Tracking: Allow users to monitor their carbon footprint over time.
- Community Engagement: Foster a sense of community through a leaderboard and carbon offset projects.

2. TECHNICAL IMPLEMENTATION

- Frontend:
    - HTML, CSS, JavaScript: Core technologies for building the user interface.
    - Chart.js: This is used to visualize carbon footprint data in charts.
    - Font Awesome: For icons and visual enhancements.
- Backend:
    - Node.js: Runtime environment for server-side logic.
    - Express.js: Framework for building RESTful APIs.
    - JSON File: Temporary storage for user data (for demonstration purposes).

- Styling:
    - Custom CSS: For responsive and modern design.
    - Google Fonts (Poppins): For typography.
    - Dark Mode: Custom implementation for better user experience.

Architecture

The application follows a client-server architecture:

1. Frontend:
    - Static files (HTML, CSS, JS) served to the client.
    - Interactive forms for data input and visualization.
2. Backend:
    - REST API built with Node.js and Express.js.
    - Handles user authentication, data storage, and carbon footprint calculations.
3. Data Storage:
    - JSON file is used to store user data (temporary solution).
    - Future integration with a database (e.g., MongoDB).

Development Approach

- Modular Design: The application is divided into reusable components (e.g., forms, charts, buttons).
- Progressive Enhancement: Basic functionality works for all users, with advanced features (e.g., dark mode) added for modern browsers.
- Agile Methodology: Iterative development with frequent testing and feedback.

3. FEATURES & FUNCTIONALITY

Key Features

1. Carbon Footprint Calculator:
   - Users input energy usage, transportation, waste, food, and shopping data.
   - The app calculates the carbon footprint using predefined formulas.
2. Personalized Recommendations:
   - Provides actionable suggestions to reduce carbon emissions based on user input.
3. Progress Tracking:
   - Displays a line chart of the user's carbon footprint history.
   - Allows users to track their progress over time.
4. Leaderboard:
   - A community leaderboard to compare carbon footprint reductions with others.
5. Carbon Offset Projects:
   - Information about real-world projects to offset carbon emissions.
   - Links to donate or get involved.
6. Dark Mode:
   - Toggleable dark mode for better user experience in low-light environments.

4. PROJECT/DEMO

Archive File: Carbon Footprint Tracker Zip File

Github: Carbon Footprint Tracker

Demo Video: Carbon Footprint Tracker Demo Video

5. SCALABILITY

- Database Integration: Replace the JSON file with a scalable database like MongoDB.
- Cloud Deployment: Deploy the app on cloud platforms like AWS, Heroku, or Vercel for better scalability.

6. FUTURE SCOPE

1. User Authentication:
    - Implement a secure authentication system using JWT or OAuth.
2. Responsive Design:
    - Optimize the app for mobile devices and tablets.
3. Gamification:
    - Add badges, rewards, or achievements to reduce carbon footprints.
4. API Integration:
    - Integrate with external APIs for real-time carbon footprint data (e.g., energy usage APIs).
5. Multi-Language Support:
    - Add support for multiple languages to reach a global audience.

7. CHALLENGES FACED & SOLUTIONS

1. Data Duplication in Footprint History

- Problem: The footprintHistory array was storing duplicate entries due to incorrect data handling.
- Solution: Ensured that the footprint value was converted to a number using parseFloat() before saving it to the array.

2. Dark Mode Not Working

- Problem: The dark mode toggle was not applying styles correctly.
- Solution: Added proper CSS specificity and ensured the dark-mode class was toggled on the <body> element.

3. Large Images in Carbon Offset Projects

- Problem: Images in the carbon offset projects section were too large and distorted the layout.
- Solution: Added CSS to control image size (width: 100%, height: 200px, object-fit: cover).

4. User Authentication

- Problem: The app lacked a proper authentication system.
- Solution: Implemented a basic login/signup system using localStorage. Future plans include integrating JWT or OAuth.

5. Responsive Design Issues

- Problem: The app was not fully responsive on smaller screens.
- Solution: Added media queries to adjust the layout for mobile devices.

8. CONCLUSION

The Carbon Footprint Tracker is a robust and user-friendly application that empowers users to measure, track, and reduce their carbon emissions. The app provides actionable insights and fosters environmental awareness by combining a clean, intuitive interface with powerful backend logic. With future enhancements like database integration, user authentication, and API integration, the app has the potential to scale and reach a global audience.

---

9. CONTACT

For questions or feedback, feel free to reach out:

- Email: shreedebleena@gmail.com
- GitHub: Debleena17
