IEEE & WIE

WEB DEVELOPMENT HACKATHON PROBLEM STATEMENT - 1

Project Idea: Carbon footprint tracker.

Problem Statement:

You need to design and create an interactive web platform that allows users to track their carbon footprint and actively engage in reducing their environmental impact. The platform should also raise awareness, educate, and encourage users to make informed decisions and take measurable steps towards a greener future.

Aim

- The aim is to provide a user-friendly interface to calculate individual carbon footprints based on various activities and lifestyle choices.
- The platform should give personalized recommendations and a comprehensive achievement system.

Note: This project statement presents a comprehensive challenge that extends beyond the scope of a mere three-day timeframe. Our immediate goal is to develop a foundational website that embodies essential features. However, the complete problem statement caters to those enthusiastic about web development, offering an avenue to craft a distinctive project while delving into advanced concepts. If you're excited about learning new things and taking on a bigger challenge, the full version of the problem statement is designed for those who are really passionate about web development and want to create something unique and advanced.

Note: To ensure accessibility and ease of use for users, it is mandatory to deploy the project on cloud-hosting websites such as Github.io or Netlify. This will enable our users to use the platform from any device with an internet connection.

List of features (Must have):

- **User Authentication**: Users can register and log in to the platform. They can create a profile, track their carbon footprint, and view their progress over time.
- **Carbon Footprint Calculator**: Users can use a carbon footprint calculator to estimate their carbon emissions based on factors such as their energy usage, transportation habits, and lifestyle choices.

Note: For calculation part of calculator , categories must be defined that a participant wants to include (some major are transportation, energy consumption,and waste management).Then they have to determine the inputs required, for example for energy consumption distance traveled, vehicle type, and fuel efficiency are some major inputs and then comes the calculation of emission factors.For calculation part majorly it would be multiplying the inputs by the corresponding emission factors to calculate the carbon emissions and then summing up the emissions from each category to obtain the total carbon footprint.

- **Achievement and Recognition System:** To reward volunteers for their dedicated service. This system can include virtual badges, certificates, or a points-based system that recognizes volunteers' achievements, milestones, and positive contributions. It fosters a sense of accomplishment and encourages continued engagement.

Additional Requirements

These are not mandatory requirements but nice to have

- **Carbon Footprint Comparison :**

  Ø Enable users to compare their carbon footprints with friends, colleagues, or averages within their region or demographic.

- **Goal Setting and Tracking :**

  Ø Allow users to set personalized carbon reduction goals and track their progress over time.

  Ø Provide reminders and notifications to help users stay on track and achieve their goals.

Tech Requirements

· Readme.md should contain all the details of the project.

  Ø Steps to set up the project.

· Mention a checklist of implemented features and additional features in the readme file

- All the code and documentation should be pushed to GitHub.
- Deploy the final application to any Cloud-hosting website.
- Write automated test cases to verify all functional requirements.

What to submit?

- GitHub repository link of the project.
- Hosting link (Heroku, Netlify, etc) for the application.
- User ID and Password for authentication.

What will be Evaluated?

- Feature list/Requirements met (listed above)
- User Interface and Design.
- Logic & method to implement the calculator.
- Overall complexity of the project.

Steps to build the project(Decreasing order of priority):

1. Define Requirements:

   - Determine the specific features and functionalities of the platform.

   - Identify the target audience and their needs.

   - Set goals for the platform's performance and usability.

2. Design the User Interface:

   - Create wireframes and mockups to visualize the platform's layout and structure.

- Ensure the interface is intuitive, user-friendly, and responsive across different devices.

3. Set up the Development Environment:

- Choose a programming language, framework, and database for the backend development.

- Select appropriate tools and technologies for the frontend development.

- Set up a development environment with the necessary software and dependencies.

4. Implement User Authentication Component:

- Develop a user registration system that allows users to create accounts.

- Implement user login functionality to authenticate users.

- Ensure proper security measures are in place, such as password hashing and encryption.

5. Develop Carbon-Footprint Calculator:

- Design and implement an algorithm to calculate the carbon footprint based on user inputs.

- Create a user-friendly interface for entering relevant data, such as transportation habits, energy consumption, and lifestyle choices.

- Calculate and display the carbon footprint to the user upon submission of the data.

6. Build Achievement and Recognition System:

- Define milestones or achievements based on carbon footprint reduction goals.

- Track and store user progress and achievements.

- Develop a system to provide recognition to users when they reach specific milestones or make significant progress in reducing their carbon footprint.

7. Provide Personalized Recommendations:

   - Utilize the user's carbon footprint data to generate personalized recommendations for reducing their environmental impact.

   - Display actionable suggestions related to transportation, energy usage, waste management, and lifestyle changes.

   - Present the recommendations in a clear and engaging manner to encourage user adoption.

8. Test and Debug:

   - Perform thorough testing of the platform, including functional testing, usability testing, and security testing.

   - Identify and fix any bugs or issues that arise during the testing phase.

   - Optimize the performance and responsiveness of the platform.

9. Deploy and Maintain:

   - Choose a hosting provider and deploy the web platform.

   - Set up necessary server configurations and ensure proper security measures are in place.

   - Monitor the platform's performance, handle user feedback, and continuously improve the system based on user needs and emerging technologies.

The choice of tech stack and 3rd party APIs can vary based on preferences, development expertise, and specific requirements. Here's a suggested tech stack and a few examples of relevant 3rd party APIs that can be utilized:

Documentation:

Create clear and comprehensive documentation for your project. Include instructions for setting up the development environment, running the application, and any specific configuration required. Write a detailed README file explaining the project, its features, and how to use it.

Tech Stack:

These are just some examples of technologies you can use for the project. You may consider using additional libraries, frameworks, or APIs based on your specific requirements and preferences.

- Frontend: HTML5, CSS3, JavaScript (React, Next)

- Backend: Node.js,Express.js,Nest.js

- Database: SQL,MongoDB

- Hosts: Netlify or Vercel for frontend, Cyclic or Render for backend

Complexity	Medium
Project Duration	3 Days.