PROJECT TITLE: **Trash to Treasure: Upcycle & Reuse Hub**

PROJECT DESCRIPTION:
A web/ mobile platform that promotes waste management by encouraging DIY recycling projects. Users can find tutorials and upcycling ideas. Target SDGs: 11, Sustainable Cities and Communities: Encouraging responsible waste management and urban sustainability; 13, Climate Action: Reducing waste pollution and promoting eco-friendly initiatives. By integrating technology and sustainability, Trash to Treasure helps reduce landfill waste, raises awareness about upcycling, and promotes creative solutions for a greener future. The platform enables users to:

- **Browse Upcycling Tutorials** – Users can explore DIY guides for repurposing common waste materials like plastic bottles, old clothes, or discarded furniture. They can also upload their own upcycling projects and share step-by-step instructions with the community.

- **Recycling & Disposal Guide** – A searchable database that provides information on local recycling centers, donation spots, and proper waste disposal guidelines based on location. This feature helps users find the best ways to discard waste responsibly.

- **Learn About Waste Management & Environmental Sustainability** – Users can access educational resources on the importance of waste management, climate change, and sustainable living. Interactive articles, infographics, and videos will highlight how small actions contribute to a greener planet. The platform will also feature expert insights and success stories to inspire positive change.

TARGET USERS:
**DIY enthusiasts, environmental advocates, students, and local communities**

**Technical Implementation Requirements:**

- The code should use React JS language
- Implementation of the following ***features***:
  - Utilize ReactJS to build the web application.
  - Follow best practices in React development, including component-based architecture, state management, and reusable components.
  - Utilize React Router for managing client-side routing.
  - Implement at least one external API for fetching data.

- Ensure responsive design for various screen sizes using CSS frameworks like Bootstrap or Material-UI.
- Implement state management using either React Context API or Redux.
- Use modern JavaScript features and ES6 syntax throughout the project.
- Employ best practices for code organization and maintainability.

**Functional Requirements:**

- Implement features outlined in the project scope.
- Ensure the web application functions as intended and meets the specified requirements.
- Handle user interactions, form submissions, and data management effectively.
- Implement user authentication and authorization (if applicable).
- Include CRUD (Create, Read, Update, Delete) functionality for managing data.
- Ensure error handling and validation for user inputs.
- Utilize asynchronous operations for data fetching and manipulation.
- Incorporate reusable components and avoid code duplication