**Green coding best practices checklist**

✔ Optimize code for better performance  
✔ Reduce memory and CPU usage  
✔ Use energy-efficient algorithms

✔ Minimize HTTP requests and API calls  
✔ Enable caching mechanisms  
✔ Reduce image and asset sizes

✔ Choose green hosting providers  
✔ Optimize cloud resource usage  
✔ Turn off unused servers and services

✔ Reduce redundant data storage  
✔ Optimize database queries  
✔ Use efficient data structures

✔ Write clean and maintainable code  
✔ Implement lazy loading where applicable  
✔ Reduce dependencies and bloat

✔ Measure energy consumption of applications  
✔ Use profiling tools to detect inefficiencies  
✔ Continuously refactor and optimize code

**📌 Features of the Interactive Checklist**

1. **User Authentication** – Allow users to log in and track progress.
2. **Progress Tracking** – Display completion percentage.
3. **Save State** – Store checked items locally or in a database.
4. **Responsive UI** – Works across devices (desktop & mobile).
5. **Gamification** – Rewards for completing tasks (e.g., badges).
6. **Community Sharing** – Allow developers to share their progress.

**🛠️ Implementation Steps**

**1️⃣ Frontend (User Interface)**

* Use **React/Vue.js** for an intuitive UI.
* Include **checkboxes** for each best practice.
* Display a **progress bar** as users check off items.

**2️⃣ Backend (Data Persistence)**

* **User authentication** via Firebase, OAuth, or NextAuth.
* Store progress in a **database** (MongoDB, PostgreSQL, or Firebase).

**3️⃣ UI Components**

* **Checklist Section** (Each best practice with a checkbox)
* **Progress Bar** (Percentage of tasks completed)
* **Save Button** (Store progress)
* **Reset Button** (Clear progress)

**🚀 Example Interactive Flow**

1. **User logs in**
2. **Sees the checklist** with unchecked boxes
3. **Toggles checkboxes** to mark tasks as done
4. **Progress bar updates** dynamically
5. **Progress is saved** in local storage or database
6. **User earns badges** upon full completion