GITHUB INSTRUCTIONS

📂 Project Structure

This project is organized as a **monorepo**, which means all our code (frontend and backend) lives in this single repository. This makes collaboration much easier.

- `backend/`
  - This is where **Tejaswini & Mayank** work.
  - It contains the **Node.js / Express.js** server, all API logic, and the database connection.
- `frontend/`
  - This is where **Abhay** works.
  - It contains the **React Native** mobile application, including all screens, components, and AR logic.
- `docs/`
  - This folder holds all our planning and documentation files, like the SRD, API contract, and these instructions.

🌿 Branching Strategy

Our workflow is based on **feature branches**. This keeps our `main` branch clean and working at all times.

The Golden Rule: Never commit directly to the `main` branch.

1. The `main` Branch

The `main` branch is our official, stable version of the project. Code only gets added to `main` after it has been reviewed and approved in a Pull Request.

2. Feature Branches (Your Workspace)

For every new task, you must create your own branch from the latest version of `main`.

Our Naming Convention:
`your-name/short-feature-description`

Branch Examples for Each Team Member:

- **Abhay (Frontend):**
  - `abhay/create-login-screen`

    - `abhay/implement-ar-placing-logic`
    - `abhay/build-greenscore-profile-page`
- **Tejaswini (Backend):**
    - `tejaswini/setup-user-auth-api`
    - `tejaswini/write-waste-prediction-script`
    - `tejaswini/build-product-endpoints`
- **Mayank (Backend):**
    - `mayank/donation-workflow-api`
    - `mayank/build-manager-web-dashboard`
    - `mayank/charity-verification-endpoint`

The Process

1. **Create your branch** from `main`.
2. **Do your work** and make commits on your branch.
3. When finished, **push your branch** to GitHub and **open a Pull Request** to merge it back into `main`.
4. Wait for a teammate to review and approve your Pull Request before merging.