Software Web Development Basics - Assignment 1 - 11/9/2025

SolarEye - App Idea Description

SolarEye is a simple web app that manages solar site inspections. Registered users create sites, log inspections, and record findings (solar panel faults) in the report. The report summarizes inspection results (including fault types, severity of the fault, photo of the solar panel, and any notes made).

App Functionalities

- **Users:** The user can create sites and view inspections if available
- **Sites:** Create and view solar sites (name, capacity installed, map location)
- **Inspections:** Schedule / complete inspections per site; add notes and status
- **Report:** Record solar panel issues found during a maintenance inspection and summarize it with the title, type of fault, severity, photo, and the creation of the report that is made per inspection.

Model, Properties & Relations

1. **User**

- Property: `id`, `name`, `email`, `password_hash`, `created_at`
- Relations:
  - 1→M — hasMany(`Site`)
  - 1→M — hasMany(`Inspection`)
- Comment:
  - A user can own many sites and do many inspections

2. **Site**

- Property: `id`, `user_id`, `name`, `capacity`, `latitude`, `longitude`, `created_at`

- Relations:
  - 1→M — hasMany(`Inspection`)
  - belongsTo(`User`)
- Comment:
  - A site belongs to one user and has many inspections

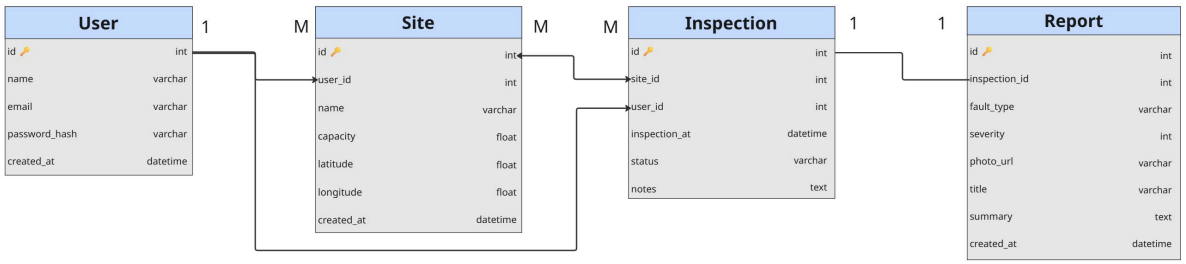
3. **Inspection**

- Property: `id` , `site_id` , `user_id` , `inspected_at` , `status` , `notes`
- Relations:
  - M→N — belongsTo(`Site`)
  - 1→M — belongsTo(`User`)
  - 1→1 — hasOne(`Report`)
- Comment:
  - An inspection is for one site, one user, and can have one report

4. **Report**

- Property: `id` , `inspection_id` , `title` , `summary` , `fault_type` , `severity` , `photo_url` , `created_at`
- Relations:
  - 1→1 — belongsTo(`Inspection`)
- Comment:
  - One report is made per inspection

Model Relationship Structure



SolarEye Web App Relationship Structure