

1 Project Overview

- **Name:** The Sunflower Post
- **Stack:** Next.js 16 (App Router), React, TypeScript, Tailwind CSS
- **Goal:** A soft, non-performative community platform with “rooms” for:
 - emotional check-ins
 - hope stories
 - music / nostalgia
 - books, TV, movies
 - visual inspo
 - gentle solution rooms

The UX is inspired by **Circle / Geneva**: left sidebar with rooms, main content area for each room, consistent layout, gentle colours, small typography.

2 Current File / Route Structure (high level)

Exact paths may vary slightly, but this is the intended structure we’ve been working towards:

```
src/  
  app/  
    page.tsx                // Homepage (marketing/landing)  
    lounge/  
      page.tsx              // Lounge room main feed  
      [id]/  
        page.tsx            // Single Lounge post thread view  
    hope-bank/  
      page.tsx              // Hope Bank main feed  
      [id]/  
        page.tsx            // Single Hope story view  
    music-room/  
      page.tsx              // Music Room (early / partial)
```

```

book-club/
  page.tsx // Book Club main: list of books
  [id]/
    page.tsx // Single book discussion hub
    topic/
      [topicId]/
        page.tsx // Single topic thread inside that
book
  components/
    CommunitySidebar.tsx // Left sidebar with rooms
    LoungeRoom.tsx // Lounge main UI
    LoungePostPage.tsx // Single Lounge post view
    HopeBankRoom.tsx // Hope Bank main UI
    HopeStoryPage.tsx // Single Hope story view
    MusicRoom.tsx? // (if already created)
    BookClubPage.tsx // Book list view
    BookDiscussionPage.tsx // View for a single book (topics)
    BookTopicPage.tsx // Single topic thread for a book
  lib/
    search.ts // (planned) helpers e.g.
matchesSearch

```

You (Claude) should assume a standard App Router + Tailwind setup.

3 What's Already Implemented (Core Behaviour)

◆ CommunitySidebar

- **File:** `src/components/CommunitySidebar.tsx`
- **Status:** Implemented and used across Lounge, Hope Bank, Book Club, etc.

Items currently shown:

- 💬 The Lounge – Daily check-ins & small joys
- 🌈 Hope Bank – Stories of things quietly getting better
- 🎵 Music Room – Playlists & feel-good songs
- 📖 Book Club – Coming soon / in progress
- 📺 TV & Movies – Coming soon

 Pinterest Wall – Coming soon

 Solution Rooms – Coming soon

-
- Some items may still be “Coming soon” in text, but the intention is they all eventually route to their own pages.

LoungeRoom (Main feed)

File: `src/components/LoungeRoom.tsx`

Route: `src/app/lounge/page.tsx`

What it does now:

- Shows a **header** with:
 - Room tag (“Room”)
 - Title: “The Lounge”
 - Description about small joys, pick-me-ups, and soft rants
 - **Search bar** (searches title + body + author)
- Supports 3 post types via `type: "joy" | "pickmeup" | "softrant"`:
 - Small joy
 - Pick-me-up
 - Soft rant
- Has filters: `"Today" | "This week" | "All time" | "Needs love"` (UI only so far; filter logic can be expanded later).
- Has a **Pick-me-up submission form**:
 - Title (optional)
 - Body
 - Optional name + “Post as anonymous” checkbox

- Gentle boundaries text
 - On submit:
 - Adds a new `pickmeup` post to in-memory `posts` state at the top.
 - Each post card shows:
 - Badge for type (Small joy / Pick-me-up / Soft rant)
 - Time ago
 - Title + body
 - Author
 - Link to comments: `href="/lounge/[id]"` with `Link`
 - **Reactions per post (per-viewer only, no global counts):**
 - 🌻 Send warmth
 - 💙 Gentle support
 - ❤️ Here with you
 - Stored via `reactions: Record<number, UserReactions>` in state.
 - Copy explicitly says: *"Reactions are for care, not counts. Only you see what you've sent."*
-

💬 LoungePost thread view

Route: `src/app/lounge/[id]/page.tsx`

Component: `LoungePostPage` (name assumed; we created a thread view)

- Shows one Lounge post + its replies.
- Uses the same **CommunitySidebar** layout (sidebar left, content right).
- Has a reply form at the bottom.
- Placeholder/mock data wired for now.

HopeBankRoom (Main feed)

File: `src/components/HopeBankRoom.tsx`

Route: `src/app/hope-bank/page.tsx`

What it does now:

- Uses `CommunitySidebar` in a 4-column grid (1 col sidebar / 3 cols main).
- Header section:
 - “Room” tag
 - Title: “Hope Bank”
 - Description: real stories about things quietly getting better
 - **Search bar** under the description

Stories defined in a local `STORIES` array:

```
type HopeStory = {
  id: number;
  title: string;
  summary: string;
  category: "Career" | "Health" | "Family" | "Money" | "Faith" |
  "Other";
  turningPoint: string;
  author: string;
  isAnon: boolean;
  timeAgo: string;
  saves: number;
};
```

-
- Category filters: "All" | "Career" | "Health" | "Family" | "Money" | "Faith" | "Other".
- Search filters across: `title + summary + turningPoint`.

- Each story card shows:
 - Category pill
 - Time ago
 - Title
 - Summary
 - “Turning point” italic line
 - Author (or “Anon”)
 - Save button with saves count
 - “View full story” [Link](#) → `/hope-bank/[id]`
 - **Reactions per story (per-viewer, like Lounge):**
 - 🌻 Send warmth
 - 💙 Gentle support
 - 🧡 Here with you
 - Right sidebar explains:
 - What is Hope Bank?
 - How to share a story (steps)
 - Gentle boundaries
-

HopeBank single story view

Route: `src/app/hope-bank/[id]/page.tsx`


Component: `HopeStoryPage` (or similar)

- Shows full story.
- Has a layout consistent with:

- CommunitySidebar on the left,
- story content + comments on the right.
- Allows comments and (optionally) reactions — some of this might still be mock/placeholder.

MusicRoom

Status: Concept + partial UI / file, but **not fully implemented**.

- We discussed:
 - Song posts with:
 - title, artist, mood/emotion tags
 - optional image upload (album cover)
 - Reactions such as:
 - 🔥 This bangs
 - ✨ Nostalgia hit
 - 🎧 Listening now
 - ❤️ Saved to playlist
 - Ability to avoid duplicates by checking a local DB before adding.
 - Optional “watchlist” / “save for later”.
- Sidebar already includes “ Music Room”.

If a `MusicRoom.tsx` exists, assume it's early/placeholder.

Book Club

Main book list:

- `src/app/book-club/page.tsx` uses `BookClubPage` (or equivalent).
- Shows a list of books (mock data):
 - Title, author, tags, mood, maybe small description.
 - Each book links to `/book-club/[id]`.

Single book discussion page:

- `src/app/book-club/[id]/page.tsx` → `BookDiscussionPage`.
- Shows:
 - Book info (title, author, cover placeholder)
 - Intro text
 - List of topics/discussions for that book (mock)
 - Button to “Start a new topic”
- Topics are things like:
 - “Chapter 3 – what does love actually look like?”
 - “How did you feel reading the ending?”
- Each topic links to: `/book-club/[id]/topic/[topicId]`.

Topic thread page:

- `src/app/book-club/[id]/topic/[topicId]/page.tsx`
`BookTopicPage` (we just wrote this).

What `BookTopicPage` does:

- Reads `params.id` and `params.topicId` and passes them as numbers to `BookTopicPage`.
- Currently uses **mock data** `MOCK_TOPIC` with:
 - `bookTitle`, `author`

- topicTitle
- startedBy, timeAgo
- body
- replies[]
- Displays:
 - Breadcrumb: Book Club / [Book Title] / Topic: [Topic Title]
 - Starter post
 - List of replies
 - Simple reactions on replies (static for now):
 - 🌻 Resonated
 - 📖 Highlight
 - 👁 Following
 - Reply textarea + “Post reply” button (no backend yet).

Dynamic route snafu we fixed:

- We made sure `[id]` and `[topicId]` are **consistently named** in App Router. Previously there was a mismatch (`[bookId]` vs `[id]`), which we corrected.

4 What’s Left To Do (Big Picture)

This is where Claude can really help you pick up the build.

A. Data & State

- Currently using **in-memory mock arrays** (Lounge posts, Hope stories, books, topics).
 - ➡️ SOON Decide whether to:
 - keep it as mock data for MVP, or

- integrate with a DB (e.g. Prisma + SQLite/Postgres, or Supabase).
 - Implement real **create/read/update** flows:
 - Create post/story/topic
 - Persist replies
 - Persist reactions per user (or per session)
 - Prevent duplicate songs/books if desired
-

B. Music Room (needs implementation)

- Build `MusicRoom` component + `src/app/music-room/page.tsx`:
 - Header explainer
 - Search bar
 - Add song form:
 - Title (string)
 - Artist (string)
 - Mood/emotion (select + free-text option)
 - Optional link (YouTube/Spotify)
 - Optional image upload (cover)
 - Song list cards:
 - Show mood tags, artist, title
 - Support reactions:
 - 🔥 This bangs
 - ✨ Nostalgia hit
 - 🎧 Listening now

- ❤️ Saved to playlist

- Comment thread for each song (own dynamic page `music-room/[id]` — optional but consistent with pattern).
-

C. Book Club (further work)

- Hook `BookClubPage` → `BookDiscussionPage` → `BookTopicPage` with shared data shapes.
 - Allow adding new books:
 - Title, author, optional cover upload
 - Optional description, themes, mood tags
 - Allow adding new topics per book:
 - Topic title
 - Body, tags: “chapter”, “theme”, “spoilers?”
 - Add **spoiler handling**:
 - Maybe a `spoiler: boolean` flag on replies or topics.
 - UI to hide/show spoiler content.
 - Implement reactions as real toggles per user, not just static buttons.
-

D. TV & Movies Room

- Create `src/app/tv-movies/page.tsx` + `TvMoviesRoom` component.
- Use similar pattern to Book Club & Music Room:
 - Add series/films (title, platform, genre, feelings)
 - Reactions: 🍿, 😭, 😂, ❤️, 🔄

- Threads for specific episodes or themes (optional: dynamic routes).
-

E. Pinterest Wall (Visual inspo)

- Create `src/app/pinterest-wall/page.tsx` + component.
 - Allow image uploads with:
 - Caption
 - Tags (“Soft life”, “Career era”, “Travel” etc.)
 - Allow users to “save” items to their own profile (requires user system later).
-

F. Solution Rooms

- Create `src/app/solution-rooms/page.tsx`.
 - Pattern:
 - People post challenges & context.
 - Others reply with frameworks, advice, resources.
 - Maybe tag posts by themes: Career, Burnout, Study, Money, etc.
-

G. Search + Lib helpers

- There’s a planned helper: `matchesSearch` in `@/lib/search` (imported in one file).
- Implement `matchesSearch` properly:
 - Accepts `searchTerm: string` and `haystack: string` or object fields.
 - Normalises and returns `boolean`.
- Optionally unify search logic across rooms using this helper.

H. Reactions (global behaviour pattern)

Right now, reactions are:

- **Per-room state only**
- Not persisted
- Only visible to the current viewer

For MVP this is fine, but Claude can:

- Extract a reusable `useReactions` hook or component.
- Allow different reaction sets per room (Lounge vs Music vs Book topics).
- Keep the “**no public counts**” principle.

I. Auth, Profiles, and “My Stuff” (future)

We’ve *talked about*:

- Member profile:
 - Saved items (songs, posts, images)
 - Sun points & badges (gamification)
 - Contributions (topics, replies, stories)

This is not coded yet.

5 How to Brief Claude (“start here”)

You can say something like:

I already have a Next.js 16 + Tailwind project called Sunflower Post.
We’ve implemented:

- `LoungeRoom` and its thread view, with reactions and a pick-me-up form
- `HopeBankRoom` and its story cards with reactions
- A shared `CommunitySidebar` that appears on all room pages
- Early structure for `BookClub`, including `book-club/[id]` and `book-club/[id]/topic/[topicId]`
I want you to:

1. Review the existing components and keep the visual + tone consistent.
2. Finish the Music Room as a full page (songs list, add-song form, reactions, comments).
3. Then help wire Book Club properly (book list → single book → topic threads with replies and reactions).
4. Use in-memory mock data first, but write code in a way that I can later plug in a database.
5. Keep reactions private (no global counts), and maintain the soft, gentle, non-performative tone.