High level goals (what “gamey / nature / tactile” means in practice)

1. Make the UI feel **alive** and responsive like a small ecosystem — things react with soft springs, gentle bounces, layered parallax, and tactile drag.

2. Use **visual metaphors** of trees / branches / leaves for information architecture and actions (e.g., branches = project sections, leaves = posts/memories).

3. Add **microinteractions** for key user actions (create post, like, invite), but respect accessibility (reduced motion).

4. Keep performance and progressive enhancement in mind (animations are additive, not required to use the app).

What to change in the repo (priority + where to implement)

1) Theme & design tokens — Tailwind config

Priority: High — affects the whole UI and makes future changes easy.

What to do:

- Add a nature-themed token set (greens, earthy neutrals, accent florals), rounded radii, and tactile utility tokens (soft shadows, grain texture variables).

- Add component tokens for: tree-bg, leaf, branch, seed, bark — so CSS classes read like the concept.

Example tailwind.config.ts additions (extend the theme):

```
// tailwind.config.ts (adds tokens — merge into your existing file)
export default {
```

```
  // ...existing config
  theme: {
    extend: {
      colors: {
        'leaf-100': '#F1FBF3',
        'leaf-300': '#C6EAD0',
        'leaf-500': '#6EC26B',
        'bark-400': '#6B4630',
        'bark-200': '#9B7A66',
        'sky-100': '#F5FAFF',
        'flower-400': '#F7C6D2',
      },
      borderRadius: {
        'soft': '1.25rem', // tactile pill shapes
        'leaf': '0.75rem',
      },
      boxShadow: {
        'soft-lg': '0 10px 30px rgba(17,24,39,0.12)',
        'leaf-press': 'inset 0 -4px 6px rgba(0,0,0,0.06)',
      },
    }
  }
}
```

Why: single source of truth for “nature” tokens; easier to style components across the app.

---

2) Create a small design system / UI primitives folder

Priority: High

Add src/components/ui/game/* (or src/components/ui/nature/*) with primitives:

- TreeCanvas (layout + background layers)
- BranchCard (holds a branch)
- Leaf (represents a post/memory)
- SeedButton (primary CTA with growth animation)

Why: consistent behavior and animation patterns; reuse across pages (dashboard, branches, onboarding).

---

3) Framer Motion patterns & component examples

Priority: High — you said you already switched to Framer Motion. Use it to create tactile gestures.

Key patterns to adopt:

- **Staggered entrance** for branches/leaves (gives the “sprouting” feel). Use AnimatePresence and variants.
- **Hover + press micro-springs**: whileHover, whileTap on leaves and CTAs. Use spring physics for tactile feeling.
- **Drag for rearranging** branches/leaves: add drag, dragConstraints, dragElastic for playful re-ordering.
- **Parallax background** with useMotionValue and useTransform for depth.

Example

Leaf

component (drop into

src/components/leaves/Leaf.tsx

)

```
import { motion, Variants } from "framer-motion";

const leafVariants: Variants = {
  hidden: { opacity: 0, y: 20, scale: 0.95 },
  visible: (i = 1) => ({ opacity: 1, y: 0, scale: 1, transition: { delay: i * 0.06, type: 'spring', stiffness: 220, damping: 18 } }),
  hover: { y: -6, rotate: -3, scale: 1.03, transition: { type: 'spring', stiffness: 400, damping: 28 } },
  tap: { scale: 0.98, rotate: -1, transition: { type: 'spring', stiffness: 700, damping: 30 } }
};
```

```
export function Leaf({ index=0, children, onOpen }) {
  return (
    <motion.div
      className="bg-leaf-100 rounded-leaf p-4 shadow-soft-lg cursor-pointer select-none"
      custom={index}
      variants={leafVariants}
      initial="hidden"
      animate="visible"
      whileHover="hover"
      whileTap="tap"
      onClick={onOpen}
      drag
      dragConstraints={{ left:0, right:0, top:0, bottom:0 }}
      dragElastic={0.15}
      role="button"
      aria-label="Open memory"
    >
      {children}
    </motion.div>
  )
}
```

Why: small code block gives a tactile, springy feel and uses drag and hover gestures. Framer Motion docs show these gesture props and spring tuning options.

---

4) Background / layout: layered nature scene

Priority: Medium — visual anchor

Ideas:

- Use a tiled SVG background that draws a distant forest silhouette + subtle grain overlay.
- Build a TreeCanvas that composes layers: sky → distant canopy → mid-branches → foreground leaves. Move layers slightly with scroll or pointer for parallax.
- Keep the content area within a rounded “tree trunk” card for major flows (login, dashboard).

Implementation hints:

- Keep SVGs small and inline where possible.
- Use pointermove -> useMotionValue -> useTransform to produce subtle layer movement.

---

5) UX Flows turned into playful interactions

Priority: High — improves perceived product-game feel

Map critical flows into game metaphors:

- **Onboarding**: “Plant your first seed” — user chooses a tree name, then plant animation grows into a sapling (progress = branches unlocked).
- **Create memory / leaf**: show a seed that users drag into a branch to “plant” the memory — animation sprouts a leaf.
- **Notifications**: Instead of toast, show a gentle falling leaf with a short bounce on arrival.
- **Permissions/roles**: visualize as tree caretakers (owner = oak, admin = tall branch, member = leaf) — small badges/icons.

---

6) Accessibility & performance guardrails (required)

Priority: Critical

- Respect prefers-reduced-motion: turn off non-essential motion. Use both CSS and JS checks. Examples and guidance: MDN and W3C recommend using prefers-reduced-motion and testing with it.

Example (JS + Framer Motion):

```
import { useReducedMotion } from "framer-motion";

const shouldReduce = useReducedMotion(); // framer-motion helper
// If shouldReduce === true, use simple fades or no animation.
```

- Add a global toggle to “Reduce motion” + “Sound on/off”.
- Debounce heavy animation updates and prefer will-change on animated elements. Use small subtrees for complex motions so React diffing stays cheap.

---

7) Performance & testing

Priority: Medium

- Avoid animating large bitmap images — animate transforms on composited layers (translate, scale, opacity).
- Use next/image for media and lazy-load offscreen leaves.
- Add visual regression tests for key scenes (jest + storybook snapshots or Chromatic) to avoid animation regressions.

---

8) Assets & illustration approach

Priority: Medium

- Use stylized, flat vector assets (SVG) so they can be animated and recolored dynamically.
- Keep a small set of tokenized leaf illustrations (different shapes) to vary posts but keep file-size low.
- Consider Lottie for more complex animated sequences (seed sprout animation), but only for “one-off” sequences (onboarding).

---

9) Suggested file/component changes in your repo

(Concrete places to add code)

- src/components/ui/ — add TreeCanvas.tsx, SeedButton.tsx, Leaf.tsx, BranchCard.tsx.
- src/components/dashboard/ — replace static lists with AnimatePresence + mapped Leaf components. Add small stagger.
- src/app/onboarding/ — create the “plant seed” onboarding screen using SeedButton.
- tailwind.config.ts — extend tokens as above.

UX rules & guardrails (keep these in PR descriptions)

- Microinteractions must be < 300ms for small feedback; entrance sequences can be longer but keep total time < 900ms. (Tune with Framer Motion visualDuration or springs.)
- Respect prefers-reduced-motion by default. Add toggle for users who want motion.

Examples of great small interactions you can emulate

- Drag a seed onto a branch to plant — gives ownership (fun, direct manipulation).
- On hover, a leaf shivers slightly and a tooltip blooms with read-time / timestamp.
- “Harvest”/archive permission: user drags the leaf to a “basket” to archive it (satisfying animation).

These microinteractions are recommended UX patterns and validated by UX literature to increase engagement while communicating feedback.

\

Perfect — let's build you a **Figma component spec** that captures the "game-like, nature-inspired, tactile" design system for **Tribe App**. You (or a designer) can drop these directly into Figma as components and tokens.

🌱 Tribe App Component Spec

1.

Color Palette (Nature-Inspired Tokens)

Token	Hex	Usage
leaf-100	#F1FBF3	Lightest background
leaf-300	#C6EAD0	Secondary surfaces
leaf-500	#6EC26B	Primary brand green
leaf-700	#3F7F3A	Hover/pressed states
bark-200	#9B7A66	Secondary text, dividers
bark-400	#6B4630	Primary text, headers
sky-100	#F5FAFF	Page background
sky-300	#BFE3F8	Accent, info highlights
flower-400	#F7C6D2	Success/positive highlights
fruit-400	#FFD66B	CTA accent

👉 **In Figma:** create a **Color Styles** set named Tribe Colors. Prefix tokens: Leaf/500, Bark/400, etc.

2.

Typography

- **Headings (H1–H4):** Rounded, friendly, “gamey” sans-serif (e.g., **Baloo 2**, **Nunito**, or **Fredoka**).
- **Body:** Neutral geometric sans (e.g., **Inter**, **Nunito Sans**).
- **Details:** Use **italic** bark colors for “whispers” (helper text, tooltips).

👉 **Figma Text Styles:**

- H1: 32px, SemiBold, Line 120%, Color: Bark/400
- H2: 24px, SemiBold, Line 130%, Color: Bark/400
- Body: 16px, Regular, Line 150%, Color: Bark/200
- Caption: 12px, Medium, Line 140%, Color: Bark/200

3.

Shapes & Radii

- **Soft, rounded corners** to feel tactile.
- Tokens:
    - radius-soft: 20px
    - radius-leaf: 12px
    - radius-pill: 9999px (full pill)

👉 Apply to **buttons, cards, and input fields**.

4.

Elevation & Shadows

- **Leaf Press (inset)**: inset 0 -4px 6px rgba(0,0,0,0.06)
- **Soft Elevation**: 0 6px 12px rgba(0,0,0,0.08)
- **Floating Card**: 0 10px 30px rgba(0,0,0,0.12)

👉 **Figma Effects Styles:** Shadow/Leaf, Shadow/Soft, Shadow/Floating.

5.

Key Components

🌿

Leaf (Card)

- Container for posts/memories.
- **Default State:** Leaf/100 bg, soft-lg shadow, radius-leaf.
- **Hover State:** Lift + shadow expand, slight rotation.
- **Press State:** Inset shadow (Shadow/Leaf Press).
- **Variants:** Leaf/Note, Leaf/Image, Leaf/Task.

🌳

Branch (Section Container)

- Holds groups of leaves.
- **Default State:** Rounded rectangle, Bark/200 border, padded interior.
- **Interaction:** Expand/collapse like a branch unfurling.

🌱

Seed Button (Primary CTA)

- Shape: pill, Fruit/400 or Leaf/500 background.
- Hover: slight scale up + soft glow.
- Tap: compress inward with inset shadow.
- Icon: small sprout / seed.

🍃

Tree Canvas (Layout Background)

- Layers: Sky/100 background, faint canopy silhouettes (SVGs).
- Parallax effect for branches + leaves.

6.

Motion & Interaction Tokens

- **Entrance:** “Sprout” — scale from 0.95 → 1, spring bounce (stiffness 220, damping 18).
- **Hover:** “Shiver” — rotate ±2°, lift 6px, duration 200ms.
- **Tap/Press:** “Press In” — scale 0.98, inset shadow.
- **Exit:** “Fall” — fade out, y+20, ease-in.

👉 Document in Figma **Variants** → **Prototype** using Smart Animate.

7.

Iconography

- Style: Rounded, playful, Duolingo-like.
- Nature metaphors instead of generic icons:
    - Add = Seed
    - Delete = Falling leaf
    - Settings = Gear + branch motif
    - Notifications = Bird chirp icon
