what at the correct prompts to use in order to create a set of concentric rings to use in a personal finance mobile application built on the lovable platform that will have each ring independently change colors based upon a number value

**ChatGPT said:**

To create **concentric rings** in a personal finance mobile application built using the **Lovable** platform, where each ring can **independently change colors based on a number value**, you'll need to use a combination of **custom UI components**, **dynamic styling**, and **state-driven logic**.

Here’s a complete **prompt structure** to use on Lovable (or similar AI-based no-code platforms) that will generate your desired UI logic:

**✅ Lovable Prompt: Concentric Rings with Dynamic Colors**

**Prompt:**

"Create a mobile screen titled Financial Health Rings that displays **3 concentric circular progress rings** stacked on top of each other. Each ring should represent a different metric:

* Outer ring: Spending Discipline
* Middle ring: Income Stability
* Inner ring: Savings Progress

Each ring should:

* Be sized relative to its position (outer: largest, inner: smallest)
* Display a numeric value in the center (0–100)
* Change color based on the value:
  + **0–49**: Red
  + **50–74**: Yellow
  + **75–100**: Green

The values for each metric will come from a JSON object:

json

CopyEdit

{

"spending\_discipline": 65,

"income\_stability": 85,

"savings\_progress": 40

}

Use state to dynamically render the color and percentage of each ring based on these values. Each ring should animate on load."

**🔧 Implementation Details (What Lovable should generate)**

1. **Three circular progress components** using layered views (SVG or Canvas-based rendering)
2. **Logic for dynamic color assignment** using conditional styling:

typescript

CopyEdit

function getRingColor(value: number): string {

if (value < 50) return "red";

if (value < 75) return "yellow";

return "green";

}

1. **Props/State mapping**:
   * JSON object values passed into component state
   * Ring color and progress value derived from state
2. **Reusable Ring Component**:
   * Pass value, label, radius, and strokeWidth as props
   * Internally calculate strokeDasharray for animation and color

**🧠 Tip for Prompt Variants**

To explore variants:

* Prompt: "Add tooltips or labels to each ring that briefly explain what the value means in personal finance terms."
* Prompt: "Create tap gestures so that tapping a ring takes the user to a detailed breakdown screen."
* Prompt: "Include ring labels that update when values change in real time from user financial behavior."

**📂 File Schema (Optional)**

If you're saving user data, request:

"Create a data model to store ring values under the user profile, with timestamped entries for each metric. Use the schema:

json

CopyEdit

{

user\_id: string,

metrics: {

spending\_discipline: number,

income\_stability: number,

savings\_progress: number,

timestamp: datetime

}

}

```"

---

Would you like me to generate the JSON config and UI wireframe mockup for this Lovable screen?

4o