Ok, here is a comprehensive product spec and feature set. There is a lot there. Please have a look and suggest how we might turn this into sprints.

The spec you wrote is rich enough to treat this like a real product backlog. A good way to turn it into sprints is to group by **capability layer** and keep each sprint shippable and small.[1] [2]

Suggested sprint themes

Sprint 1 – Visual & motion foundations

Goal: lock in the **look and movement** of the galaxy before adding editing and data.[2]

- Implement design palette:
  - Text and UI colors, central sun `#dedede`, node color cycle you specified.
- Update orbit visuals:
  - Light gray rings instead of bright green.
- Orbit speed control:
  - Top-bar or panel slider with range 0.1–1.5; 1.0 as default.
- Clean up instructions text to match current behavior.

Outcome: the current minimal app but styled per your principles, with a working speed slider.

Sprint 2 – Modes: Solar vs Static

Goal: introduce **view modes** and basic node repositioning.[2]

- Add mode toggle in top-right:
  - “Solar system” (default) vs “Static”.
- Behavior:
  - Solar: nodes orbit as now, obey speed slider.
  - Static: freeze current positions; no orbit updates.
- Dragging in Static:
  - Click–drag planets along their ring (constrained radial distance).
- “Reposition camera” control:
  - Button to reset camera to $0, 20, 50$ without resetting node positions.

Outcome: two modes you can use in workshops: dynamic overview vs hand-placed configurations.

Sprint 3 – Node editing (Central + planets)

Goal: make nodes **editable objects** instead of just visuals.[2]

- Node detail panel (left/top panel reuse of Welcome surface):
  - Click central node or planet → open panel.
  - Fields: title, description text area, color picker, icon picker.
  - Save button persists changes in memory.
- Hover behavior:
  - On hover, show read-only title + description in the same panel area (no inputs).
- Astronaut "focus line":
  - 1px light gray line from explorer to last-clicked node.

Outcome: you can meaningfully author a system with titles, descriptions, and icons.

Sprint 4 – Additional node types: Moons & Satellites

Goal: introduce the full **hierarchy** while keeping the interaction simple.[2]

- Add "Add object" button + modal (top-right):
  - Choose type: Planet / Moon / Satellite.
  - For Moon: chooser listing existing planets.
  - For Satellite: chooser listing existing moons.
  - Fields: title, description, color, icon.
- Layout and motion:
  - Planets: new outer ring each time; ring adjustable later.
  - Moons: 1–2 radial distances around parent planet, with invisible orbit radius.
  - Satellites: single radial distance around parent moon, invisible orbit.
- Solar vs Static behavior:
  - Solar: moons orbit planets; satellites orbit moons.
  - Static: children stay where placed within allowed radius.

Outcome: a fully hierarchical galaxy that still obeys your two-mode model.

Sprint 5 – Layout editing: rings & orbits

Goal: give facilitators **control over system geometry**.[2]

- Orbit ring adjustments (from default camera view):
  - Per-ring radial distance slider(s).

    - Vertical offset controls per ring.
    - Tilt controls per ring.
- Persist these adjustments in the in-memory model.

Outcome: the atlas can be shaped to match conceptual relationships, not just default orbits.

Sprint 6 – Persistence & export

Goal: preserve work and enable sharing.[2]

- In-memory model extension:
    - Store system title, node hierarchy, positions (Solar and Static), icon, color, orbit geometry.
- Local persistence:
    - Save to `localStorage`.
    - Optional import/export JSON.
- PNG export:
    - "Export view" button:
        - Hide UI overlays.
        - Render scene with system title and node labels.
        - Download as PNG.

Outcome: a usable, self-contained tool someone can adopt for real sessions.

How to phrase sprint items for us

For each sprint, you can send something like:

- "Sprint 1 / Task 1: apply palette and orbit ring color exactly as in spec; here is any tweak."
- "Sprint 2 / Task 2: implement static mode with drag-along-ring only; no vertical or tilt yet."

Keeping each request scoped to **one sprint theme** and **2–5 concrete tasks** will keep the code and conversation clear.

⁂

1. sozo-product-spec.rtf
2. sozo-feature-set.rtf
3. stretch-roulette.txt
4. normalize-silence-prod.txt
5. README.md
6. script.js
7. index.html

8. styles.css