

Rule-Based Image / Meme Generator (Framer Website)

Project Overview

I will build a rule-based image/meme generator for a Framer website.

The system will generate images instantly based on user selections.

There will be no AI generation, no randomness, and no text on images.

Only the assets provided by the client will be used.

The output will always be consistent:

Same input will always produce the same image.

How I Will Build the Project

The project will be built as a deterministic configurator, not a generator.

- All visuals will be controlled by predefined rules
- Only valid combinations will be allowed
- Visual style will remain flat and consistent
- Maximum of three colours per image
- Rendering will be instant and predictable

The focus is reliability, clarity, and long-term maintainability.

Technology Stack

- React (functional components)
 - SVG rendering (preferred for clean and scalable visuals)
 - Framer React Code Component
 - No external runtime dependencies
-

4-Day Development Plan

Day 1 – Planning & Setup

- Review and organize all provided assets
- Define visual and logical rules

- Create the rule-mapping structure
- Plan SVG layer structure
- Prepare Framer component structure

Output:

Clear architecture and finalized rules

Day 2 – Rule Engine & Rendering

- Build the rule engine
- Implement SVG rendering logic
- Enforce visual constraints (colours, flat style, asset usage)
- Prevent invalid combinations at logic level

Output:

Working image rendering with consistent results

Day 3 – Framer Integration

- Convert the component into a Framer-ready code component
- Expose user options using Framer property controls
- Ensure instant updates on selection change
- Test inside Framer canvas

Output:

Fully usable Framer component

Day 4 – Testing & Final Delivery

- Test all rule combinations
- Fix edge cases
- Clean and document the code
- Apply final feedback

Output:

Production-ready project with clean source code

Requirements From Client

To proceed smoothly, I will need:

1. Final asset files (SVG or PNG)
2. Allowed and restricted combinations (if any)
3. Approved colour palette
4. Export requirement (download image or display only)
5. Any future expansion plans (optional)

Project Folder Structure

/Rule-based-image-generator

```
|
|
|— components
|   |— Generator.tsx
|   |— Controls.tsx
|
|— engine
|   |— rules.ts
|   |— validator.ts
|
|— renderer
|   |— SvgRenderer.tsx
|
|— assets
|   |— shapes
|   |— icons
|   |— backgrounds
```

```
|
|
|— utils
|  └─ constants.ts
|
|
└─ index.ts
```

GitHub Profile

You can review my code and previous work here:

GitHub:

<https://github.com/sksaxena6688>

Ownership & Rights

- Full source code ownership will be transferred to the client
 - No third-party licensed logic will be used
 - The project will be built specifically for this requirement
-

Final Note

This project is clearly defined and well-suited for a rule-based approach.
I am ready to start immediately and deliver within the proposed timeline.