

# Nano Banna Integration Help

can you plug nano banna into claude code?

---

## RouteLLM

I need to check my memory first before responding. I'm not able to integrate or "plug" the nano\_banana image generation model into Claude's code execution environment or any external code editor.

However, I can help you in a few ways:

1. **Generate images using nano\_banana** - I can generate images for you using the nano\_banana model directly through my image generation tool
2. **Provide API integration code** - If you want to use nano\_banana through an API in your own code, I can help you write code that calls image generation APIs (though you'd need the appropriate API credentials)
3. **Generate images for your project** - I can generate images with nano\_banana that you can then download and use in your code projects

What are you trying to accomplish? Are you looking to generate images with nano\_banana, or integrate image generation into a coding project you're working on?

---

well not you specifically but is it possible?

---

## RouteLLM

Yes, it's absolutely possible to plug the "nano banana" image model into Claude Code (and Claude Desktop / other MCP clients). People are already doing it via MCP (Model Context Protocol) servers.

## How it works at a high level

You don't "install nano banana inside Claude" directly. Instead you:

1. Run or reference an **MCP server** that talks to Google's Gemini image API (the nano-banana model).
2. Register that MCP server in your Claude config (claude\_desktop\_config.json or Claude Code MCP settings).
3. Claude then sees new tools like Image Generation / gemini\_Image Generation, and can call them when you ask it to "generate an image".

There are a few ready-made options:

## 1. Nano Banana MCP by ConechoAI

Repository: [Nano-Banana-MCP](#)

This is a packaged MCP server specifically for nano-banana that works with Claude Code, Cursor, etc. Setup for Claude looks like:

```
{
  "mcpServers": {
    "nano-banana": {
      "command": "npx",
      "args": ["nano-banana-mcp"],
      "env": {
        "GEMINI_API_KEY": "your-gemini-api-key-here"
      }
    }
  }
}
```

Once configured, you can ask Claude Code things like “Generate an image of X” and it will route through that MCP tool [source: Nano-Banana-MCP README](#).

## 2. nanobanana-mcp

Another implementation: [nanobanana-mcp](#), a Gemini Vision & Image Generation MCP for Claude Desktop and Claude Code.

Example Claude Desktop config from the repo:

```
{
  "mcpServers": {
    "nanobanana-mcp": {
      "command": "node",
      "args": ["/absolute/path/to/nanobanana-mcp/dist/index.js"],
      "env": {
        "GOOGLE_AI_API_KEY": "your_api_key_here"
      }
    }
  }
}
```

After that, tools like `gemini_image_generation` and `gemini_edit_image` become available inside Claude Code [source: nanobanana-mcp README](#).

### 3. nano-banana Claude Skill / MCP server

There's also a "nano-banana" skill entry with detailed MCP instructions, including a one-liner:

```
claude mcp add nano-banana --env GEMINI_API_KEY=your-key-here -- npx -y nanobanana-mcp
```

And manual config like:

```
{
  "mcpServers": {
    "nano-banana": {
      "command": "npx",
      "args": ["-y", "nanobanana-mcp"],
      "env": {
        "GEMINI_API_KEY": "your-api-key-here"
      }
    }
  }
}
```

[source: nano-banana skill docs](#)

### So, net answer

- **Is it possible?** Yes.
- **How?** Via an MCP server (e.g., Nano-Banana-MCP / nanobanana-mcp) configured in Claude's MCP settings, using your Gemini API key.
- **Result:** Claude Code can generate and edit images through the nano-banana (Gemini image) model as a first-class tool.

If you tell me which environment you're on (Claude Desktop vs Claude Code CLI vs Cursor), I can give you an exact config snippet tailored to that setup.

---

**which mcp has the most development for nano banana?**

---

## RouteLLM

Short version: based on current repos, **ConechoAI's Nano-Banana-MCP is the most developed/general-purpose MCP for nano banana**, with YCSE's **nanobanana-mcp** a close second focused on richer Gemini vision flows.

## 1. ConechoAI / Nano-Banana-MCP

Repo: [ConechoAI/Nano-Banana-MCP](#)

Why it looks like the “most developed”:

- **Higher adoption:** ~40+ stars, 10+ forks vs a handful for others.
- **Feature set:**
  - Image Generation, `edit_image`, `continue_editing`
  - `get_last_image_info`
  - `configure_gemini_token`, `get_configuration_status`
  - Handles cross-platform paths and automatic image saving [source](#).
- **Good UX around config:** clear priority order for API key (MCP env, system env, local config), “configure” tool built in.
- Designed explicitly as “a Nano Banana MCP server, which you can integrate to Cursor/Claude Code and any MCP client” [source](#).

For most people who just want “nano banana in Claude/Cursor,” this is the one I'd start with.

## 2. YCSE / nanobanana-mcp

Repo: [YCSE/nanobanana-mcp](#)

Strengths:

- Tight integration with **Claude Desktop / Claude Code** configs out of the box [source](#).
- Tools like:
  - `gemini_Image Generation`
  - `gemini_edit_image`
  - `gemini_chat`
  - `get_image_history`
  - `clear_conversation`
- More emphasis on **session history**, “**history:N**” references, and **Google Search grounding** for images [source](#).

This is great if you want more advanced workflows (image history, chat + vision, search grounding).

### 3. Others (mcp-image, nano-banana skill, etc.)

- The “**nano-banana**” skill on Claude plugins is mostly a wrapper around an MCP like nanobanana-mcp, with good docs and one-liner installation [source](#).
  - **mcp-image** from the DEV article is more of a tutorial/reference implementation, not as “productized” as the two above [source](#).
- 

If you want a recommendation:

- **General use, easiest to live with:** Nano-Banana-MCP (ConechoAI).
- **Power-user / multimodal workflows (history, chat, search grounding):** nanobanana-mcp (YCSE).

If you tell me whether you're on Claude Desktop vs Claude Code vs Cursor, I can give you a copy-paste config using whichever of these you prefer.

---

**alright, and what are some good prompts to use to generate logo's, icons, ect.. for a google chrome extension i'm working on? i want to create them with nano unless there are better mcp servers / tools out there to do this? this is for the glippy glean clipper chrome extension i've been working on.**

---

### RouteLLM

I will perform a web search to gather current options for MCP/image tools for logo/icon generation, then share concrete prompt ideas tailored to your "glippy glean clipper" Chrome extension. You're on the right track using nano banana / Nano Banana Pro for logos and icons, especially because it's very strong at **text rendering and graphic design-style prompts** [Nano Banana Pro overview](#) and [Gemini image generation docs](#).

I'll break this into:

1. Good models / tools for logo & icon work (MCP + model choice)
  2. Prompt patterns and concrete examples for **Glippy Glean Clipper** (Chrome extension)
  3. How to iterate to a polished icon set
-

# 1. Are there better tools than nano banana for this?

For **Claude/MCP workflows**, nano banana (Gemini image) is actually one of the best options right now for:

- **Legible text & typography in the image** (logo text, UI-ish icons)
- **Flat / vector-like graphics** (the docs explicitly show minimal logo prompts)
- **Brand systems & mockups** (e.g. “make 8 minimalistic logos...” example)

From Google DeepMind’s own Nano Banana Pro examples:

- “Make 8 minimalistic logos, each is an expressive word...”
- “A vibrant, eye-catching ‘TYPOGRAPHY’ design...”
- Multi-logo/vector sheets on a white background [DeepMind Nano Banana Pro page](#).

Given that, for your use case (Chrome extension logo + icons), **Nano Banana Pro via your nano-banana MCP is a very solid choice**, arguably better than most generic diffusion-style MCP tools for clear text/logo work.

I wouldn’t switch tools unless:

- You need pure SVG vectors out-of-the-box (then you might look at a dedicated vector-logo generator, not an MCP), or
- You already have another MCP wired up to a specialized logo service.

Staying with nano banana via an MCP (like Nano-Banana-MCP or nanobanana-mcp) is a good call.

---

## 2. Prompt patterns for your Chrome extension (Glippy Glean Clipper)

Let’s design for:

- **Extension icon (16×16 / 32×32 / 128×128, square)**
- **Logo wordmark / lockup (“Glippy Glean Clipper”)**
- **A simplified glyph for toolbar / favicon**

### 2.1. General prompt structure for logos/icons

For nano banana, use a structure like:

[Subject] + [Concept / Metaphor] + [Style] + [Color palette] + [Composition details]  
+ [Technical constraints]

Example skeleton:

“Minimal flat logo for a Chrome extension called ‘Glippy Glean Clipper’, [concept/metaphor], [style keywords], [colors], on a plain white background, centered, no extra text, vector-style, high resolution, 1:1 aspect ratio.”

## 2.2. Concept directions for Glippy Glean Clipper

Glippy Glean Clipper = clipping + gleaning (smart capture + summarizing). Some concepts:

- **Paperclip + sparkle** (smart clipping, like a friendlier Clippy)
- **Scissors + highlight** (clipping + important bits)
- **Clipboard + spark / star** (captured knowledge)
- **Browser tab + magnet** (pulling content in)
- **Glowing snippet brackets {}** (code/text clipping)

You can ask nano banana for **multiple logo variations on one canvas** (like the DeepMind “8 minimalistic logos” example), then crop the best ones.

## 2.3. Concrete prompts you can paste into your nano banana MCP

Adjust “model”/size params according to your MCP, but keep the textual part close to this.

### A. Icon glyph concepts (square, no text)

1) Friendly paperclip + spark:

“Flat, minimal icon for a Chrome extension. A friendly, modern paperclip forming a subtle ‘G’ shape, with a small sparkle or star accent to suggest ‘glean’ and smart clipping. Use a clean, vector-style design with bold, simple shapes and no gradients. Color palette: teal and electric blue with a white background. No text, no border, centered composition, 1:1 aspect ratio, high resolution.”

2) Clipboard + highlight:

“Simple flat logo icon for a browser extension that saves web snippets. A clean clipboard outline with a highlighted strip of text on it, plus a small gleam/spark on the top-right corner to suggest intelligence. Use 2–3 colors max: deep indigo, bright lime accent, and white. No words, no background scene, just the icon on a white background, vector-style, 1:1 aspect ratio.”

### 3) Scissors + snippet:

“Minimal flat icon for a Chrome extension that clips important content. Stylized scissors cutting out a rounded rectangle snippet, with one corner glowing. Use a modern flat illustration style, thick clean lines, and a cyan + purple color palette on white. No text, no extra elements, centered, 1:1 aspect ratio.”

## B. Full logo with the name “Glippy Glean Clipper”

### 4) Wordmark + small glyph:

“Brand logo for a Chrome extension called ‘Glippy Glean Clipper’. Design a modern, friendly wordmark with ‘Glippy’ more prominent and ‘Glean Clipper’ lighter or smaller below it. Integrate a simple clipping icon (paperclip, scissors, or clipboard) either to the left of the text or as part of the letter ‘G’. Style: flat, clean, slightly playful but still professional, suitable for a productivity tool. Color palette: deep navy for text with teal and lime accent. White background, high resolution, 1:1 aspect ratio, vector-style.”

### 5) Logo sheet / variations (good for brainstorming):

“Create 6 different minimal logo concepts for a Chrome extension called ‘Glippy Glean Clipper’. Each logo should combine a simple icon related to clipping or collecting information (paperclip, scissors, highlight marker, clipboard, browser tab, or star) with the text ‘Glippy Glean Clipper’ or the initials ‘GGC’. Draw all 6 as flat black vector logos on a single white background, arranged in a grid, no extra decorations.”

(This matches the pattern from the Nano Banana Pro examples: multiple logos in black on white [DeepMind example](#).)



## C. Extension store banner / promo image

6) Chrome Web Store card-style:

“Promotional graphic for a Chrome extension called ‘Glippy Glean Clipper’. Show the extension icon large on the left, and to the right a simple mock browser window with highlighted text being clipped into a neat card. Add short tagline text: ‘Glippy Glean Clipper – Save the smart parts of the web.’ Style: clean, flat, modern UI illustration in the Google Chrome / Material aesthetic. Color palette: light background, blues/teals with one bright accent color. 16:9 aspect ratio, high resolution.”

---

## 3. How to iterate to something usable

Here’s a practical workflow using nano banana via MCP:

### 1. Exploration phase

- Use prompts like #5 (“create 6 different minimal logo concepts...” ) with **black-only vector-style output**.
- Generate several sheets until you see 2–3 marks that feel on-brand.

2.

### Refinement phase

- Pick 1–2 concepts and run:

“Refine the [chosen concept]: keep the overall shape, but simplify details, make it more symmetric and recognizable at 16×16 pixels, and adjust the proportions so it reads clearly as a Chrome toolbar icon.”

- Ask explicitly for: “no gradients, no small details, bold shapes, centered, 1:1”.

3.

### Text tuning

- Once you like the glyph, move to prompts that **combine icon + ‘Glippy Glean Clipper’**.

- If the name text is messy, ask:

“Keep the icon exactly the same, but render the text ‘Glippy Glean Clipper’ in a clean, modern sans-serif font, with correct spelling and good kerning.”

4.

#### **Export & vectorization**

- Export at high resolution (1K–2K is enough) [Gemini image docs](#).
- Then run it through a vectorizer (e.g., Figma’s image-trace, Illustrator’s Image Trace, or an online SVG converter) to get a clean SVG for the Chrome extension manifest.

5.

#### **Icon sizes**

- Create a neutral, square composition with some padding; then downscale to:
  - 16×16, 32×32, 48×48, 128×128, 256×256 for Chrome.

---

If you tell me which **visual direction** you like most (paperclip vs clipboard vs scissors vs tab), I can:

- Tighten a **set of final production prompts** tuned specifically for that motif
- And outline an exact manifest/icon pipeline for the Glippy Glean Clipper extension (file names, sizes, etc.).

---

**alright, i'd want like hero images and to build out a full webpage as well to show off the exstension so i would need images or logos for that too and i can get the screenshots for teh application. write me up a full action paln on how to achieve this and all of the above recommendataions, prompts ect so i can use it as a guide. questions?**

---

## **RouteLLM**

Here’s a concrete, end-to-end action plan you can use as a working guide for:

- Hero images
- Logos/icons + supporting visuals
- A full landing page to show off the **Glippy Glean Clipper** Chrome extension

I’ll assume you’ll use **Nano Banana / Nano Banana Pro (Gemini 3 Pro Image)** via MCP (or similar) for visuals, and either code or a no-code builder for the site.

---

# 1. Clarify the product and brand baseline

Before assets and prompts, lock in some basics.

## 1.1. Brand positioning (write this down explicitly)

Answer quickly:

1.

Who is this for?

- e.g., “Power users and researchers who save a lot of web content and want a smarter, structured, and searchable clipping workflow.”

2.

One-liner:

- e.g., “Glippy Glean Clipper turns messy browser clippings into clean, structured knowledge you can find later.”

3.

Top 3 benefits:

- “Save any web content in 1 click.”
- “Auto-organize and tag clippings for later.”
- “Search and resurface highlights when you need them.”

4.

Brand vibe:

- 2–3 words: e.g., **playful, smart, trustworthy, minimal.**

You’ll reuse these exact phrases in prompts and copy.

## 1.2. Visual direction

Decide:

•

Color palette:

- Primary: e.g., teal or cyan for “smart clarity.”
- Accent: a friendly purple or coral.
- Backgrounds: off-white / very light gray.

•

Style:

- Flat/2.5D illustration vs soft 3D, glassmorphism, etc.
- Rounded corners, subtle gradients, not overly skeuomorphic.

You can literally tell Nano Banana Pro to “stick to this palette and visual style across all images” consistent branding is a recommended use case in Nano Banana Pro’s own tips and it’s strong at brand look consistency.

---

## 2. Plan the landing page structure

Adapt best-practice landing page structures (see examples in [this Gemini web design article](#) and AI-assisted workflows like [this guide on AI-built landing pages](#)).

### 2.1. Suggested sections

1.

#### **Hero**

- Headline, subheadline, main CTA
- Hero illustration/image + browser mockup of the extension

2.

#### **How it works (3 steps)**

- Step cards with simple visuals

3.

#### **Key benefits / features**

- 3–6 feature cards (each with a tiny icon or micro-illustration)

4.

#### **Social proof (if any)**

- Quotes, logos, or “coming soon / early adopter” framing

5.

#### **Screenshots / gallery**

- You’ll use real extension screenshots here, plus one “composed” image

6.

#### **FAQ**

- 4–6 short Q&A

7.

#### **Final CTA strip**

- “Add to Chrome” / “Join waitlist”

You can use me later to generate HTML/CSS or a builder-ready copy and layout based on this structure.

---

### 3. Image asset checklist

You'll want:

1.

#### **Logo & wordmark**

- Square version (favicon, extension icon)
- Horizontal version (header / hero)

2.

#### **Hero illustration(s)**

- 16:9 or 3:2 for desktop hero
- Optionally a 9:16 variant for social or mobile focus

3.

#### **Feature illustrations / spot art**

- 3–6 simple, consistent illustrations

4.

#### **Store assets for Chrome Web Store**

- Icon set (128x128, etc.) – you can derive from the logo
- Prominent banner image

5.

#### **Marketing mockups**

- Browser window with Glippy Glean Clipper overlay
- “Workspace” / productivity hero scenes

You said you can grab extension screenshots – perfect. Use those for credibility and use AI images for all the “conceptual” visuals around them.

---

### 4. Prompts for logo and icon system

Use Nano Banana Pro's text rendering strengths and style controls ([Google's prompt tips for it are here](#)).

## 4.1. Logo, glyph, and wordmark (exploration pass)

### Prompt 1 – Glyph only (square app/extension icon)

“Design a clean, modern logo icon for a Chrome extension called ‘Glippy Glean Clipper’.

Concept: a stylized paper clip merged with a highlight marker or glint, representing clipping and gleaning useful info from the web. The icon should be simple, bold, and readable at small sizes.

Style: flat or 2.5D vector-style illustration, soft rounded corners, minimal details. Use a calm, intelligent color palette: teal or cyan as the primary, with subtle purple or indigo accents. White or very light background.

No text in this version. Focus on a unique, memorable shape that would work well as a Chrome extension icon and favicon.”

### Prompt 2 – Wordmark + glyph composition

“Create a full logo lockup for ‘Glippy Glean Clipper’ using the previously designed glyph style as inspiration.

- Include both the icon to the left and the text ‘Glippy Glean Clipper’ to the right.
- Typography: friendly, modern sans-serif, slightly rounded, good legibility at small sizes.
- Color: apply the same teal / cyan primary color with a neutral dark gray or navy for the wordmark.
- Layouts: generate a few variations – 1) horizontal lockup with icon on the left, 2) stacked lockup with icon above the name, centered.
- Keep the overall look professional but a bit playful; it should feel like a trustworthy productivity tool, not a toy.”

Then iterate, e.g.:

“Make the word ‘Glippy’ slightly bolder and more playful, and ‘Glean Clipper’ lighter and more understated. Keep the same icon, colors, and overall balance.”

---

## 5. Hero image prompts

Aim for a hero that clearly shows: “this is a browser extension that organizes clippings.”

### 5.1. Primary homepage hero (desktop, 16:9 or 3:2)

“Create a 16:9 hero illustration for a landing page of a Chrome extension called ‘Glippy Glean Clipper’.

Scene: a modern browser window open on a webpage, with a floating extension panel on the right showing captured highlights, tags, and a clean list of saved clippings. The UI elements should be generic enough not to copy any brand, but clearly recognizable as a browser + extension.

Mood: bright, focused, and calm. The background should be a soft gradient or abstract shapes that match the Glippy Glean Clipper brand colors (teal / cyan primary, subtle purple accents, white/light gray base).

Style: flat or 2.5D vector-style, subtle depth, no photographic realism. Use clean lines, high contrast, and strong information hierarchy.

Include subtle visual metaphors for organization, like tidy stacks of cards, labels, or lines connecting clippings to a central ‘knowledge hub’.

Leave enough clean space on the left side for a headline and CTA button overlay.”

### 5.2. Alternative hero / social promo (portrait, 9:16)

“Create a 9:16 vertical hero image for social promotion of ‘Glippy Glean Clipper’, a Chrome extension.

- Top: the logo and name ‘Glippy Glean Clipper’.
- Middle: a simplified browser window with a prominent extension panel showing 3–4 neat clipping cards, each with a title, a short snippet, and a color tag.

- Bottom: a short tagline area with empty space for text.

Same brand style: flat / 2.5D vector, teal / cyan primary, purple accent, white background. Design for legibility on mobile screens.”

---

## 6. Feature illustrations / spot art prompts

Think in 3–4 repeatable concepts:

### 1. Capture anything in one click

“Create a small, square illustration showing a cursor clicking a ‘clip’ or ‘save’ button on a browser, and the content turning into a tidy card that flies into a neat stack. Use the Glippy Glean Clipper color palette and vector style. Minimal, no text.”

### 1. Auto-organize & tag

“Illustrate a set of saved cards being automatically sorted into labeled columns or folders, with colorful tags appearing on them. Keep the style consistent with the main hero image: flat / 2.5D, teal and purple palette, light background.”

### 1. Find it later instantly

“Show a person or abstract figure searching through a chaotic pile of pages on the left, which transforms into a tidy, searchable list of clippings on the right. Add a simple search bar with a highlighted result. Same brand style and colors.”

### 1. Sync / multi-device (optional)

“Design a minimal illustration of a browser on desktop and a mobile device, both showing the same organized clipping list, connected by a subtle line or glow, in the Glippy Glean Clipper brand style.”

---



## 7. Chrome Web Store assets

You can re-use and slightly adapt your hero and logo.

### 7.1. Store banner

“Create a wide promotional banner image for the Chrome Web Store listing of ‘Glippy Glean Clipper’.

- Include the logo and product name clearly.
- Short tagline: ‘Turn messy web clippings into structured knowledge.’
- Visual: a simple browser + extension panel composition, showing neat clipping cards.
- Style: same brand colors and flat vector illustration as previous images, with high contrast and good legibility at small sizes.”

### 7.2. Icon refinements

“Generate a set of icon variations for the Glippy Glean Clipper glyph that remain visually consistent but explore:

- solid vs outlined style,
- different background shapes (rounded square, circle),
- small tweaks in the paper clip / highlight motif.

Keep all versions simple enough to be recognizable at 16x16 px, with strong contrast.”

---

## 8. Screenshots and composite mockups

You’ll capture **real screenshots** from your extension, then you can:

- Use them directly on the page (best for trust).
- OR ask Nano Banana Pro to generate **mock browser frames** and manually composite your screenshots in a design tool (Figma, etc.), or use image editing if supported by your MCP.

Example prompt for the **browser frame** if you want AI to create an empty frame:

“Create a clean, generic browser window frame (no brand logos), with a light UI and an empty content area where I can overlay my own screenshot. Keep the browser chrome minimal and modern. Provide it in a neutral color scheme that won’t clash with the Glippy Glean Clipper color palette.”

---

## 9. Building the webpage itself

You can either:

- Use a **web builder** (Framer, Webflow, etc.) and copy these sections visually, or
- Ask me to generate HTML/CSS and then tweak.

A Gemini-style “structure-first” prompt similar to what’s suggested in [this web design guide](#):

“Create a responsive landing page for a Chrome extension called ‘Glippy Glean Clipper’.

Sections:

- 1) Sticky header with logo and ‘Add to Chrome’ button.
- 2) Hero with headline, subheadline, main CTA, and space for a hero image on the right.
- 3) ‘How it works’ section with 3 numbered steps and icons.
- 4) Key benefits section with 3–6 feature cards.
- 5) Screenshot gallery section for 3 browser screenshots.
- 6) Short FAQ section.
- 7) Final CTA strip.

Use semantic HTML5 and modern, minimal CSS. Layout: single column on mobile, two-column hero on desktop. Typography: modern sans-serif, good contrast. Add class names so I can easily hook in my own CSS later. Assume I will manually insert my own images and screenshots where you add placeholder <img> tags.”

Then iterate:

“Tighten spacing on mobile, slightly increase hero headline size on desktop, and add a subtle background gradient in the hero that matches the teal / cyan brand color.”

If you prefer, I can generate the full initial HTML/CSS in a follow-up message.

---

## 10. Workflow overview (step-by-step checklist)

You can literally follow this as a to-do list:

1.  
**Define brand basics**
  - One-liner, benefits, tone, color palette, style.
2.  
**Generate logo + icon**
  - Run the glyph and wordmark prompts.
  - Pick your favorite and possibly iterate.
3.  
**Generate hero images**
  - Primary desktop hero (16:9 or 3:2).
  - Optional vertical hero (9:16).
4.  
**Generate feature illustrations**
  - 3–4 spot illustrations mapped to your key benefits.
5.  
**Capture extension screenshots**
  - Install your extension locally.
  - Take clean screenshots focused on core flows.
6.  
**Generate Chrome Web Store banner + icon variants**
  - Use banner and icon prompts to get marketing assets.
7.  
**Build the landing page structure**
  - Either: ask me for HTML/CSS, or
  - Use a no-code builder with sections above.
- 8.

### Insert images and screenshots

- Replace placeholders with your generated assets and real UI.

9.

### Polish copy & CTAs

- Refine headlines, benefit bullets, and FAQ; we can iterate on these together.

10.

### Test and iterate

- Check on mobile and desktop.
  - Adjust spacing, contrast, and image crops.
  - Optionally A/B test hero headlines or CTAs.
- 

## 11. How I can help you next

If you paste:

- Your **preferred logo direction** (or one image link), and
- Your current **value prop / copy (even rough)**

I can:

- Draft the full landing page copy (all sections).
- Generate a **ready-to-paste HTML/CSS** starter that fits your assets.
- Tailor or extend any of the prompts above toward a specific style (e.g., more playful, more serious, more 3D, etc.).

If you want, tell me:

- 1) Whether you're planning to host this as a static site (GitHub Pages, Vercel, etc.) or using a builder, and
  - 2) If you already have any color hex codes chosen.
- 

**Review all of these to ensure we are on track**

**<https://youtu.be/YWe3lXihP3c?si=U1pY6Nx8txS-HGYv>**

**<https://youtu.be/gYNgw0lhksY?si=aEf6yGofTvgTlJ7Z>**

**<https://youtu.be/woE-5gO6aIM?si=ddup8jcNgs-0t5an>**

**<https://youtu.be/FlueWTpO6sl?si=PBFH6qUJPKgDx1wV>**

---

## RouteLLM

Here's how what you're planning lines up with those videos, plus any adjustments I'd recommend so we stay on track with Glippy Glean Clipper.

I'll reference the videos as:

- Video 1 – “How to Use NANO BANANA For LOGO DESIGN”
  - Video 2 – “Generate a Full Brand Identity with Nano Banana Pro & Veo 3.1 Automation”
  - Video 3 – “How to Design a \$10,000 Brand with Google's Nano Banana AI”
- 

## 1. Logos & icon workflow vs Video 1

### What Video 1 emphasizes:

- Use Nano Banana via Gemini “Create images” tool.
- Prompt pattern:
  - “a logo for [type of brand] with [colors] and [subject details]”
  - Decide text vs “without text”.
  - Optional: “isolated on a [color] background.”
- Iterate with **small, single changes**:
  - “Create a different version”
  - “Make the bear look happy”
  - “Add glowing outline”
  - “Replace the bear with an evil skull”
- Keep edits atomic; don't bundle 4–5 changes into one prompt.

### How your current plan lines up:

- You're already:
  - Describing logo concept (paper clip + glean/highlight metaphor).
  - Separating glyph (no text) and wordmark (with “Glippy Glean Clipper”).
  - Planning to iterate: variations in layout, color, etc.
- You're also thinking about usage contexts (extension icon, favicon, hero), which is more advanced than the YouTube example.

### Adjustments / tighten-ups:

- Apply Video 1's “**one change at a time**” rule:
  - Start from a solid base logo.
  - Then: “Change only the background to white.”
  - Then: “Make the clip thicker for small sizes.”
  - Then: “Remove any inner gradients; make it flat.”

- For extension icon readability, explicitly add:
  - “The icon must be very clear at 16x16 px; avoid tiny details.”

Conclusion: You’re fully on track with logo/icon generation, just be disciplined about incremental edits like in Video 1.

---

## 2. Full brand identity + automation vs Video 2

### What Video 2 emphasizes:

- Using a **creative AI automation tool** (Create Nodes / similar) wired with:
  - Brand strategy node
  - Color palette generation
  - Nano Banana Pro image generation
  - Icon/logo generation + background removal
  - Photography guidelines + generated photos
  - Video snippets
  - A **website prompt generator** for Lovable/Replit
- Inputs:
  - 1 brand reference image
  - 1 company description
- Output:
  - Brand strategy doc
  - Color palette with hex codes
  - Logo, icons, static images, videos
  - JSON design system
  - A big, pre-written website prompt
- Tone of advice:
  - Use automation for **breadth**, then manually **tweak**.
  - Expect “good but not perfect” and refine prompts/outputs.

### How your plan compares:

- You’re effectively doing a **manual version** of that automation:
  - Brand basics (positioning, benefits, vibe).
  - Palette + visual style decisions.
  - Logo + icon generation.
  - Hero + feature illustrations.
  - Landing page structure + prompt.
- You’re *not* yet:
  - Having a single pipeline that spits out everything in one click.
  - Generating a **JSON design system** (tokens, typography, spacing).

- Autogenerating a website prompt from a brand-strategy node.

### Should you change anything?

- For Glippy Glean Clipper (solo builder, one product), your manual workflow is actually **more controlled and less brittle** than a huge automation.
- If you want to lean into this style:
  - Use the idea of a **single master brand brief** prompt:
    - From that, generate:
      - Color palette
      - Typography choices
      - Prompt templates for Nano Banana images
      - A website prompt skeleton

Quick, “on-track” verdict: You are aligned with the spirit of Video 2 (unified brand → consistent outputs). Full automation is optional overhead; your current guided/manual pipeline is appropriate.

---

## 3. Brand quality & cohesion vs Video 3

### What Video 3 emphasizes:

This one is closest to what you’re trying to do in terms of quality bar.

Core principles from Video 3:

1.

#### Minimum viable brand ≠ random AI outputs

- Reference a high-quality studio brand as inspiration (not to copy directly).
- Define a checklist:
  - Logo + logo mark
  - Fonts (heading + body, or one family with weights)
  - Primary brand color + full palette
  - Custom illustrations & icons in **one consistent style**
  - Layout/spacing/white space discipline
  - Alignment with a clearly defined target customer

2.

#### Details make it feel like a \$10k brand

- Plenty of white space.
- Carefully tuned letter spacing.
- Consistent line weights between icons and illustrations.
- Tight color palette with re-used tokens.
- Subtle hierarchy and typographic rules (headings vs body, opacity).

3.

## Use AI as style copier, not random generator

- Feed in:
  - Reference brand images
  - Your color palette
- Use a prompt like: “In exactly this style, but change the subject to X.”
- Then extrapolate a **whole system**: icons, additional illustrations, mockups, etc.

4.

### Output set for a serious product

- Logo + mark (vectorized)
- Color palette with hexes and naming
- Base typography styles
- Icon set (in same style as illustrations)
- A couple of mockups (billboard, devices, etc.)
- A visual “brand board” combining all of the above.

### How we’re doing against that:

•

You already planned:

- Color palette and style locked early.
- Re-used style for hero + spots + banners.
- Chrome Web Store banner and icon variants.
- Feature illustrations and consistent metaphors.

•

Where we can level up, based on Video 3:

#### 1. Explicit brand board / checklist

- Make a single canvas (Figma or similar) where you lay out:
  - Final logo + icon.
  - Typography styles (H1, H2, body).
  - Color palette & usage (primary, background, accent).
  - 3–4 icons.
  - 2–3 spot illustrations.
  - 1–2 mockups with your real screenshots.

#### 2. Icon/illustration style alignment

- When prompting for icons and spot art, **reference both**:
  - One of your favorite illustrations.
  - The logo glyph, if possible.
- Prompt: “In exactly this style, but depict [subject].”

#### 3. White-space and hierarchy in the landing page

- When we generate your layout code, we’ll:
  - Use ample padding.
  - Limit max-width (~960–1200px) for text blocks.



- Explicitly define font sizes and line-heights.

Conclusion: You're absolutely on track conceptually. The only missing piece is formalizing the "brand board" and ensuring icons/illustrations share a clearly enforced style. That's easy to add.

---

## 4. Implications for your Chrome extension landing page

Pulling all three videos together, the key "sanity check" points:

1.

### **Brand system first, then page**

- Videos 2 and 3 align with what we're doing:
  - Lock in name, tone, palette, logo, styles.
  - Then build the site from that base.
- You're doing this correctly (no need to start from the page and then retro-fit branding).

2.

### **Nano Banana usage is exactly as recommended**

- Video 1: Use detailed prompts, iterate with small adjustments.
- Video 2: Use Nano Banana Pro as a node in a broader system.
- Video 3: Use Nano Banana with strong reference images and a style prompt:
  - "Minimal, modern vector illustration, soft gradients, consistent stroke weight, etc."
- Your plan to have specific, reusable prompt "recipes" for:
  - Logo icon
  - Wordmark
  - Hero
  - Features
  - Store banner
- ...is right in line with that.

3.

### **Landing page structure**

- Although we hit the usage limit for analyzing your 4th link, Videos 2 and 3 plus the earlier sources we pulled on landing pages all point in the same direction:
  - Clear hero (what it is + who for + primary CTA).
  - Simple sections, each with visuals, not text-walls.
  - Use your banded color sections and iconography to make scanning easy.

- The outline I gave you (hero → how it works → benefits → screenshots → FAQ → final CTA) is fully aligned with current “AI-assisted landing page” best practices.
- 

## 5. Concrete next step to ensure we’re really dialed in

To lock this in and apply the lessons from the videos, the next best move is:

1.

**Create the Glippy Glean Clipper “brand board”** (per Video 3):

- Even roughly:
  - Chosen logo glyph + wordmark.
  - Chosen typeface(s).
  - 4–6 colors with hex codes.
  - 1 hero illustration sample.
  - 1 feature icon sample.

2.

**Send me:**

- Your chosen logo direction (description or image if you have it).
- Your current or desired color hex codes (even just primary + background).
- Any typeface you’re leaning toward (e.g., Inter, Poppins, etc.).

With those, I can:

- Draft the **full, concrete landing page copy** section-by-section.
- Generate a **clean HTML/CSS starter** that bakes in:
  - Proper spacing, hierarchy, and typography (per Video 3’s advice).
  - Clear slots for each of the images you’ll create with Nano Banana.
- Refine any of the prompts we wrote so they line up exactly with your chosen look.

From reviewing the videos and your plan:

You’re not only on track; you’re actually aiming slightly higher than most “Nano Banana logo tutorial” workflows by treating this as a real, coherent brand and not just “cool AI graphics.” The last mile is just tightening consistency and then implementing the page.

---