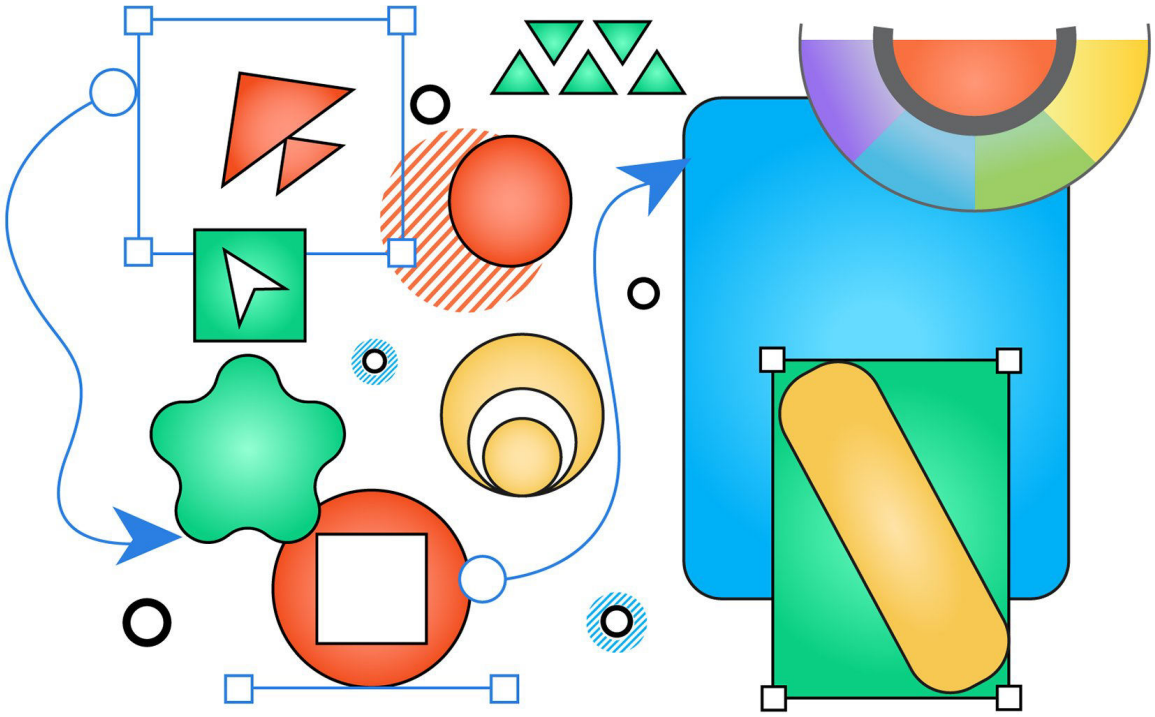
‹packt›

Design Beyond Limits with Figma



Build scalable design systems through advanced design-to-development collaboration strategies

Simon Jun

Design Beyond Limits with Figma

Early Access Publication: Design Beyond Limits with Figma

Early Access Production Reference: B31829

Published by Packt Publishing Ltd.

Livery Place

35 Livery Street

Birmingham

B3 2PB, UK

ISBN: 978-1-83620-771-9

Table of Contents

Design Beyond Limits with Figma: Build scalable design systems through advanced design-to-development collaboration strategies

Welcome to Packt Early Access. We're giving you an exclusive preview of this book before it goes on sale. It can take many months to write a book, but our authors have cutting-edge information to share with you today. Early Access gives you an insight into the latest developments by making chapter drafts available. The chapters may be a little rough around the edges right now, but our authors will update them over time.You can dip in and out of this book or follow along from start to finish; Early Access is designed to be flexible. We hope you enjoy getting to know more about the process of writing a Packt book.

1 Advanced Collaborative Design with Figma

The first time I used Figma, I was blown away by how seamlessly it supported collaboration. It was the first design tool truly built with teamwork at its core. While this approach might seem standard today, it was groundbreaking at the time. It shifted the mindset from isolated design work ("I'll perfect this until the deadline and then share it") to a dynamic process of ongoing feedback and iteration. And as we all know, consistent feedback leads to better products. Figma excels at enabling this, but it’s essential to understand the full range of collaborative features it offers.This chapter dives into eight crucial areas of collaboration that will empower your team to create outstanding products by improving communication, streamlining workflows, and ensuring alignment between design and development. Mastering these areas will help you avoid common pitfalls, reduce inefficiencies, and deliver high-quality products faster, with fewer roadblocks along the way.The following is the list of recipes we will cover:

- Communication gaps between designers and developers
- Repetitive design meetings waste time and delay progress
- Designers may not anticipate technical limitations, leading to missing details for developers
- Constant changes in the design
- Streamlining Version Control and Iterations
- Unlocking Advanced Sharing and Permissions Settings
- Real-Time Co-Designing for Maximum Team Productivity
- Facilitating Feedback Loops in Collaborative Designs

Communication gaps between designers and developers

One of the most persistent challenges in product development is the interaction between designers and developers. While both disciplines work

toward the same goal—creating functional, user-friendly products—they often approach problems from entirely different perspectives, leading to friction and miscommunication.

What's the problem?

Designers and developers are two essential forces in product development, but they often struggle to understand each other. As a designer, it's important to remember that what you create in Figma isn't the final product—the production code is. However, despite speaking the same language, many teams face miscommunication due to differing terminologies and perspectives. This language barrier can lead to inefficiencies, errors, and unnecessary back-and-forth discussions.Here are some common terminology differences between Figma and code:

Figma	**Code**
Auto Layout	Flexbox
Corner Radius	Border-radius
Frames	Divs
Color Styles	CSS Variables
Prototypes	Clickable Mockups (via JS/HTML)

Table 1.1 – Terminology difference between Figma and code

Understanding these differences is crucial to ensuring a smoother workflow and more effective collaboration between designers and developers.

How to fix it?

To bridge this gap, designers and developers must actively work toward a common understanding:

1. **Involve Developers Early**: Bringing developers into the design process from the start ensures technical feasibility and helps avoid late-stage surprises.
2. **Establish a Shared Vocabulary**: Document and define key terms that designers and developers use differently. This makes communication

clearer and reduces misunderstandings.
3. **Use Figma's Dev Mode**: Leverage Figma's developer handoff features to provide precise specifications and make it easier for developers to inspect design elements.
4. **Encourage Cross-Team Learning**: Designers should develop a basic understanding of development constraints, while developers should familiarize themselves with design principles.
5. **Create a Feedback Loop**: Regularly review designs together to ensure alignment and address potential issues before they escalate.

There are two primary ways to establish continuous feedback loops:

- **Scheduled Weekly Feedback Sessions**: This is the most common approach, where teams set up recurring calendar meetings for feedback. Designers present their progress, and other designers and developers discuss any necessary changes or improvements.
- **Daily Asynchronous Feedback**: A more fluid approach, which I personally prefer, involves integrating feedback into the team's everyday workflow. At dotidot, designers and developers provide feedback asynchronously at the end of each day using Loom recordings. This fosters a culture where feedback is a natural and ongoing part of collaboration, rather than being confined to structured meetings.

Even if your team lacks hybrid designer developers, following these practices will save significant time, reduce confusion, and create a more seamless workflow between both disciplines.

Repetitive design meetings waste time and delay progress

What's the problem?

Endless design meetings often consume valuable time without adding significant value. While collaboration is essential, excessive discussions over the same design elements can lead to delays, frustration, and decision

fatigue. Instead of refining and progressing, teams can get stuck in cycles of unnecessary revisions and redundant conversations.A key difference between junior and senior designers is how they approach documentation and handoff. Experienced designers go beyond aesthetics—they consider functionality and development requirements upfront, reducing the need for excessive meetings. By structuring Figma files effectively and using built-in collaboration tools, teams can minimize disruptions and maximize efficiency.

How to fix it?

To break free from the cycle of excessive design meetings, we can leverage screen annotations. Effectively using screen annotations helps convey the full story behind your design. Here's how to utilize them better:

- **Designer Notes**: Add context directly next to your design with designer notes. Avoid using comments or Figma Dev annotations, as these require all users to have **Dev** mode enabled. Instead, rely on prepared elements for clarity.

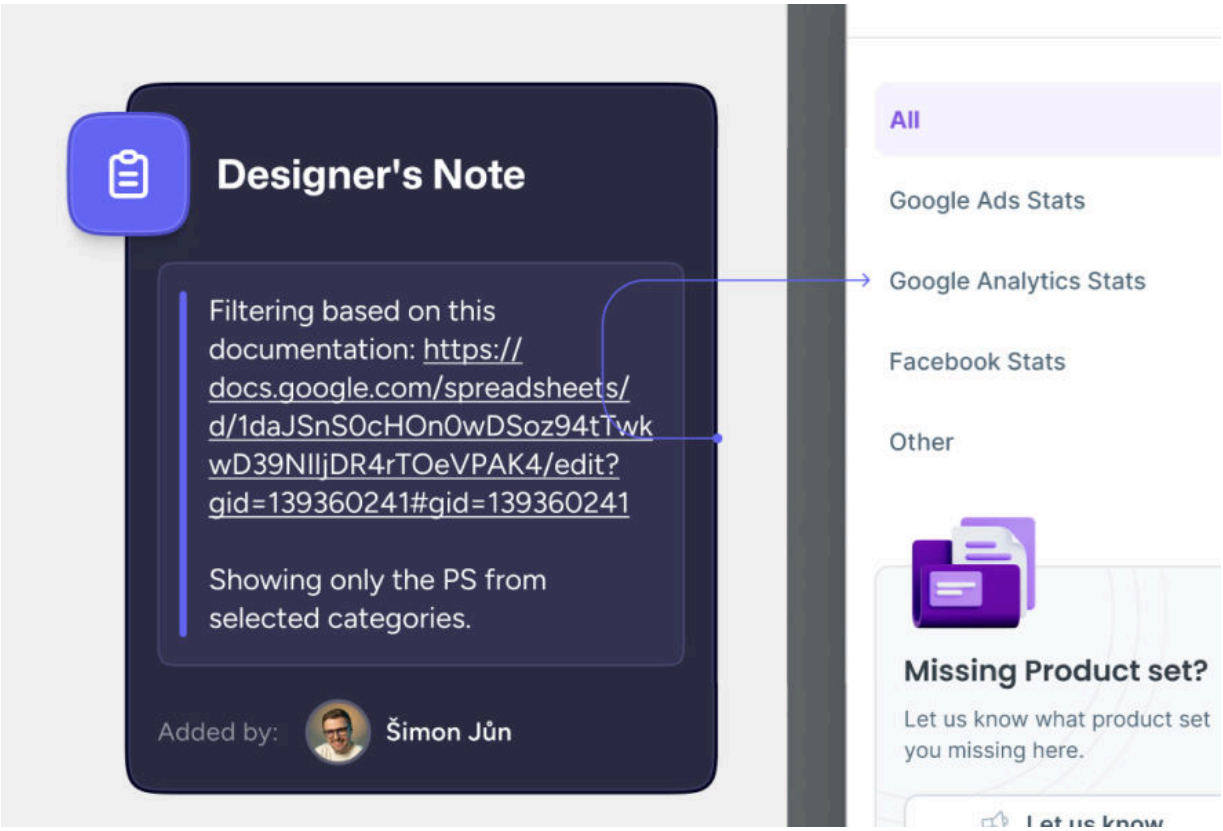


Figure 1.1 – An example designer note

This card demonstrates how to provide additional context within Figma designs. The note explains that filtering is based on provided documentation and highlights specific parameters used in the project. Visual annotations direct attention to essential elements, ensuring clarity and fostering collaboration.

- **Copy Note**: Provide specific copywriting instructions to ensure developers understand how to implement text accurately within the design.

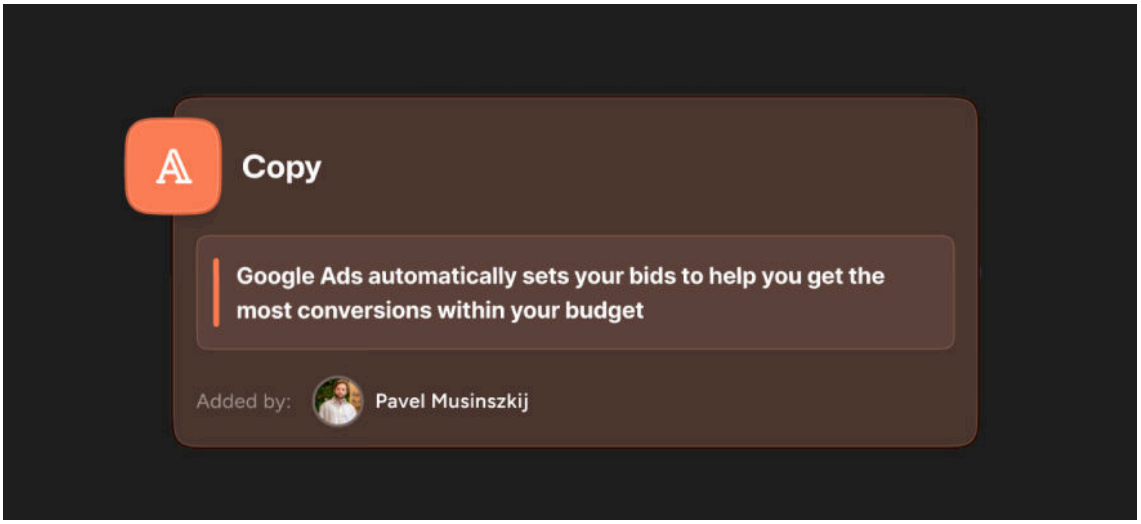


Figure 1.2 – An example copywriting instruction

- **Variant Annotations**: Don't duplicate your design just to showcase small changes. Duplicating entire designs can create confusion about development complexity—on first glance, a project may seem much larger and more complex than it actually is. Additionally, excessive duplication can overwhelm developers, making it easier to miss crucial details buried within an abundance of similar-looking screens. Use variant cards to illustrate differences clearly and help developers grasp the full scope of adjustments. For example, opening a Figma file with 10 screens might appear overwhelming but could represent small changes on a few screens. Variants streamline first impressions, time estimation, and overall clarity. In the image below, you can see an example of a dropdown and an alternative version of a tab component.

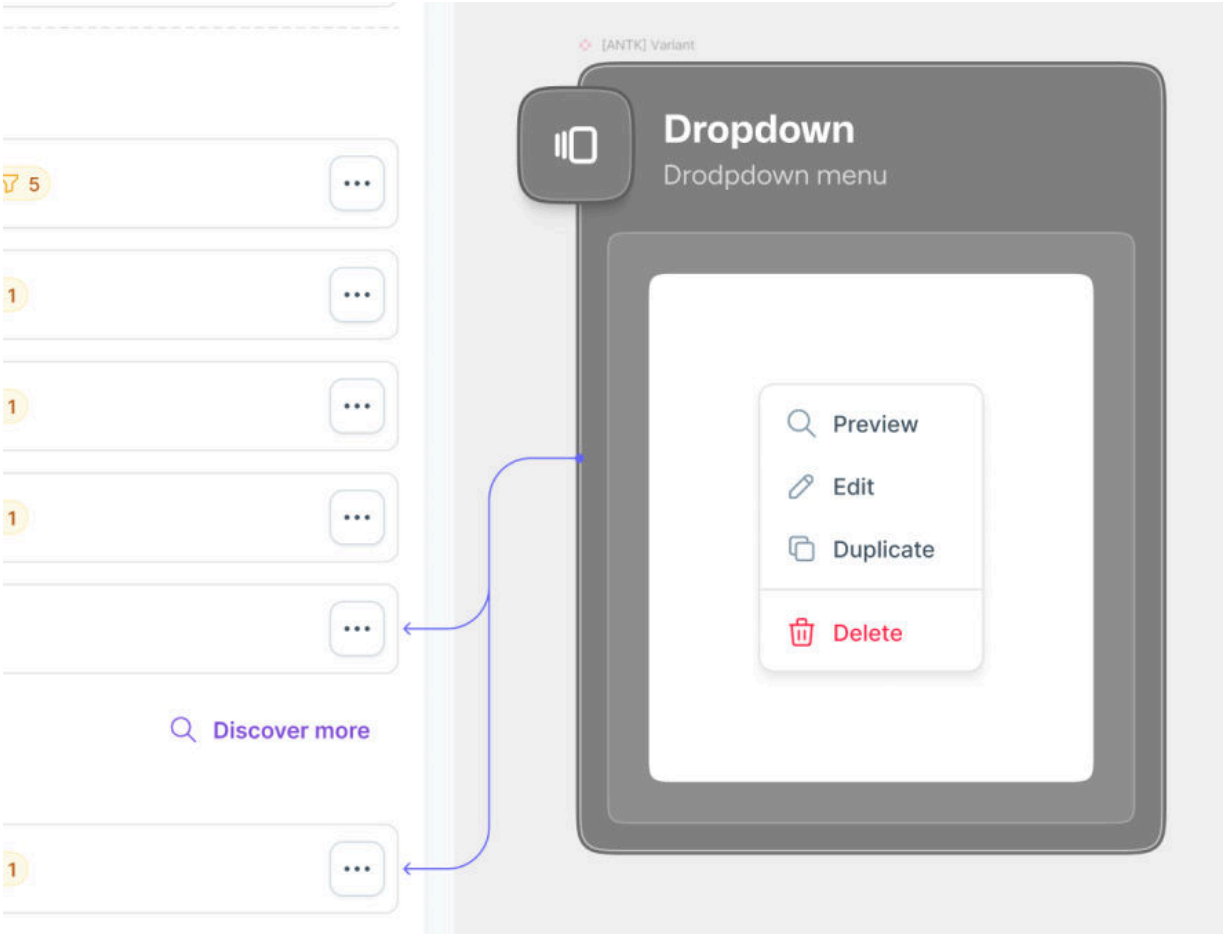


Figure 1.3 – Dropdown use case

Here’s an alternate version of tab variation case:

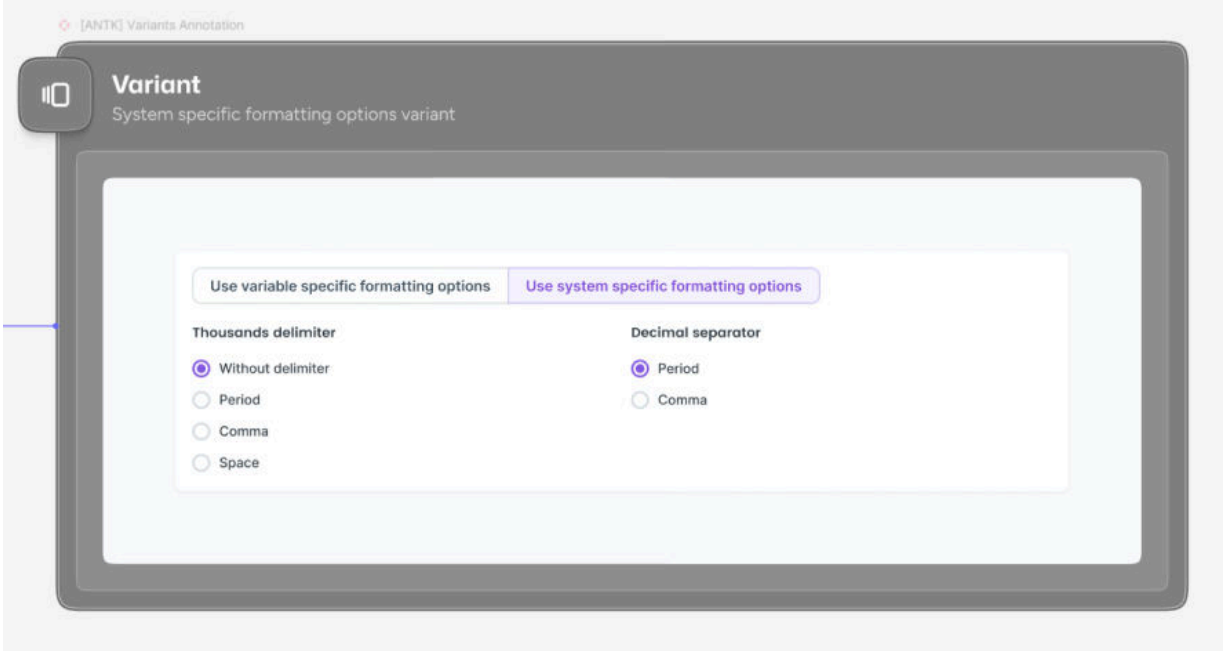


Figure 1.4 – Tab variation case

- **Highlight Nice-to-Have Features**: While agile workflows prioritize speed, it's sometimes beneficial to design beyond the **Minimum Viable Product** (**MVP**), which is the simplest version of a product that includes only the core features needed to satisfy early users and gather feedback for future improvements, to showcase the broader idea.. Use visually distinct elements to denote nice-to-have parts (for example, a feature like **Search**). This approach improves communication with stakeholders, ensuring they see the complete vision and grasp your intent more effectively.

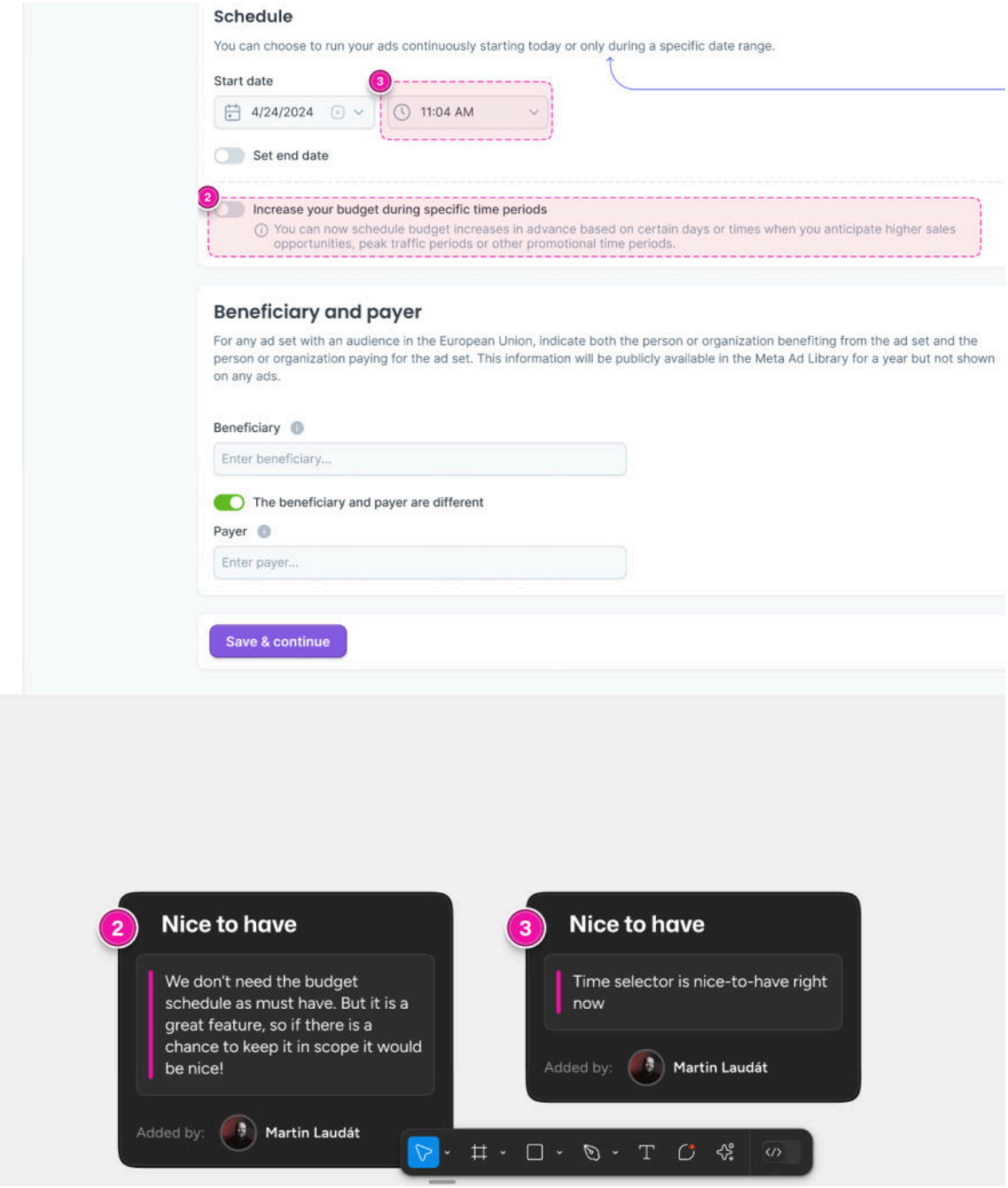


Figure 1.5 - Nice to Have Cards in Design

- **Ticket Cards**: Link your Figma designs directly with project management tools like Jira by adding ticket cards. These cards can display responsible team members and task statuses right in Figma.

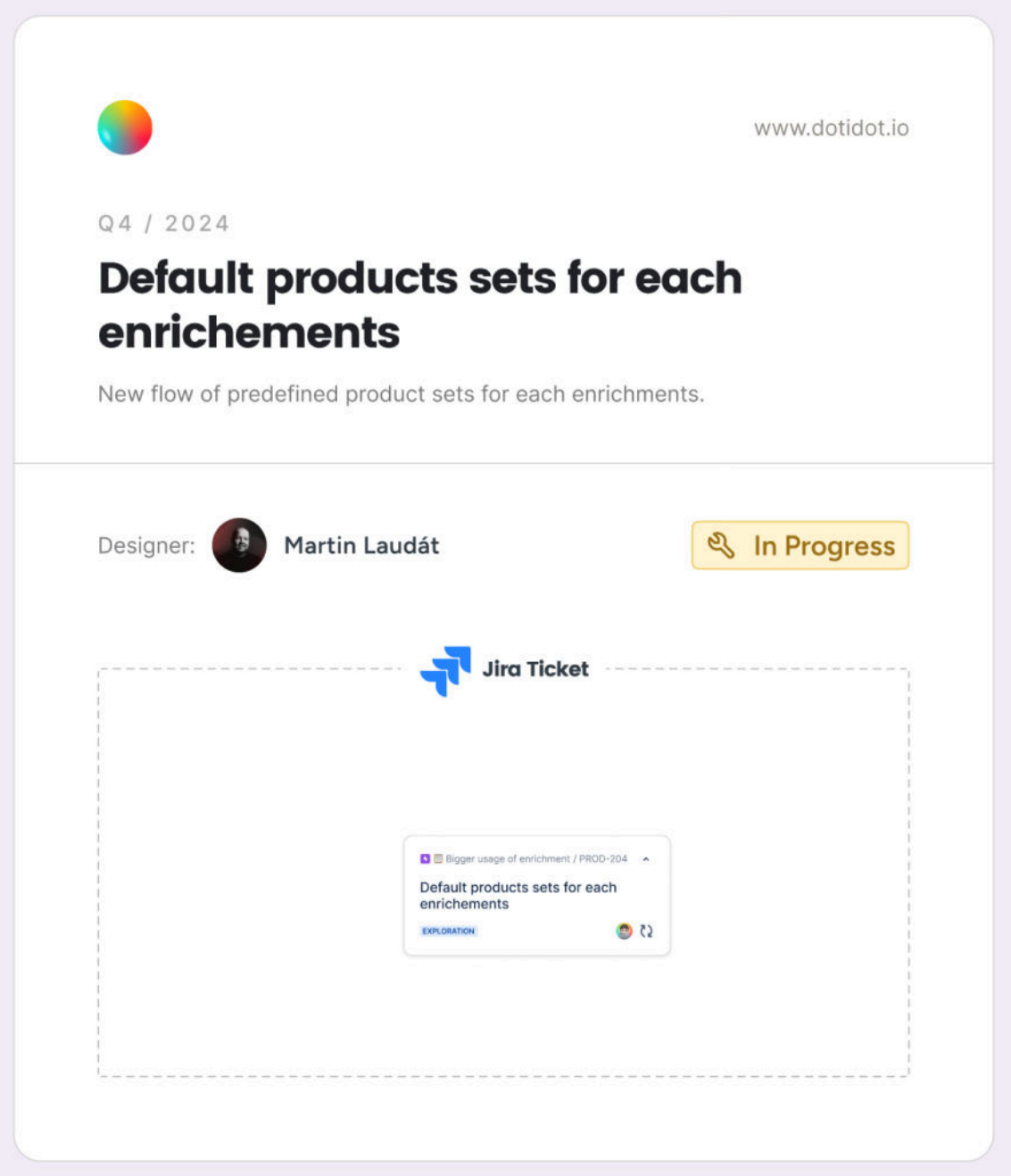


Figure 1.6 - Ticket Card for new design project

- **Flow Headings**: Include flow headings to outline logical sections of your design. This helps collaborators and developers understand the structure and implement the design as cohesive, story-driven blocks for quicker development.

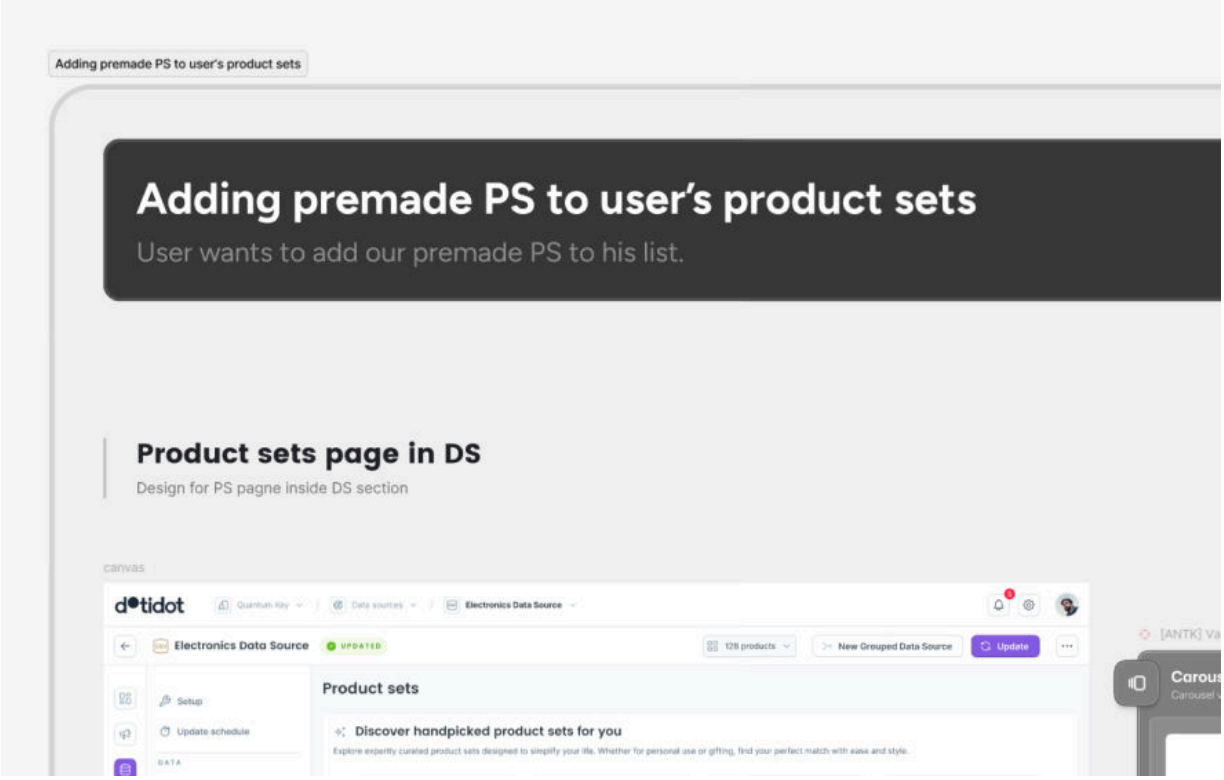


Figure 1.7 - Flow heading for a new section in design file

All of these annotations are crucial for every Figma handoff. Remember, a good design file isn't just about frames with designs; it's about effectively transferring the right information from design to development.

Loom Video Walkthrough

Record videos for everything! Loom is an incredible tool—you just click, record, and your video is automatically uploaded to the cloud. You can then paste the link directly into Figma. Everything is clearly explained, and anyone can revisit the video as needed.I shared my workflow on LinkedIn:

1. Record a short Loom video explaining the feedback or a new feature idea. I include examples and walk through our app directly in the video.
2. Loom AI automatically generates a transcript of the video.
3. I use the transcript as input for GPT, paired with a custom prompt tailored to our Jira ticket structure (Description, Requirements,

Acceptance Criteria, etc.).

4. Within seconds, I have a complete Jira ticket ready to go! I added the Loom video link to the ticket for reference, ensuring clarity for designers.

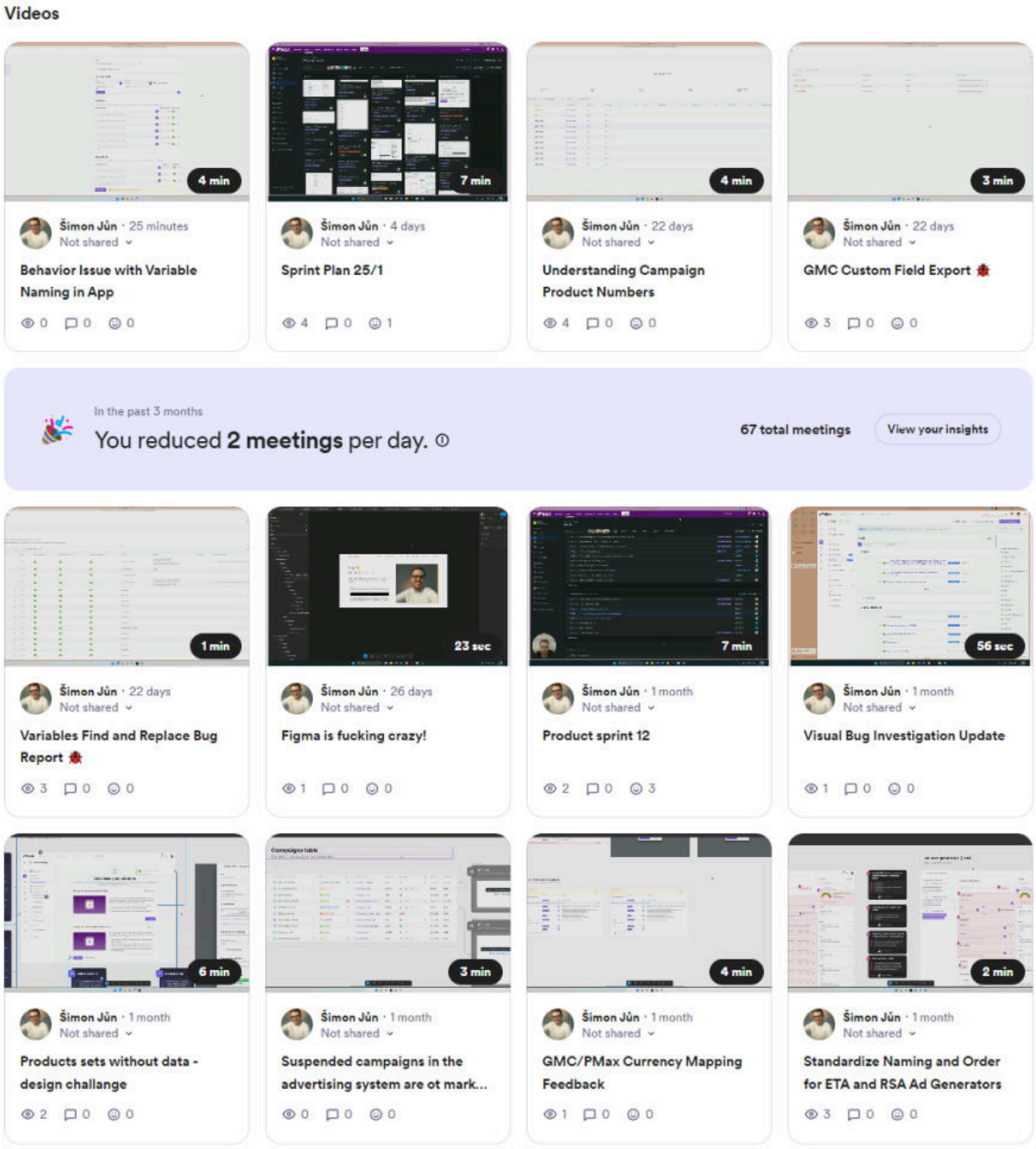


Figure 1.8 - Loom video gallery

Designers may not anticipate technical limitations, leading to missing details for developers

What's the problem?

Designers often lack awareness of technical limitations, which can lead to missing details for developers. While designers excel at understanding user needs and business goals, gaps in technical knowledge can create challenges. Legacy tech debt, platform constraints, and performance considerations are often overlooked. Without early collaboration, these issues surface late in development, causing costly revisions.

How to fix it?

The following steps will help designers to know and understand the technical limitations early in the design process, ensuring that the collaboration will be faster and smoother for both sides.

1. **Involve Developers Early**: Bring developers into the design process from the start to identify technical constraints before they become blockers.
2. **Assign a Developer Partner**: Designate a developer as a support resource for the project to provide real-time technical insights.
3. **Use Figma's Spotlight Feature**: Leverage quick, real-time calls to align on design feasibility without unnecessary scheduling overhead.
4. **Clarify Technical Needs**: Ask developers for input on system limitations, reusable components, and performance considerations upfront.
5. **Document Technical Constraints**: Keep a shared space where designers can reference key limitations and best practices to avoid repetitive issues.

By fostering an open dialogue between design and development, teams can avoid misalignment, reduce unnecessary rework, and build products more efficiently.

Constant changes in the design

What's the problem?

Frequent design changes are inevitable in many projects, but they can create confusion and inefficiencies. A major challenge arises when development teams are already working on one version while designers are iterating on a newer one. This leads to uncertainty about what should be implemented, potential rework, and a frustrating experience for both designers and developers. In one project with a large insurance company, we faced this exact issue—multiple design iterations were being worked on simultaneously while the older version was still under development. Developers struggled to build a product that kept evolving beneath them.

How to fix it?

To manage continuous design changes effectively, consider these two primary approaches:

1. **Branching for Isolated Iterations**: Works similarly to development branching—designers create a branch from the master file, make changes, and merge it back when ready.

It prevents unnecessary changes from affecting the live version.*Challenges*: Requires Figma's Organization or Enterprise plan; designers may be unfamiliar with branching workflows. To overcome this, consider a developer-led training session on Git-based workflows.

1. **Versioning for Clear Change Management**
    A. Manually create a version checkpoint before significant updates.
    B. Label versions clearly (for example, `Onboarding_Flow_v1.2`) to document progress and provide a stable reference.
    C. Use version descriptions to highlight key changes, reducing confusion for developers and stakeholders.

By adopting structured version control and branching techniques, you can reduce confusion, minimize rework, and create a smoother workflow

between design and development teams.

Streamlining Version Control and Iterations

What's the problem?

Many teams underutilize Figma's versioning features, missing out on its full potential. The tool offers two types of versioning—automatic (autosaves) and manual—but manual versioning, when used strategically, can significantly enhance workflows. However, teams often struggle with knowing when and how to create manual versions, leading to inefficiencies in sharing and collaboration. Without clear guidelines, this powerful feature becomes underused or misapplied, limiting its impact on productivity.

How to fix it?

When you decide to create a manual version, it should be in time when you are ending a workflow cycle. It can be the version that will be sent to stakeholders or client, or the version you are working on in two-week sprints, and you need to send the version to development. When to do it is up to you, and it is different across the teams that I worked with, but the *how to do* is more important.Every version can have a name and description. Let's talk about naming first. You can use several naming structures to maintain consistency:

- **By Milestones**: `ProjectName_MilestoneName_vX`

Example: `Dashboard_v1` or `Onboarding_Flow_v3`

- **By Date**: `ProjectName_YYYY-MM-DD`

Example: `Onboarding_Flow_2025-01-15`

- **By Sprint or Release Cycle**: `ProjectName_SprintX` or `ProjectName_ReleaseX`

Example: `Onboarding_Flow_Sprint12` or `Onboarding_Flow_Release2.1`

- **By Stakeholder Review Stage**:
  `ProjectName_StakeholderType_Stage_vX`

Example: `Onboarding_Flow_Review_v1` or `Onboarding_Flow_Feedback_v2` Consistency is key—ensure the same structure is used across all your files to avoid confusion.For descriptions, keep them concise but informative. Use them as a changelog to highlight differences from the previous version. Here's a quick tip: leverage emojis for clarity:

- ✅ Approved changes
- ➕ Additions
- ❌ Deletions

The image below showcases the Figma Version Modal, where you can document and track all changes for future reference, ensuring clarity and seamless collaboration.

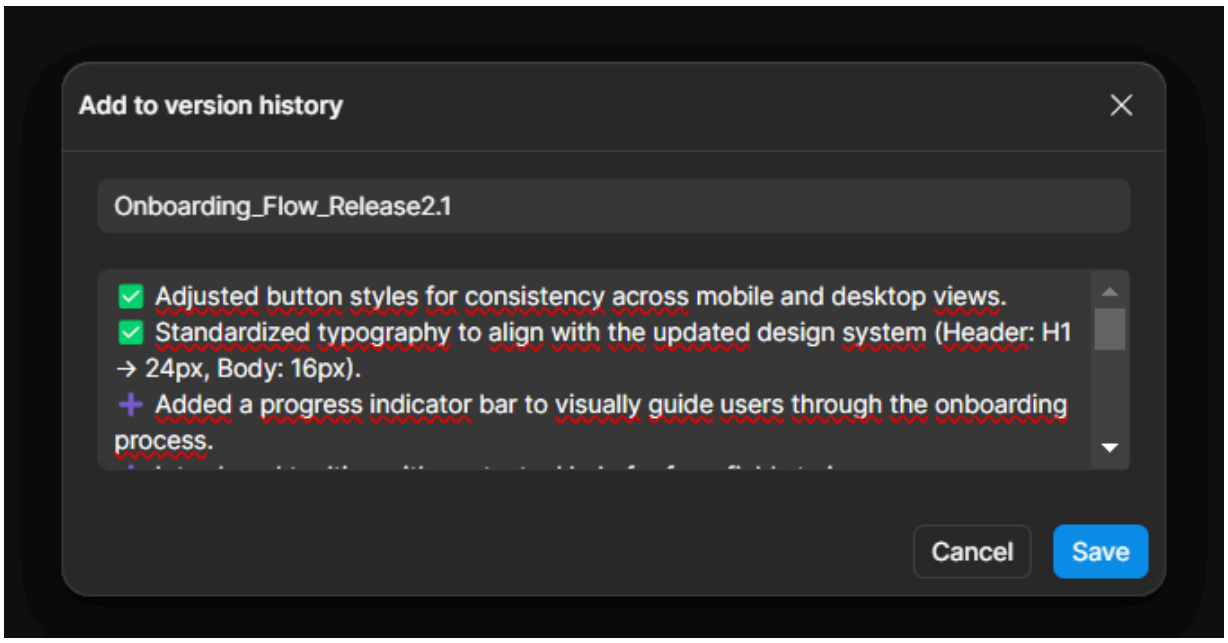


Figure 1.9 - Figma New version Modal

Once you've established versions, you can easily copy the link for a specific version and share it with developers or stakeholders. However, note that older versions are view-only—people can't comment on them. A

workaround for this limitation is duplicating the version into a separate file where comments and edits can continue. Meanwhile, keep the "hard" version as a separate file to maintain a clear record. Just remember to archive or close older versions to prevent an overflow of active files.

Unlocking Advanced Sharing and Permissions Settings

What's the problem?

Managing sharing permissions in Figma can become a challenge, particularly when dealing with external collaborators or large teams. Improper settings, such as leaving files open to "anyone," can lead to unauthorized access, unintentional edits, or even the mishandling of sensitive content. Additionally, using group email addresses for collaboration can limit individual accountability in comments. Without a clear structure for separating internal and external files, informal content can unintentionally reach professional audiences, creating potential embarrassment or confusion.

How to fix it?

Figma has a standard level of sharing permissions—not overly complex—but there are a few valuable tricks I've learned the hard way during my career:

1. **Rule number one**: If you open the file link to **anyone**, ensure that the **What can they do on View** setting and **Viewers can copy, share, and export this file** options are unchecked. This will save you from unnecessary expenses caused by unintended editors accessing the file (trust me, it happened to me!) and provide some guardrails against unauthorized copying of your file by anyone online.
2. **Rule number two**: Reread rule number one! Following this step alone will save you both money and headaches.

There's more...

For better security, consider adding a password to the file or avoid using **Anyone** as an access setting altogether. Instead, add specific email addresses. When inviting a large number of people, you can copy and paste multiple email addresses at once, or use group email addresses like `marketing@company.com` or `figma@company.com`. Group access works seamlessly, as Figma doesn't limit access from the same account being used by multiple people. The only downside is that comments from group accounts won't display individual names unless explicitly noted.For Organization and Enterprise plans, you gain access to **Activity Logs**, providing a detailed overview of who accessed your files and what actions they performed.Enterprise users also benefit from a **Password Protection Required** feature, which enforces password use for all shared files—adding an extra layer of security.Another smart approach is to establish a system where files shared externally, such as with clients or partners, are clearly separated from internal ones. This minimizes the risk of accidentally including informal content in professional files. Let's be honest—most of us have added memes or jokes in Figma files for team fun at some point. I once experienced the embarrassment of such a file being shared with a client's C-level executives. Luckily, they found it amusing, but it was a lesson learned! To avoid such situations, I recommend tagging shared files with an emoji and text in the name, such as `[📎 SHARED WITH CLIENT] File Name`. This simple step can save you from potential awkwardness in the future.

Facilitating Feedback Loops in Collaborative Designs

Figma changed my perspective on design when I realized how deeply it integrates collaboration into the core of the experience. However, many designers still don't take full advantage of its potential. I've mentored numerous young designers who are hesitant to ask for feedback.*"What if someone finds out I'm not that good?"* they often ask me. My response is always, *"So what?"* Design is fundamentally about solving problems, and humans are naturally better at solving problems together. If we can hunt mammoths as a team, why can't we share an early Figma file? Feedback is your best ally in design.I frequently tell my team that having them to rely

on allows me to work faster. I don’t need everything to be perfect before sharing—it’s through collaboration and early input that we achieve our ultimate goal: building the best possible product.There are countless ways to foster this collaborative feedback process. While it takes experimentation to find what works best for you, I’ll share a few methods to get you started.

What’s the problem?

Many designers hesitate to fully embrace Figma's collaborative features, often avoiding feedback due to fear of judgment or exposing their skills. This reluctance hampers the potential of collaborative problem-solving and slows down design progress. The lack of early input leads to delayed iterations and missed opportunities for improvement. Establishing a culture of open and effective feedback is crucial for leveraging Figma's full potential and creating the best possible products.

How to fix it?

Effective communication is crucial for a product’s success. In recent years, the rise of remote teams and increasing workloads have made seamless communication even more critical. Teams must work proactively to ensure clarity and collaboration despite these challenges.

- **Feedback Slack channel**: Sometimes, you need feedback from people outside your design team. At dotidot, for example, we often turn to our Automation Strategists or performance marketing geeks for their opinions—they align closely with our users’ perspectives and often provide insights smarter than ChatGPT :D.

To streamline this, we created a dedicated Slack channel specifically for design feedback. It’s crucial to make the request as simple and straightforward as possible to keep people engaged over time. Every message should include:

- Title of the request
- Brief description of the problem or assumption
- Include a video explanation for more complex issues

- Link to a specific section in Figma (not the entire file)
- Deadline for responses to help them plan their time
- Mentions of all relevant team members for feedback, along with those who can contribute to the discussion in the thread below.

- **Recording of every design critique**: Yes, here it is again—Loom. You might think I own the company given how often I mention it, but I genuinely love the tool. Whenever we have a quick call or live design critique, we always record the session and link it directly in Figma next to the design. This allows designers to focus on the discussion—thinking critically about the design—instead of scrambling to write comments or notes. After the meeting, you can revisit the recording and review specific parts as needed. Loom's AI-generated chapters make it easy to jump to the sections that matter most.

If you plan a long meeting (though in 99% of cases, long meetings aren't necessary), consider breaking it into multiple shorter videos instead of one lengthy Loom recording. This approach will help you move faster in the future and make it easier to revisit important points.By the way, these video recordings also serve as excellent training materials for new team members. They can quickly learn from past challenges or *mistakes*, significantly speeding up the onboarding process.

2 Leveraging Figma's Plugin Ecosystem

Plugins are an incredible addition to Figma, enabling users to accomplish much more. However, with tens of thousands of plugins available, it can be challenging to select the right ones. In this chapter, I will share my approach to choosing plugins, when to encourage your team to explore plugins, and finally, how to decide if developing your own plugin from scratch is the best solution. We will discuss these 6 areas in this chapter:

- Selecting Essential Plugins for Enhanced Functionality
- Streamlining Workflow Efficiency with Time-Saving Plugins
- Automating Repetitive Tasks with Plugin Integration
- Specific Project Needs solved by plugins
- Linking Figma to Other Software for Cross-Platform Integration
- When to Consider Writing Your Own Plugin

Selecting Essential Plugins for Enhanced Functionality

Before showcasing selected plugins, it’s important to discuss the process of selecting them. At first glance, choosing a plugin might seem simple—just pick one and test it. But what happens if you build your workflow around a plugin that later becomes unsupported or outdated? This could disrupt your processes. Selecting plugins shouldn’t be a quick decision, much like choosing new design tools, website hosting, or any platform critical to your success. Here are the questions I always ask myself when selecting plugins:

- Am I the only user?
- Is this a one-time use?
- Is this plugin paid?
- Is it developed by a company?
- Are there any reliable alternatives?

These questions are important for ensuring the longevity of your decision.

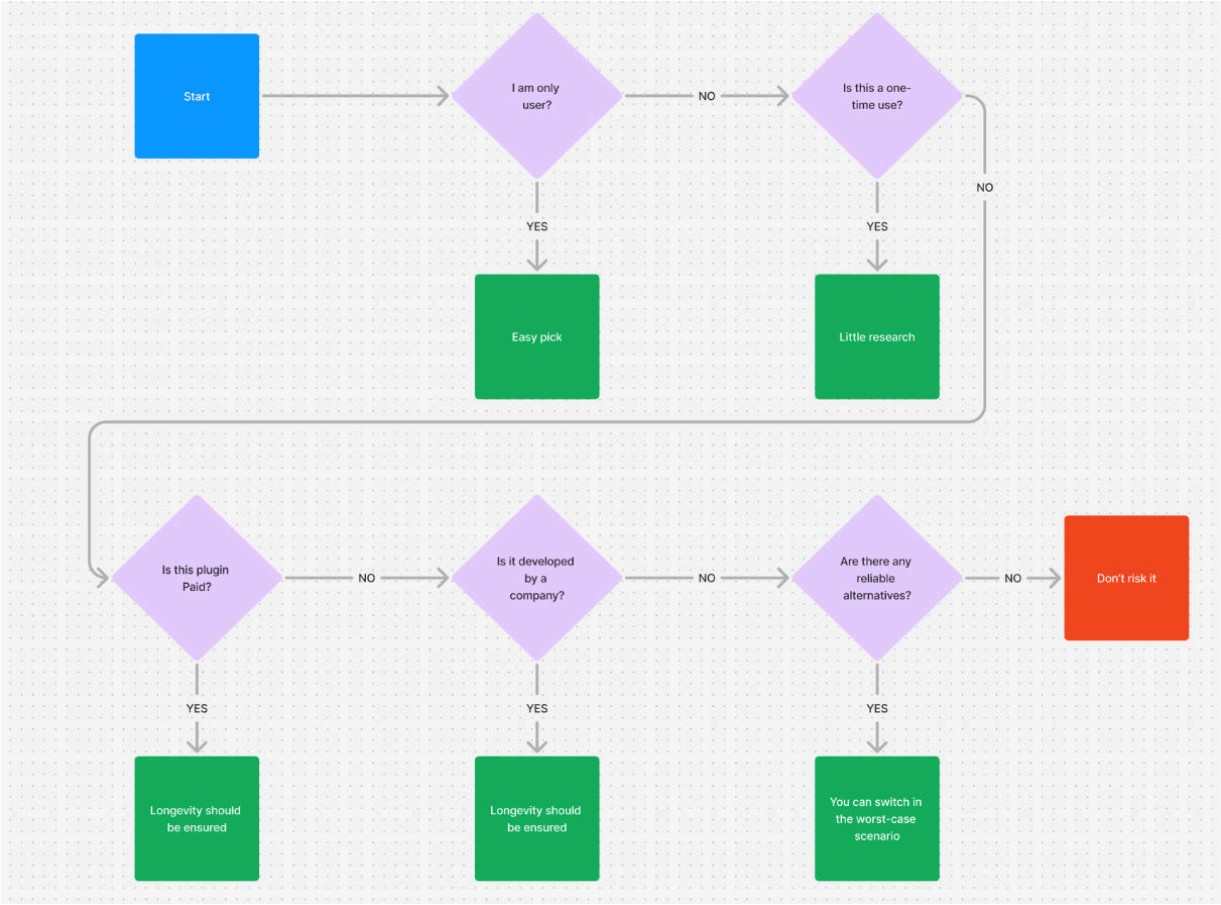


Figure 2.1 – My plugin decision workflow

Let's look at the above-mentioned questions in detail.

Am I the Only User?

Let's look at our first question: the problem and its solution.

What's the problem?

Relying on a plugin that is used only by you might seem low-risk, but what happens if it stops working tomorrow? While your team's workflow won't be affected, your efficiency might take a hit, requiring time to find an alternative or adjust your process.

How to fix it?

If a plugin is limited to your personal workflow, you have the flexibility to replace it or adjust your process if it becomes unsupported. Small productivity boosters, such as the CSV Data to Figma Plugin, fall into this category. Before committing, always check for alternative plugins and ensure there are backups available to avoid unnecessary disruptions.

Plugin Example: data.to.design

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/1133729773197702197/data-to-design-by-divriots-google-sheets-csv-json-airtable-or-notion-to-figmaOne of the most impactful plugins I used during my consulting days helped bring prototypes to life with real data. Imagine this: you're presenting a new project to a client or the management team. These folks aren't designers—they don't speak the language of Figma prototypes or production code. But what they do understand is data that feels real and relevant.Here's where these plugins shine. With just a few rows of data from the client's CRM, you can populate your prototypes with authentic information. Picture presenting an internal reporting dashboard that displays actual client names and real revenue figures instead of placeholders like "John Doe" or "$11111." The impact? Game-changing. It fosters stronger connections with stakeholders and drives home the project's relevance.You can easily map your data fields to your design and populate it in one click.

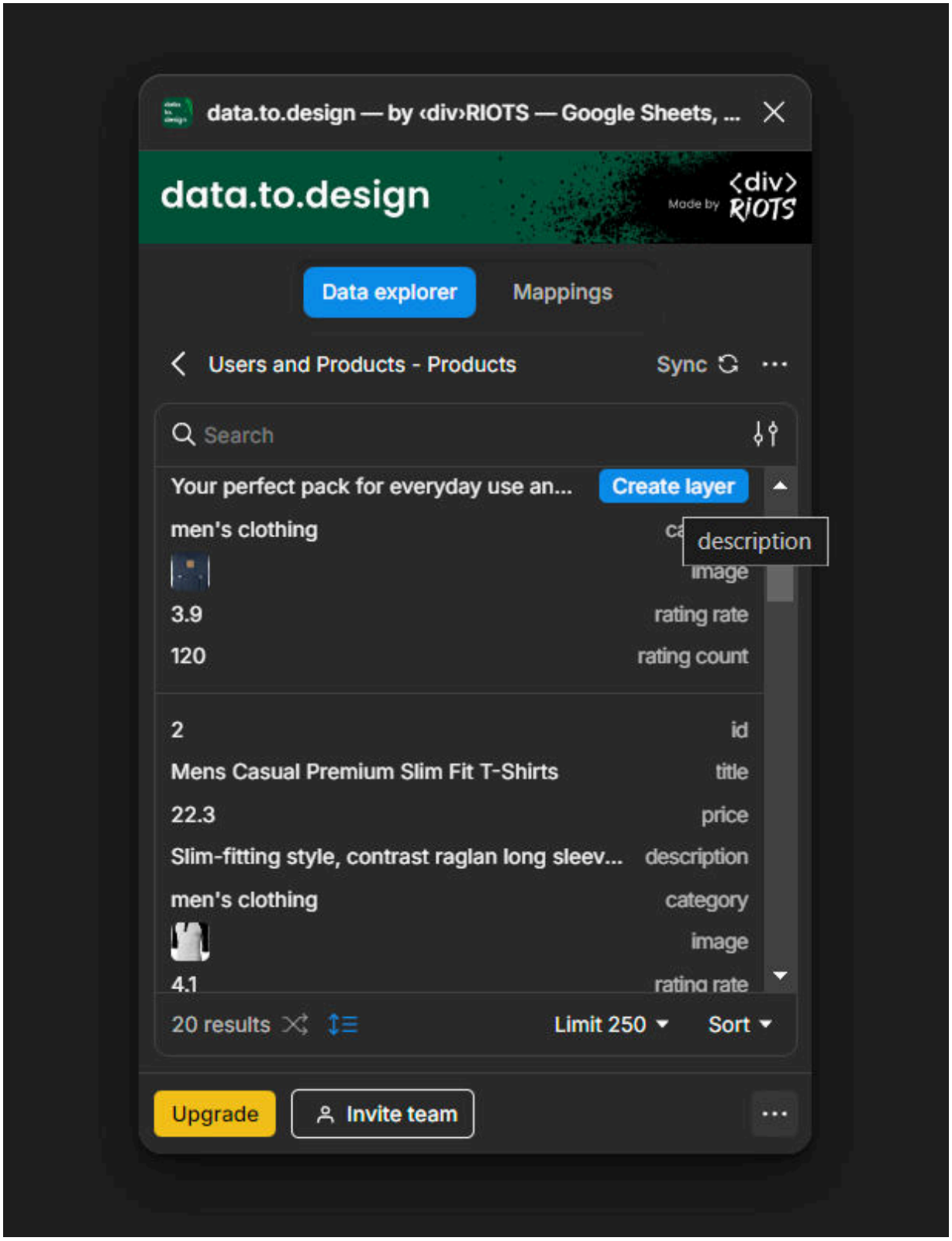


Figure 2.2 - Data explorer and data mapping in the plugin

In an era where AI is rendering Lorem Ipsum obsolete, the mantra is simple: make it real. A design packed with genuine data delivers an entirely different experience—one that resonates.

Is This a One-Time Use?

Not every plugin is meant to be a long-term addition to your workflow.

What's the problem?

When selecting a plugin for a single-use task, long-term compatibility isn't a concern. But if you expect to use it repeatedly, you need to ensure that it will still function in the future. Will the plugin work the same way next year? If not, you might find yourself scrambling for alternatives.

How to fix it?

If a plugin is intended for a one-time task, you can be more flexible in your selection. However, if it's something your team will depend on regularly, check its update history and ensure it has long-term support. A great example of a one-time-use plugin is Styles to Variable Converters, which helped many teams transition their Styles into Variables when Figma introduced the feature.Another great example is Batch Styler by Jax Six, which simplifies managing multiple styles at once.

Plugin Example: Batch Styler

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/818203235789864127/batch-stylerI used this plugin extensively in the past, so it deserves an honorable mention. As the name suggests, it works with Styles. Nowadays, most workflows have shifted to Figma Variables or Token Studio (which I'll cover below), so I don't use it as often. However, I know some designers still rely on Styles, especially for text formatting.The plugin's main advantage is its ability to update multiple styles simultaneously,

dramatically reducing the time needed for design system maintenance and updates.

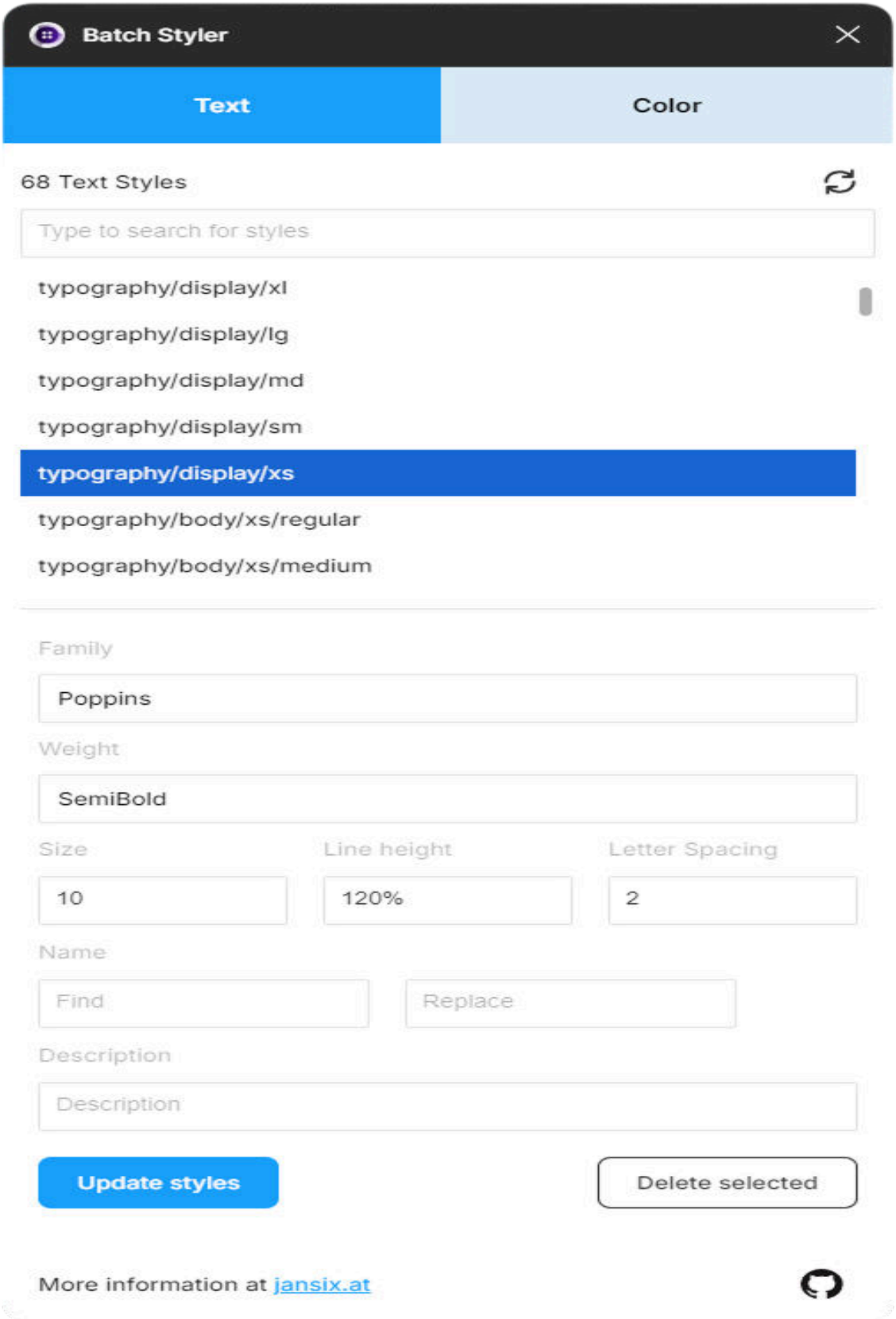

Batch Styler
Text
Color
68 Text Styles
Type to search for styles
typography/display/xl
typography/display/lg
typography/display/md
typography/display/sm
typography/display/xs
typography/body/xs/regular
typography/body/xs/medium
Family
Poppins
Weight
SemiBold
Size
Line height
Letter Spacing
10
120%
2
Name
Find
Replace
Description
Description
Update styles
Delete selected
More information at jansix.at

Figure 2.3 - Updating styles in bulk

There's more...

When browsing any plugin on the Figma community page, you can check the date of its last update. If a plugin hasn't been updated in over two years, it's wise to be cautious about integrating it into your long-term workflow, as it's unlikely to receive further updates. Another useful indicator is the comment section below the plugin, where you can gauge user feedback and recent experiences.

Is This Plugin Paid?

Not everything can be community driven forever.

What's the problem?

Many plugins are developed as passion projects by independent designers or developers. While this contributes to a thriving design community, relying on unsupported plugins for long-term workflows can pose risks. Without a monetization model or sustained company backing, a plugin may become outdated, break, or be abandoned, leaving users without support or updates.

How to fix it?

To ensure reliability, prioritize plugins that have a sustainable business model or corporate backing. A well-maintained plugin is more likely to receive updates, and ongoing support. A great example is Stark, which has a strong commercial foundation and continues to evolve with the needs of its users.

Plugin Example: Stark

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/732603254453395948/stark-contrast-accessibility-checkerI'm happy to see that accessibility is becoming one of the key focuses for designers. One driving factor is the **European Accessibility Act** (**EAA**), which will take effect in June 2025. However, beyond compliance, I truly believe that when given the opportunity, we should aim to design inclusive experiences that benefit everyone.In the past, I used Stark primarily for color contrast checks and vision impairment simulations. Today, the plugin offers even more—features like the Typography Checker and Touch Targets (Area) Check help ensure designs are both readable and user-friendly. These tools also assist developers by defining appropriate click zones, making accessibility a natural part of the design workflow.

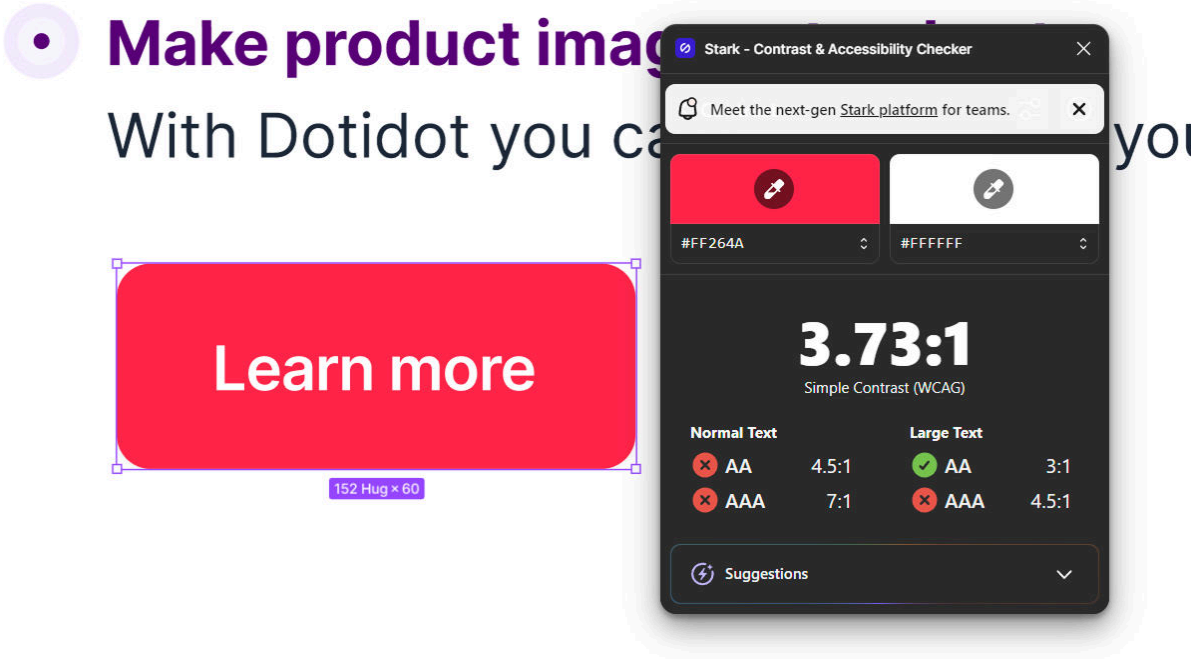


Figure 2.4 - Contrast Checker for selected button and text color.

Testing your designs through vision impairment simulations takes only moments but provides invaluable insights into how your product will be experienced by users with different visual abilities.

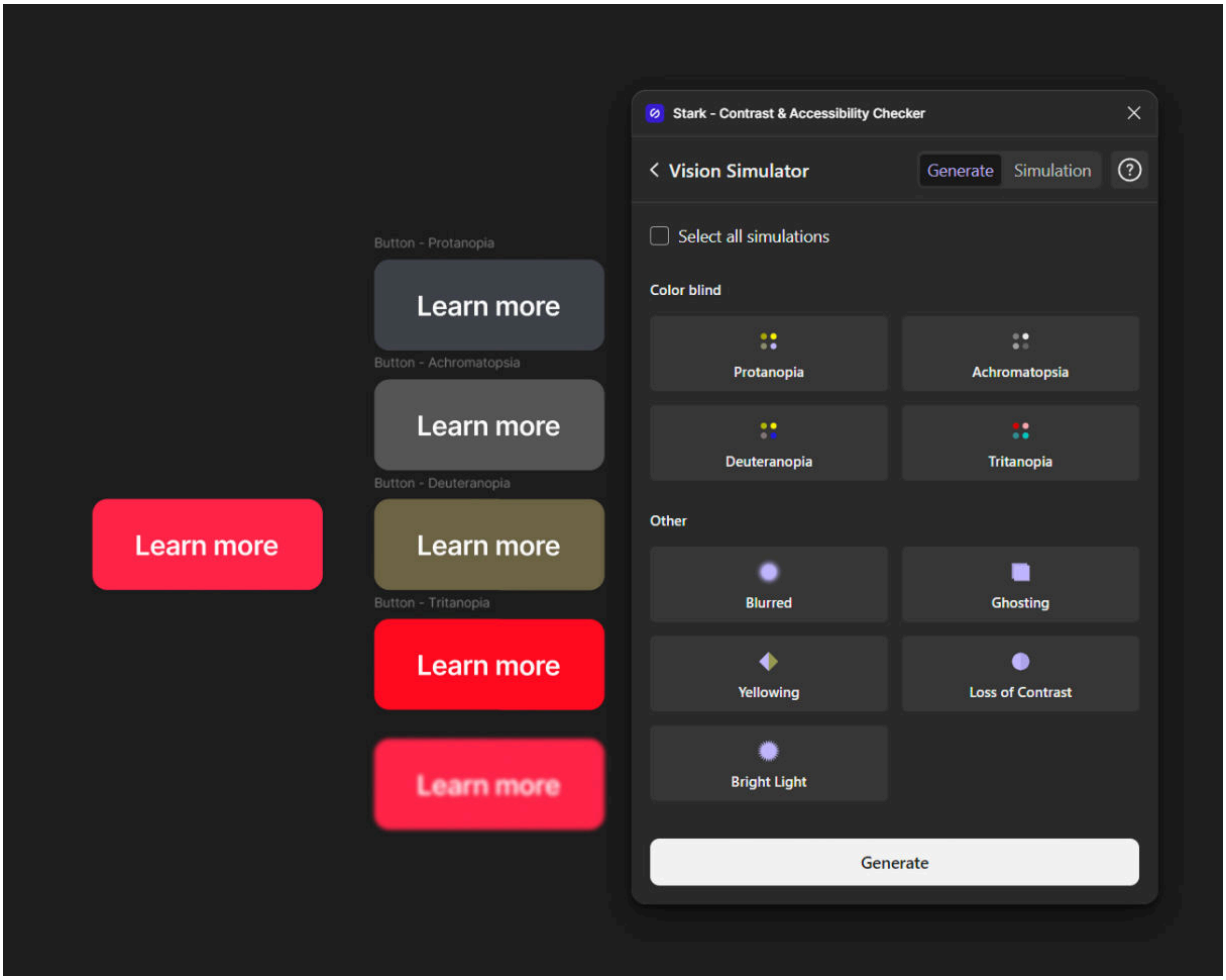


Figure 2.5 - Simulation of different vision disorders for you to see how your design will look.

We should always keep accessibility at the forefront of our design process —it's not just a compliance checkbox but a fundamental principle that ensures our products serve everyone, regardless of their abilities.

Is It Developed by a Company?

The source of a plugin can tell you a lot about its long-term viability. While individual developers create amazing plugins, company-backed tools often have more resources behind them for maintenance and updates.

What's the problem?

Not all plugins are independently developed—some are backed by companies as part of their larger ecosystem. These plugins often come with greater reliability and long-term support compared to those developed by individual creators. A company has a vested interest in maintaining and updating its tools to ensure seamless integration with its core products. While there is always a risk that a plugin's functionality might shift in priority, businesses are generally more accountable for providing consistent updates and support, making them a more stable choice for long-term workflows.

How to fix it?

To mitigate risks, prioritize plugins developed by companies with a vested interest in maintaining them. Plugins like Figma to Webflow or Figma to Framer exist because these companies need smooth integration between their products. If a plugin is backed by a company that actively supports its product ecosystem, there's a higher likelihood it will be maintained. A strong example of this is Jira Connector and Widgets, which is consistently updated and widely used in project management workflows.

Are There Any Reliable Alternatives?

Every tool needs a backup plan. Before committing to a plugin that might become a critical part of your workflow, it's important to know if there are viable alternatives you could switch to if needed.

What's the problem?

When selecting a plugin, it's crucial to consider whether there are reliable alternatives. While many CSV Data to Figma plugins exist, making it easy to switch if one stops working, some plugins lack viable substitutes. If a plugin becomes unsupported, teams relying on it may struggle to find a replacement, causing workflow disruptions and additional development overhead.

How to fix it?

To mitigate risks, always check if alternative plugins exist before committing to one. If a plugin is critical to your workflow and has no direct competitors, consider its long-term sustainability. Assess the developer's track record, update frequency, and user community engagement. For example, Token Studio is a highly complex plugin for managing large design systems, and replacing it would require significant effort. In such cases, ensure you have a contingency plan or explore whether an in-house solution is feasible.

Streamlining Workflow Efficiency with Time-Saving Plugins

Let's face it—design workflows can often get bogged down with repetitive tasks and inefficient handoff processes. The right plugins can transform these pain points into smooth, automated experiences. In this section, I'll share some of my favorite plugins that have saved my team countless hours and significantly improved our collaboration with developers.

What's the problem?

The handoff between design and development is one of the most time-consuming steps in any design project. Miscommunication and missing details can lead to inefficiencies, delays, and errors in implementation. Ensuring that developers receive the correct information is crucial for a smooth transition and to prevent unnecessary revisions.

How to fix it?

Using Figma plugins designed for handoff can streamline this process and improve workflow efficiency. Tools like **Token Studio** help manage design tokens, while **Variables to CSS** and **Variables to JSON** convert Figma Variables into developer-friendly formats. These plugins ensure that developers have the necessary information at their fingertips, reducing friction and accelerating project timelines.

Plugin Example: Token Studio

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/843461159747178978/tokens-studio-for-figmaToken Studio is an essential tool for teams managing and implementing design tokens within Figma. It acts as a bridge between design and development, converting tokens into code-ready formats. If your team handles large-scale design systems, Token Studio is indispensable for maintaining consistency across multiple projects.

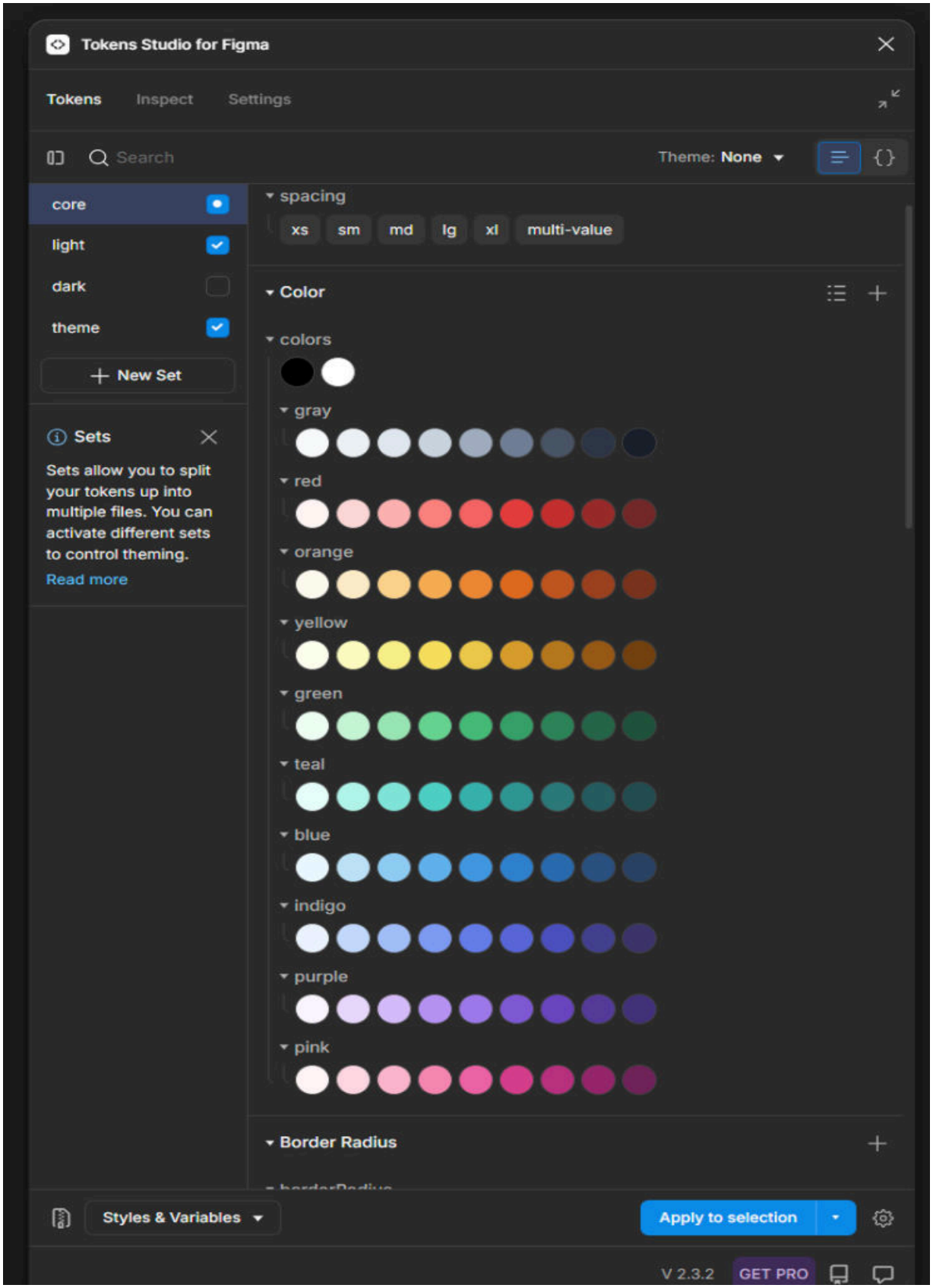

Tokens Studio for Figma
Tokens
Inspect
Settings
Search
Theme: None
core
light
dark
theme
New Set
Sets
Sets allow you to split your tokens up into multiple files. You can activate different sets to control theming.
Read more
spacing
xs
sm
md
lg
xl
multi-value
Color
colors
gray
red
orange
yellow
green
teal
blue
indigo
purple
pink
Border Radius
Styles & Variables
Apply to selection
V 2.3.2
GET PRO

Figure 2.6 - Token Studio showcase in Figma. This is demo project from Token Studio Team. Real structures are much more complex.

Plugin Example: Variables to CSS

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/843461159747178978/tokens-studio-for-figmaFigma variables are powerful, but remember—your final product isn't the Figma file itself, but the production code. While variables in Figma enhance design consistency, they remain useful only if developers can easily access and implement them. Variables to CSS helps bridge this gap by exporting Figma variables into a developer-friendly CSS format, ensuring smooth handoff and better collaboration between designers and developers.In the image below you can see the structured output of our design tokens exported to CSS, demonstrating how Figma variables are transformed into code-ready format for developers.

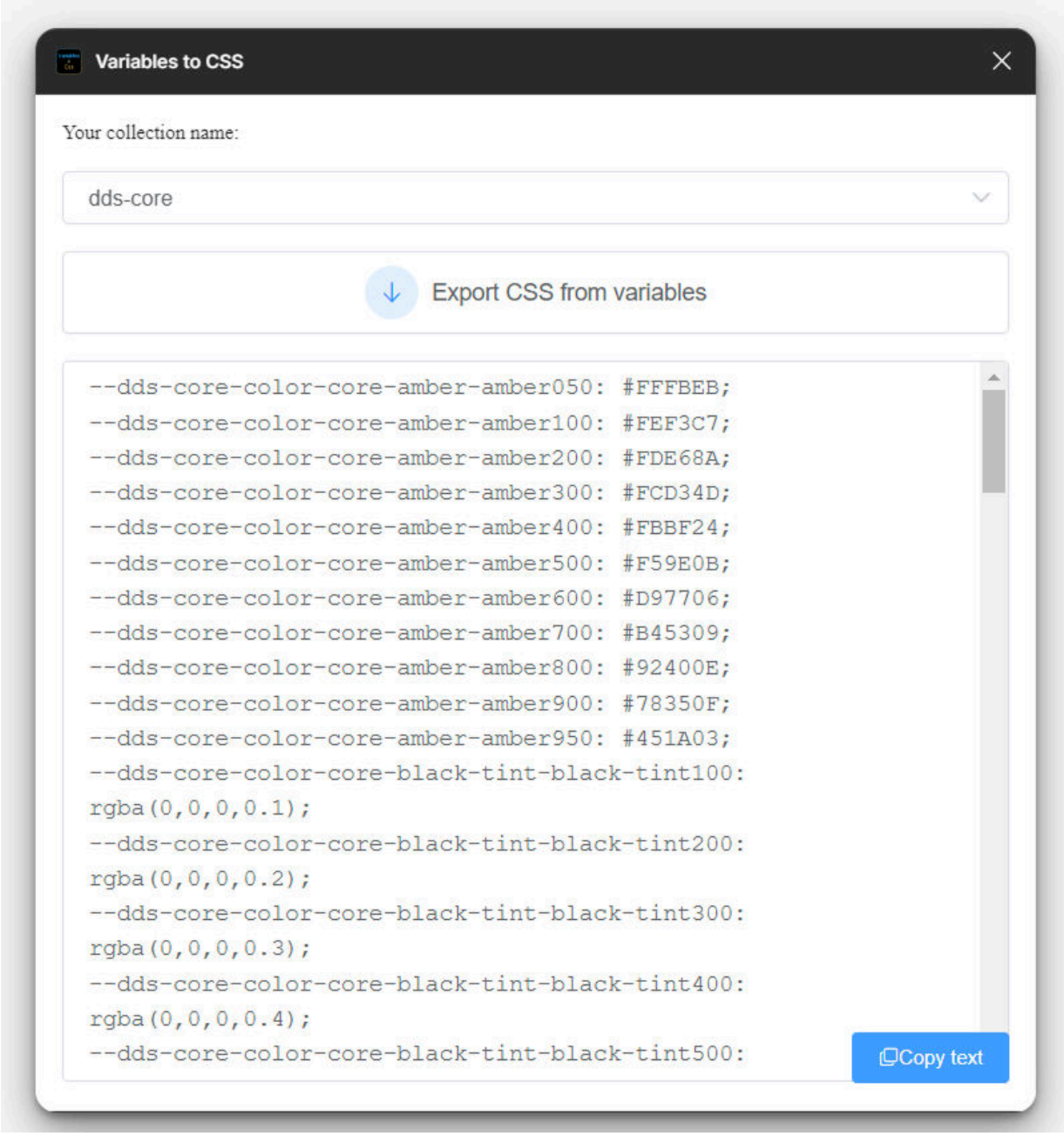


Figure 2.7 - Showcase of exported CSS code from Figma Variables in dotidot design system

Plugin Example: Variables to JSONFigma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/1345399750040406570/figma-variables-to-jsonSimilar to Variables to CSS, this plugin exports Figma variables in a JSON format. The benefit? Developers can easily integrate

design tokens into their workflow and even build custom export pipelines tailored to their project's needs. JSON-based exports provide flexibility, enabling teams to adjust naming conventions, value formats, and integration processes based on specific requirements.Just like in the CSS example above, you can see how Figma Variables maintain their hierarchical structure when exported to JSON. The plugin offers customizable export settings, allowing you to adjust the output format to match your team's specific requirements

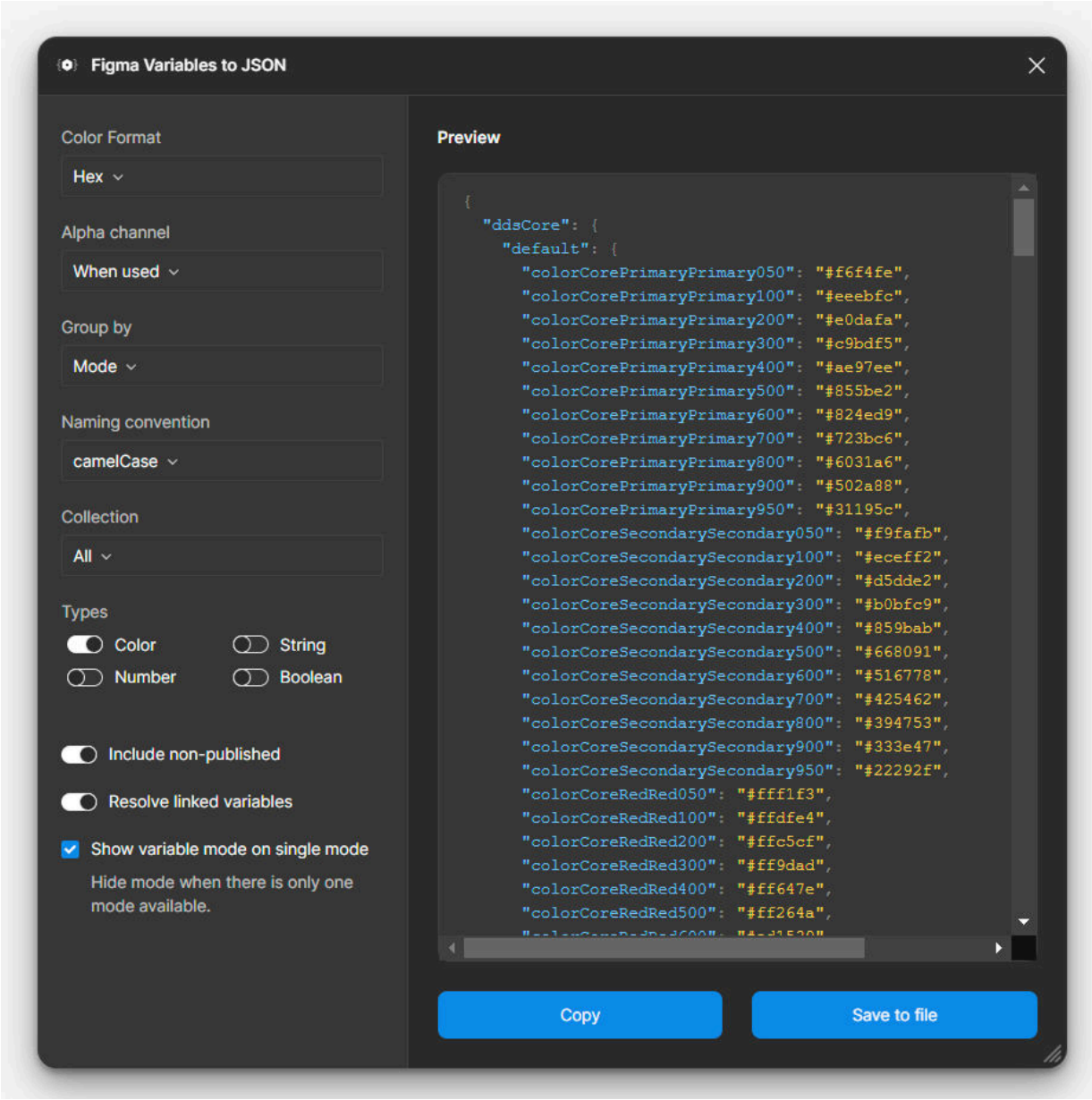


Figure 2.8 - Showcase of exported CSS code from Figma Variables in dotidot design system

Plugin Example: Builder.io

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/747985167520967365/builder-io-ai-powered-figma-to-code-react-vue-tailwind-moreDevelopers in my community highly recommend this AI-powered design-to-code plugin, which can accelerate the development process. It’s not meant to replace developers but rather to enhance their efficiency and speed. If you're a developer, keep an open mind and give it a try—you might find it a valuable addition to your workflow.

Automating Repetitive Tasks with Plugin Integration

Every designer knows the feeling—you're deep in creative flow when suddenly you hit a roadblock of mundane, repetitive tasks. Whether it's populating designs with realistic content, organizing layers, or preparing assets for handoff, these necessary but tedious activities can drain your creative energy. This is where automation plugins truly shine.

What’s the problem?

Designers and developers often find themselves repeating the same tasks, such as documentation, handoff preparation, and populating designs with real content. On larger projects, these tasks become time-consuming and, frankly, tedious. This repetition can slow down workflows, lead to inefficiencies, and take time away from more creative and strategic work.

How to fix it?

I have prepared a few plugins for you that solve completely different problems, but all of them can automate your repetitive work so you will

have more time to focus on what truly matters—solving problems for your users.

Plugin Example: Unsplash Plugin

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/738454987945972471/unsplash
Every design should be presented or reviewed in its final form—essentially how it will appear in production. However, in many cases, final assets, such as images, may not be readily available. Unsplash, one of the largest free image banks in the world, offers a dedicated Figma plugin that simplifies the process. If you're working on a large project with hundreds of images, this can save you valuable time by eliminating the need to switch between Figma and a browser, download images, and manually place them into your design. Instead, you can insert high-quality images directly, allowing you to focus on refining your design rather than handling repetitive asset management tasks.Plugin Example: Content Reel PluginFigma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/731627216655469013/content-reelContent Reel is a powerful plugin that allows you to quickly populate your designs with real-world content. It is especially useful for different types of teams:

- **Agencies/Freelancers**: When presenting designs to clients, realistic content can make a big difference. Content Reel provides instant access to diverse names, surnames, email addresses, phone numbers, country lists, and people's profile pictures. No more generic placeholder text or repeated `user@gmail.com` entries.
- **In-House Teams**: When testing edge cases, Content Reel helps validate design. By inserting real addresses, varied file names, or long pieces of text, designers can stress test components and identify UI issues, such as improper text truncation or input field overflow. This will help you save a huge amount of time in development.

Best of all, you can upload your own datasets and reuse them as needed. If your project requires specific types of content, such as long text in different

languages, Content Reel ensures you can work with relevant information at all times.As shown in the image below, Content Reel provides an extensive library of pre-populated, realistic data categories that you can instantly drag and drop into your designs.

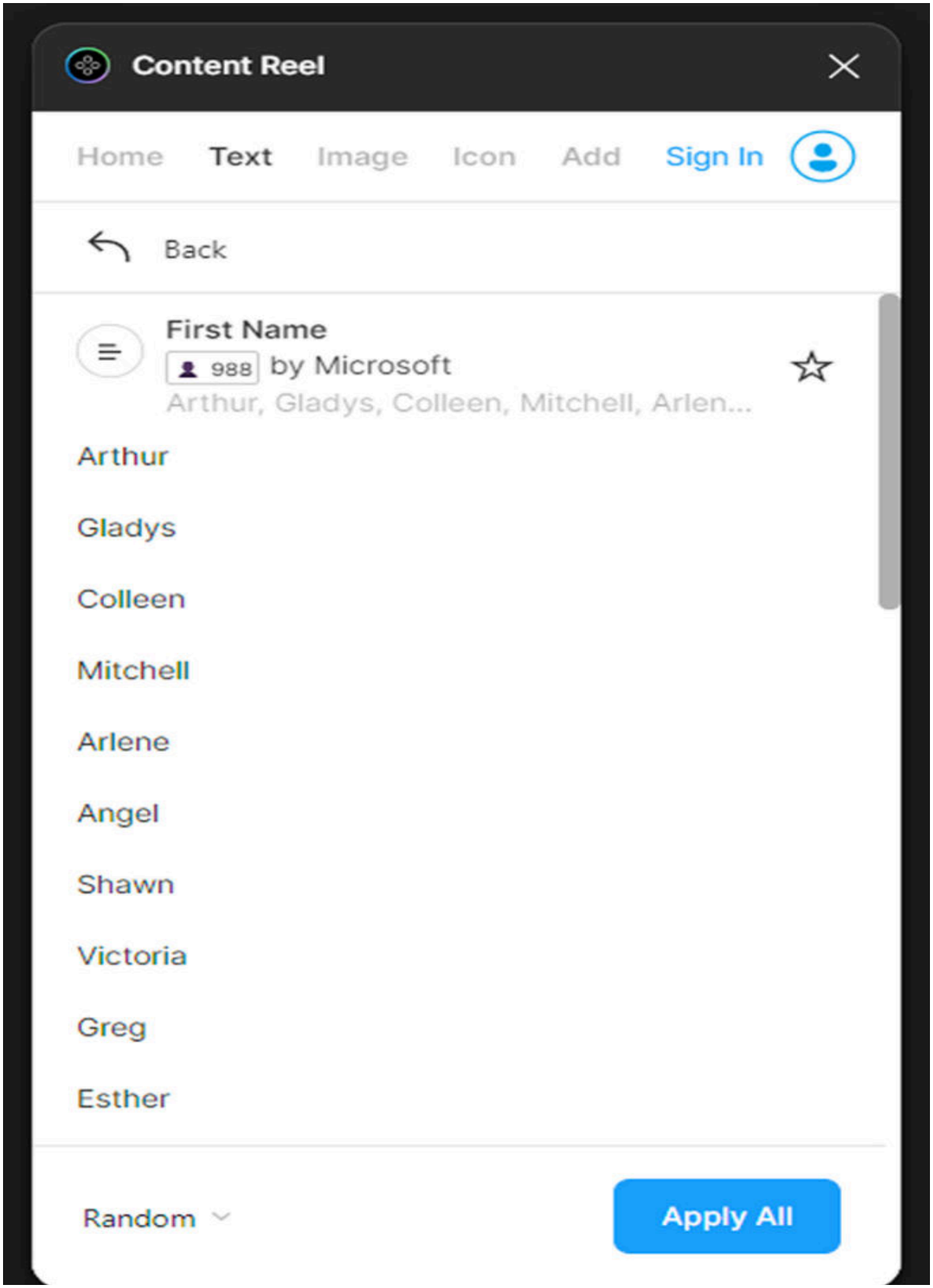

Content Reel
Home
Text
Image
Icon
Add
Sign In
Back
First Name
988
by Microsoft
Arthur, Gladys, Colleen, Mitchell, Arlen...
Arthur
Gladys
Colleen
Mitchell
Arlene
Angel
Shawn
Victoria
Greg
Esther
Random
Apply All

Figure 2.9 - Content Reel example of First Names

Plugin Example: BrandFetchFigma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/733590967040604714/brandfetchWhen working with brand logos, you have two options: store a massive file containing logos from around the world or use **BrandFetch** to fetch them on demand. The choice seems simple, but if you find yourself Googling brand names + logos repeatedly, it quickly becomes frustrating and inefficient. With BrandFetch, you can instantly access multiple versions of logos for major brands, ensuring consistency and saving valuable time in your workflow. One of the standout features is having multiple variations of logos in one place, as demonstrated in the Spotify example below.

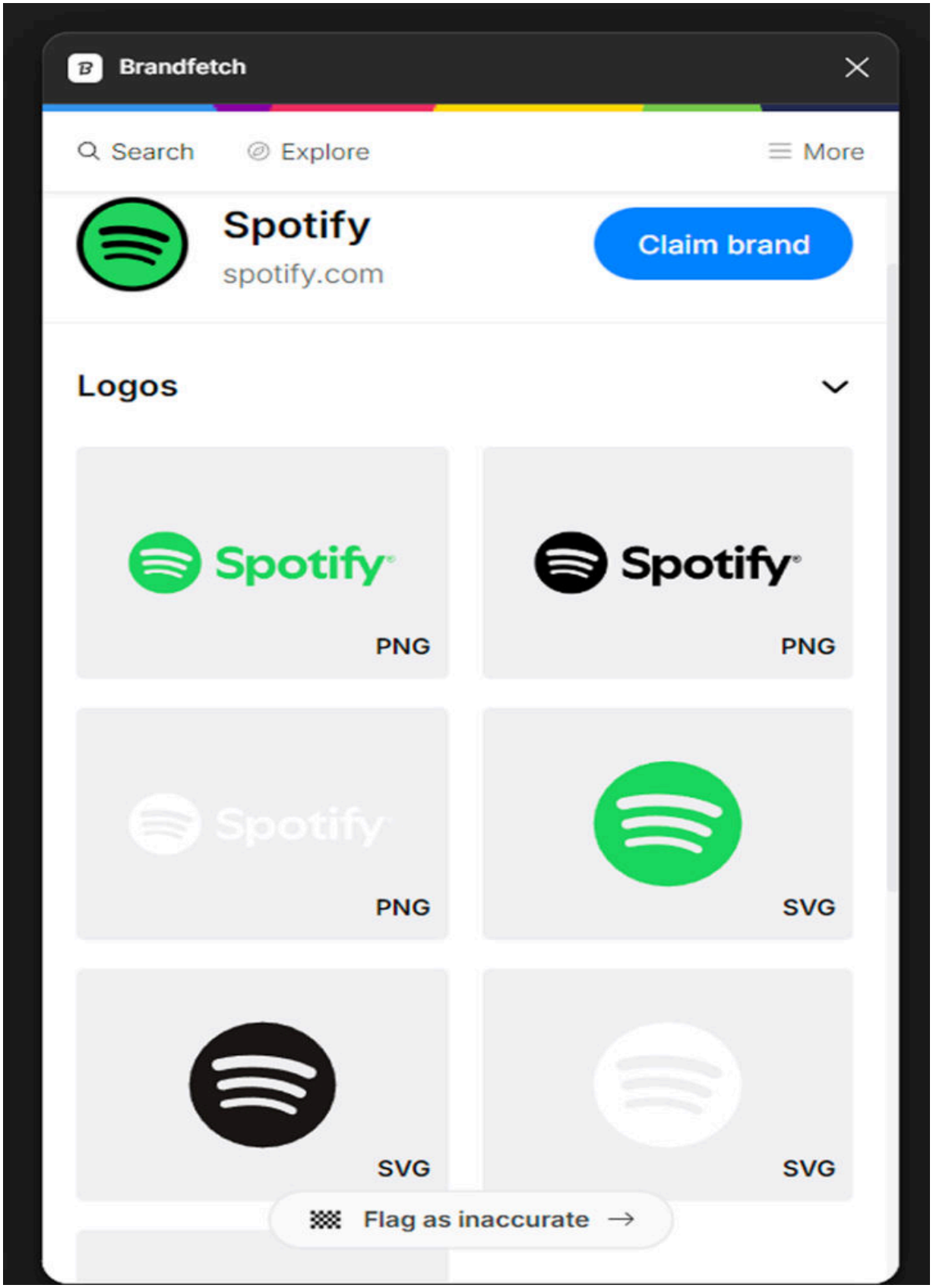

Brandfetch
Search
Explore
More
Spotify
spotify.com
Claim brand
Logos
Spotify
PNG
Spotify
PNG
Spotify
PNG
SVG
SVG
SVG
Flag as inaccurate

Figure 2.10 - All logos of popular brands in one place

Plugin Example: Auto DocumentationFigma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/1134018716847999330/auto-documentationDesign system documentation is essential. This plugin can generate a complete structure for your Figma Variables or Styles in seconds. For large systems with hundreds of tokens, this can save you an entire day —time better spent refining key design decisions rather than manually organizing styles.

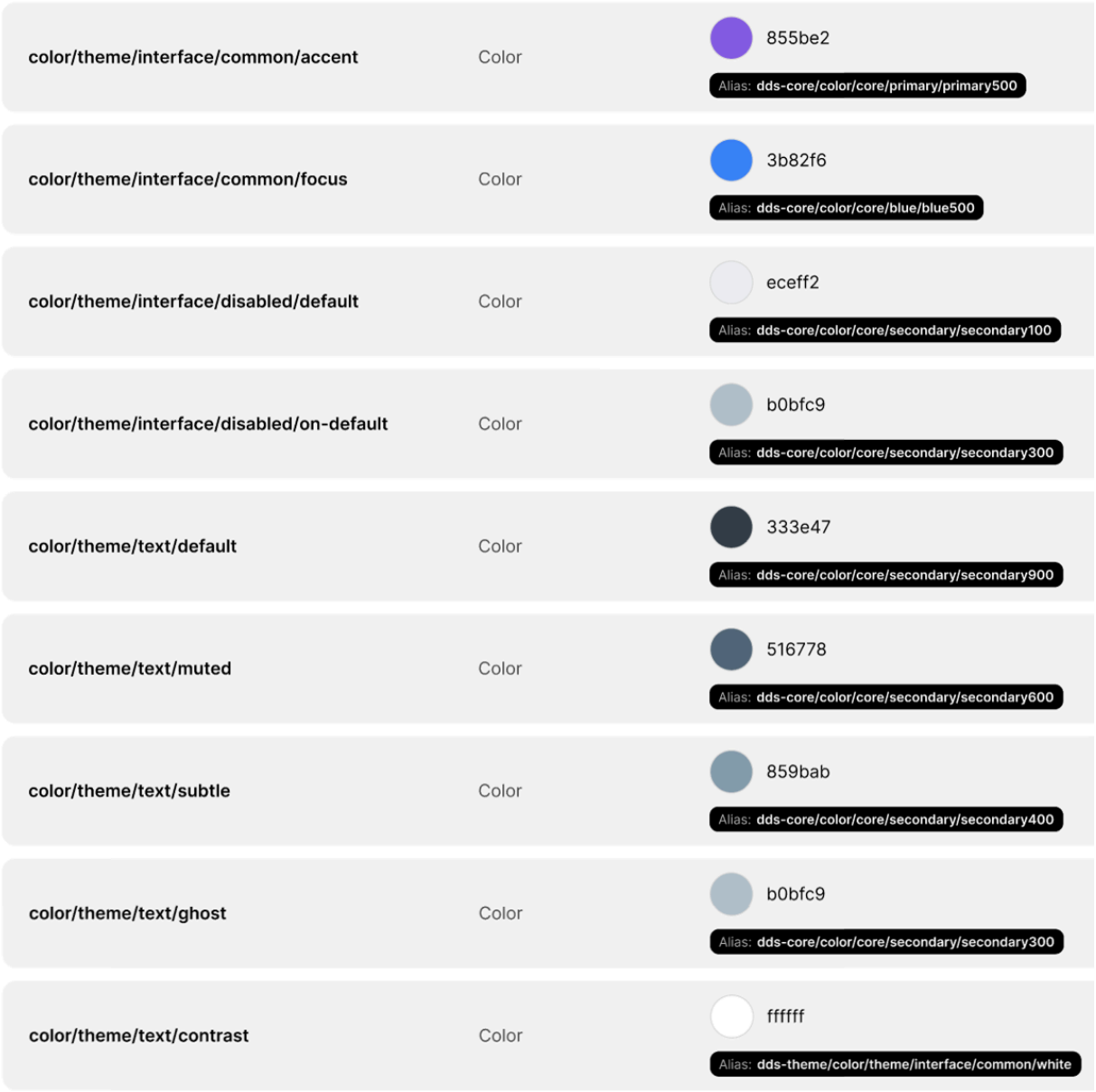


Figure 2.11 - Documentation of color Design Tokens/Variables

Specific Project Needs solved by plugins

Every project comes with its unique challenges and requirements. While standard design tools cover the basics, specialized plugins can be the secret weapon that helps you tackle those project-specific hurdles with ease. Let's explore some plugins that solve particularly niche but important design problems.

What's the problem?

Some projects are highly specialized. I encountered this frequently during my freelancing days. One month, I was working on a large multinational e-commerce store; the next, I was designing for a fintech startup. These projects were exciting, but they often had unique requirements that needed to be solved at the design level. Unfortunately, you can't always copy and paste a solution from one project to another because each has its own constraints and challenges.

How to fix it?

Over time, I discovered several plugins that helped me navigate these unique design challenges. While these tools may be niche, they can be invaluable if you find yourself in a similar situation.

Plugin Example: Charts plugin

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/731451122947612104/chartsDesigning charts can be challenging. If you've worked on complex dashboards before, you know the difficulties involved. Chart generation plugins were among the first I searched for and paid for because, without them, I wouldn't have been able to complete certain projects in the past. Always ensure you discuss charts with developers, as they are unlikely to build them from scratch. Instead, they will rely on existing libraries, so your chosen plugin should align with the functionality of the selected library to ensure seamless integration.

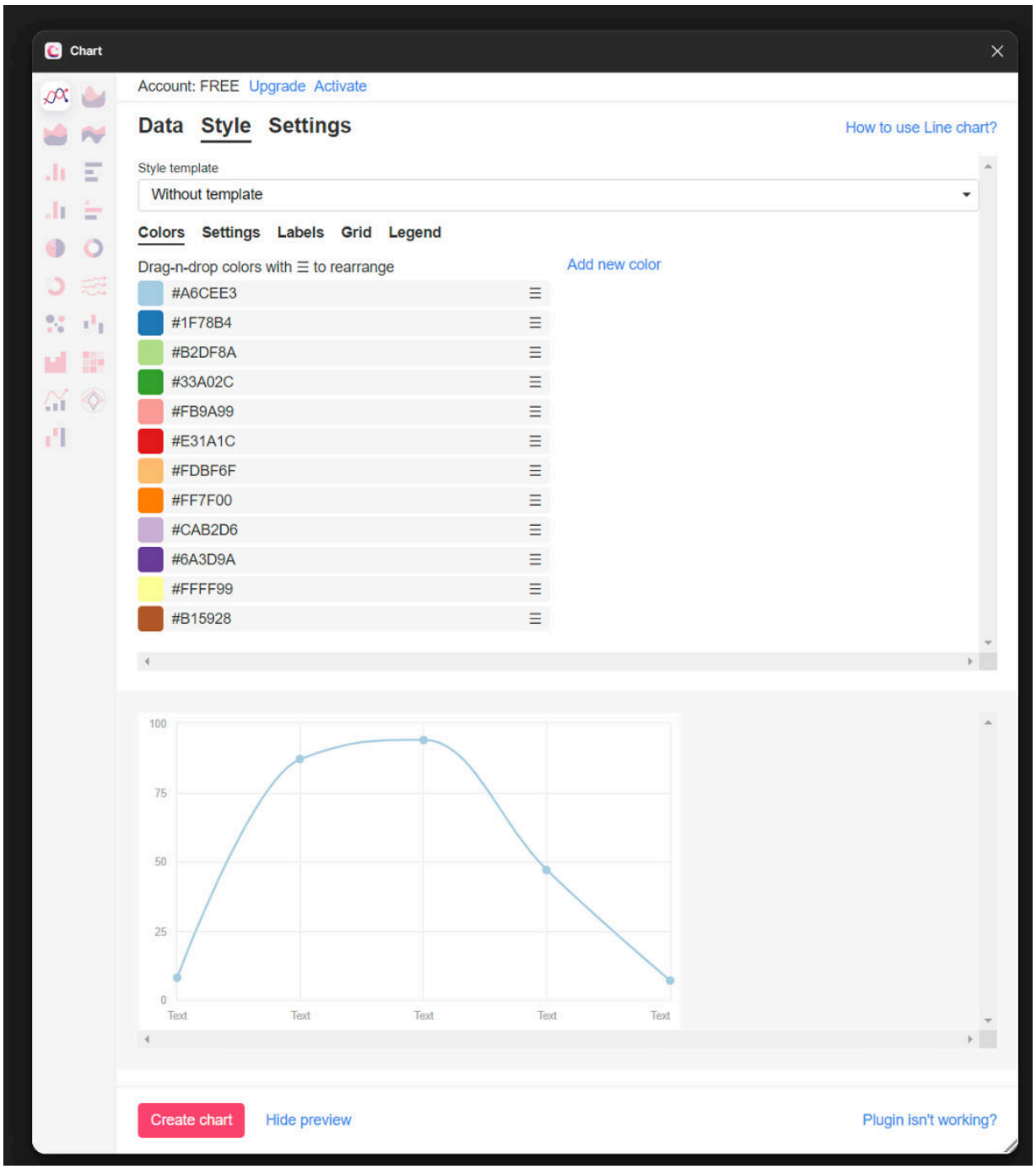


Figure 2.12 - Chart preview and settings before insert into Figma

Plugin Example: Downsize

Figma community plugin page, where you can read more and active the plugin:

https://www.figma.com/community/plugin/869495400795251845/downsize
Large Figma files often contain numerous images, which can be particularly challenging for visually heavy projects like brand websites. Figma's ability to save images in their original resolution provides a great starting point for design work. However, many developers fail to optimize images, simply downloading them from Figma as JPEG or PNG files and using them directly in their projects. This can lead to slow load times due to large asset sizes.Downsize is a one-click solution for image optimization, ensuring that assets are properly compressed before they are exported and used in development. Another significant benefit of this plugin is improving Figma's performance. If your Figma files feel sluggish, optimizing images with Downsize can greatly enhance speed and responsiveness. I've personally found it to be an effective way to streamline workflows and improve project efficiency.

Linking Figma to Other Software for Cross-Platform Integration

Design rarely exists in isolation. In today's interconnected workflow environments, your Figma designs need to communicate seamlessly with other tools in your tech stack. The right integration plugins can eliminate tedious manual transfers and keep your project information synchronized across platforms.

What's the problem?

Figma doesn't exist in a vacuum. It's part of a broader workflow involving multiple tools to transform designs into production-ready code. Through my mentoring sessions, I've often encountered teams that are highly skilled in Figma but struggle to integrate it seamlessly with their other tools, leading to inefficiencies in their workflow.

How to fix it?

Take a step back and analyze your daily workflow. Identify the tools you and your team rely on and explore plugins that bridge the gap between Figma and those platforms. I will highlight some of the ones I use, but depending on your stack, you may find alternatives that better fit your needs. Since every team's toolset is unique, the key is finding integrations that streamline your specific workflow.

Plugin Example: Jira

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/1220802563996996107/jiraEasily integrate Jira issues in real-time with your Figma file. If your team relies on Jira, this plugin makes it much easier for designers, developers, and product managers to check specifications directly within Figma. Since this is an official plugin, you can trust its reliability and long-term support, ensuring a smooth workflow for your team.

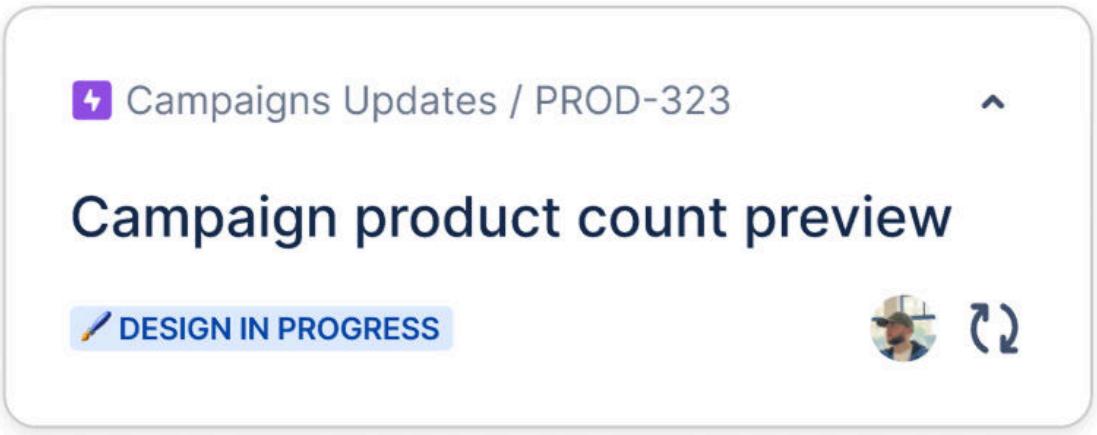


Figure 2.13 - Jira widget in Figma

Plugin Example: Asana

Figma community plugin page, where you can read more and active the plugin:

https://www.figma.com/community/widget/1098405969270214551/asanaSimilar to the Jira integration, this official plugin connects Asana tasks in real-time with your Figma file. It streamlines collaboration by allowing designers, developers, and product managers to access task details directly within Figma, reducing the need to switch between tools.

Plugin Example: GitHub

Figma community plugin page, where you can read more and active the plugin:
https://www.figma.com/community/plugin/1220512233196109878/githubSeamlessly integrate GitHub issues with your design workflow. This plugin is particularly useful for technical designers and developers, enabling them to access issues directly within Figma. By consolidating everything in one place, it enhances efficiency and eliminates the need to switch between platforms.

Plugin Example: Figma to Weblfow

Figma community plugin page, where you can read more and active the plugin:
https://www.figma.com/community/plugin/1164923964214525039/figma-to-webflow-html-css-and-websiteWebflow is a low-code/no-code website-building platform that allows marketing teams to move faster without relying on developers. Many major brands use Webflow for its flexibility and ease of use, but it’s also a great choice for personal projects. Having used Webflow alongside Figma for years, I was excited about this integration.The plugin had a rough start, so if you tried it before and dismissed it—like I did—it's worth giving it another shot. Now, you can sync Variables, Styles, and even entire components effortlessly. If you’re building on Webflow, grab a coffee and try it out!

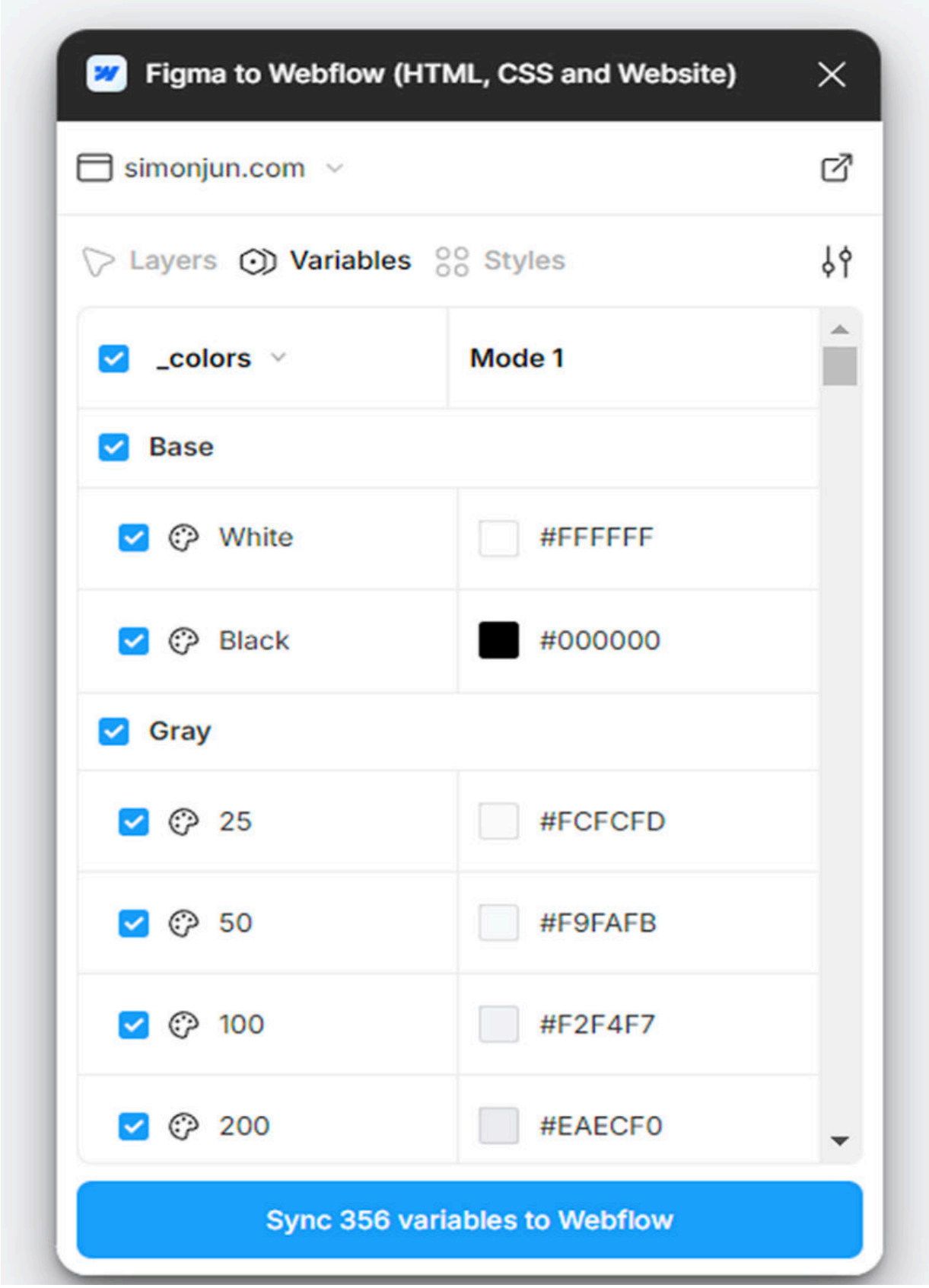

Figma to Webflow (HTML, CSS and Website)
simonjun.com
Layers
Variables
Styles
_colors
Mode 1
Base
White
#FFFFFF
Black
#000000
Gray
25
#FCFCFD
50
#F9FAFB
100
#F2F4F7
200
#EAECF0
Sync 356 variables to Webflow

Figure 2.14 - Figma to Webflow Variables Sync

Plugin Example: Figma to Framer

Figma community plugin page, where you can read more and active the plugin: https://www.figma.com/community/plugin/1037108608720448600/figma-to-html-with-framerSimilar to the Figma-to-Webflow integration, this plugin connects Figma with Framer. While I don't personally use Framer, I reached out to my community, and the feedback has been overwhelmingly positive. Given its strong reception, I wanted to highlight it as a valuable option for those working with Framer.

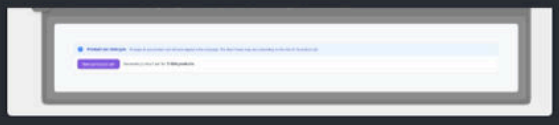
There's more

The last point I want to highlight is Figma's integration with other tools. Figma has its own plugins within external platforms, enhancing connectivity and streamlining workflows. Two key use cases I rely on daily are Google Workspace and Jira, where these integrations make it much easier to link designs with documents or tickets, ensuring seamless collaboration across teams.

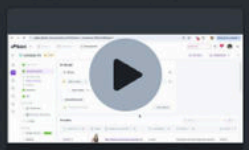
Figure 2.15 - In all Google Documents, you will see this Figma FIle Name pills instead of the whole link.

The image below demonstrates how seamlessly you can embed Figma into Jira tickets, allowing you to access designs directly without opening Figma separately.

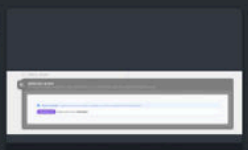
PROD-378 / PROD-329



Attachments 4

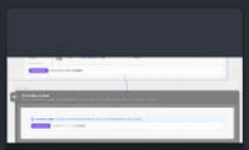


Screen Recordin... .05.mov
28 Jan 2025, 03:09 PM



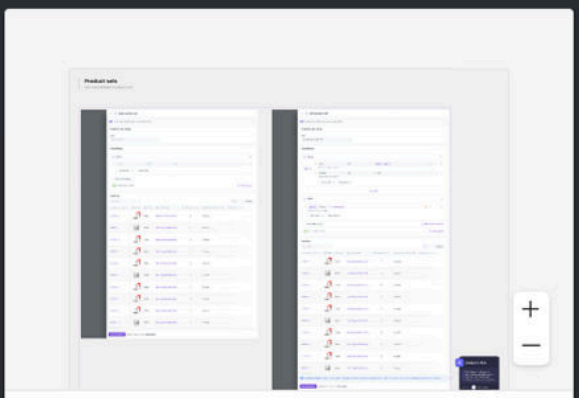
Screenshot 2025... .51.png
28 Jan 2025, 02:57 PM

CleanShot 2024-... 218.png
18 Dec 2024, 12:22 AM



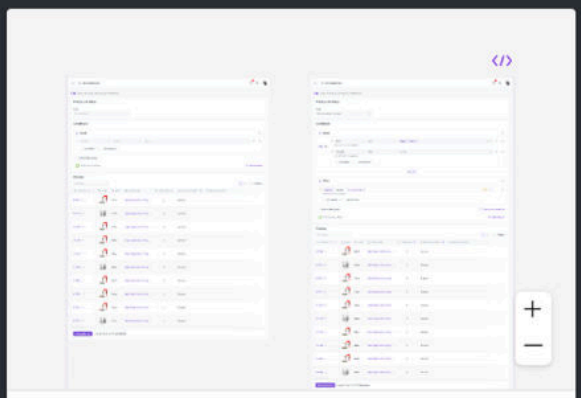
CleanShot 2024-... 139.png
18 Dec 2024, 12:22 AM

Designs (2)



06 [DDS] Templates - M... · Produ... Edited 11 days ago

Expand in Jira



05 [DDS] Temp... · [TEMPLATE]-pr... Edited 7 days ago

Expand in Jira

Activity

Show: All Comments History Worklogs Zendesk Support Newest first

Add a comment...

Looks good! Need help? This is blocked... Can you clarify...? This i

Pro tip: press **M** to comment

Figure 2.16 – An overview of Jira ticket preview of the Figma right inside the Jira. No need to open up Figma.

When to Consider Writing Your Own Plugin

Sometimes, despite the vast plugin marketplace, you might find yourself thinking, "I wish there was a plugin that could..." That might be your cue to consider creating a custom solution. While it may sound intimidating, developing your own Figma plugin can be more accessible than you might think—and it could be the perfect answer to your specific workflow challenges.

What's the problem?

Sometimes, the available plugins on the market just don't meet your needs. What's the solution? Build your own! Before you turn the page, hear me out—I know most of you are designers, but many of my designer friends have successfully created their own plugins. Even my younger brother built one while still in high school!

How to fix it?

While building your own plugin is one option, remember that you don't need to be a developer to create one. I've worked with several companies where designers approached plugin creation as simply another design task. They created the specifications and designs, then someone from the development team built it during their regular sprints, treating it like any other ticket.If you want to create something simple, don't be afraid to try building it yourself. However, if you need something more complex—particularly if it needs to connect to your databases or other systems—don't hesitate to ask your development team for help. Figma provides excellent documentation, making it relatively straightforward for developers to implement your ideas efficiently.This book isn't about plugin development, so I won't dive too deep into the technical details. But don't worry—it's easier than you think! Grab a coffee, fire up ChatGPT, and check out the

Figma tutorial below. You've got this!https://help.figma.com/hc/en-us/articles/4407260620823--BYFP-1-Overview.

3 Harnessing AI in Figma and Beyond

AI is integrated into almost every tool nowadays—if a tool doesn’t have AI, some might jokingly say it's already obsolete. Over the past few years, nearly every software has jumped on the AI bandwagon, often implementing features that feel experimental or lack a clear use case. In this chapter, I will explore Figma’s AI capabilities and how I leverage other AI tools to enhance my productivity in Figma. We will cover these six key topics:

- Exploring Figma's Built-in AI Features
- Automating Routine Tasks with AI in Figma
- Integrating AI Tools for Faster Prototyping
- Implementing AI into Your Design Workflow
- Analyzing AI-Enhanced Design Workflows: Case Studies
- Navigating Ethical Challenges in AI-Driven Design

Exploring Figma's Built-in AI Features

Figma includes several AI features, each aimed at addressing different challenges. We’ll begin with the most useful ones before discussing those that feel less refined. Keep in mind that my perspectives are based on extensive experience using Figma in various teams. If you’re a lighter user, some of these features might be more beneficial for you, as your expectations may differ. Figma categorizes its AI features into three groups:

- Design Tools
- Riffing and Writing
- Image Editing

Disclaimer

> All insights into Figma’s built-in AI features are based on my experience at the time of writing—mid-2025. If I express skepticism about a feature or state that it doesn’t work well, the real question isn’t *"if"* it will work, but *"when."* If you're reading this later, check again—because right now, these features are at their least developed, and tomorrow they will be better, and the day after that, even better.

Design Tools

Design tools have the most AI features, so we will start with them.

- Rename layers
- Search with image or selection
- Add interactions
- Replace content
- First Draft

Rename layers

Let's start with a seemingly small feature that delivers massive impact in collaborative design environments. Layer naming might not be the most exciting topic, but any designer who's inherited a messy Figma file knows how crucial proper organization becomes during complex projects.

What’s the problem?

One of the biggest challenges in Figma collaboration is dealing with disorganized and poorly named layers. Many designers skip proper layer naming, leading to chaos in large team projects. Imagine a file filled with labels like `Frame236472` —no one wants to waste time deciphering that. When multiple people collaborate on a project, unclear naming makes handoffs to developers and other stakeholders frustrating. Even solo designers need structured files because Figma designs often move into development or marketing assets.

How to fix it?

Figma’s Rename Layers AI feature automates this process, making files easier to navigate and reducing errors. Here’s how you can use it effectively:

1. **Select Multiple Layers**: Choose all the layers in your design that need renaming.
2. **Run the AI Rename Tool**: Let Figma analyze the structure and suggest logical names.
3. **Review and Adjust**: While AI-generated names are often helpful, a quick review ensures they follow your team’s conventions.
4. **Adopt Naming Standards**: If working in a team, define clear naming guidelines so AI-generated names remain consistent.

Figure 3.1 - Same component with right naming (right) and wrong naming (left(

Properly named layers improve workflow efficiency and enhance other Figma features like **Smart Animate** and **Select Matching Layers**, making interactions smoother and troubleshooting easier.

Search with Image or Selection

This feature might be one of my favorites that I use daily. Finding specific elements in large design systems used to be a major workflow bottleneck, but Figma's AI-powered search has transformed how we locate and reuse components across projects.

What's the problem?

Searching for specific elements across multiple Figma files can be incredibly time-consuming, especially when working on large-scale projects with extensive design libraries. Designers often waste valuable time scrolling through files or sending Slack messages asking for the correct link. This inefficiency disrupts workflow and slows down production. When dealing with complex design systems, finding the right asset at the right time is critical for maintaining consistency and efficiency.

How to fix it?

Figma's Search with Image or Selection feature simplifies asset retrieval by allowing users to find elements using a screenshot or selection. Here's how to leverage it effectively:

1. **Capture a Screenshot**: If you come across an element in another project or on a live website/app, take a quick screenshot.
2. **Paste or Select in Figma**: Upload the screenshot or choose an element directly within Figma.
3. **Run the AI Search Tool**: Figma scans its database and returns matching assets, streamlining the search process.

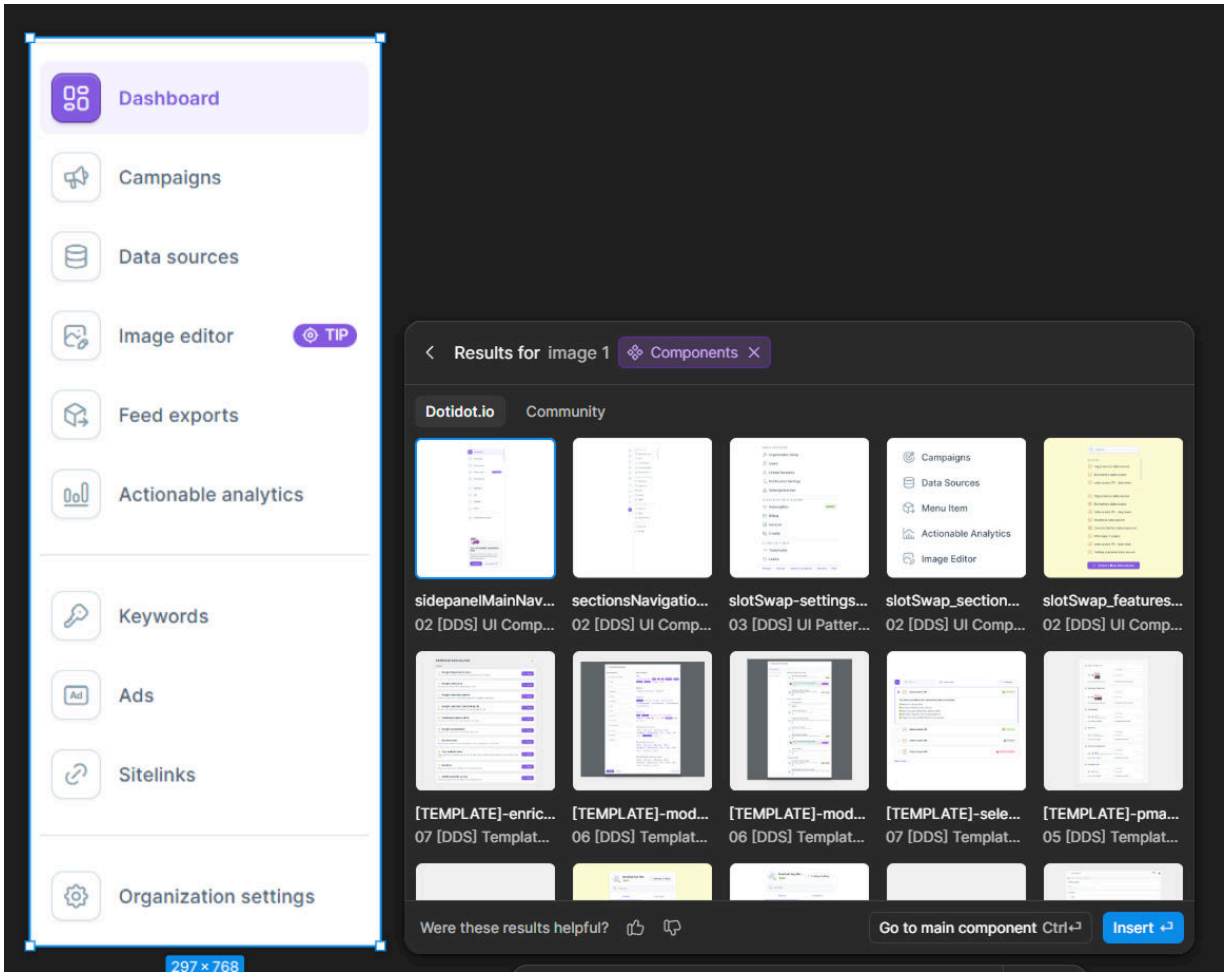


Figure 3.2 - Simple screenshot of Dotidot navigation from your live app, and Figma finds me the right components in seconds

For example, if I need to locate the navigation design from our Dotidot app, I can simply screenshot the live app, paste it into Figma, and instantly retrieve the relevant component. This feature works reliably in nearly every case, making it an invaluable tool for managing complex design systems.

Add interactions

Creating prototypes manually can be tedious, especially for complex flows with multiple states and transitions. Figma's AI-powered interaction builder aims to streamline this process, though its success varies depending on the complexity of your design.

What's the problem?

Creating a complex prototype for user testing or showcasing can be a time-consuming and error-prone process. Manually setting up interactions between multiple screens requires careful planning and execution. While AI-driven prototyping aims to automate this process, its effectiveness depends heavily on understanding the design's intent. For simple user flows, AI-generated interactions work well, but for complex scenarios involving multiple user journeys, AI often falls short. Designers may find themselves spending more time fixing auto-generated interactions rather than benefiting from automation.

How to fix it?

Figma's AI-powered interaction tool can assist in automating the prototyping process. Here's how to make the most of it:

1. **Start with a Clear Flow**: Define the user journey (positions of your frames on the artboard) beforehand to guide AI in setting up logical interactions.
2. **Enable AI-Powered Interactions**: Allow Figma to auto-generate connections between frames based on common user behaviors.
3. **Test the Prototype**: Run the AI-generated interactions to check for accuracy.
4. **Refine and Adjust**: Manually tweak the interactions to ensure they align with your intended user experience.

For simple interactions, this tool can save significant time. However, for more intricate workflows, manual refinement is often necessary to achieve the best results.

Replace content

Populating designs with realistic content is essential for accurate testing and client presentations. Figma's Replace Content feature attempts to automate this often repetitive task, reducing the time spent manually inserting placeholder text and images.

What's the problem?

When working with long lists, tables, or repetitive components in Figma, manually inserting diverse content can be tedious and time-consuming. Designers often struggle with maintaining variety in placeholders while ensuring a realistic representation of the final product. Without automation, this process can slow down workflows and lead to inconsistencies in design.

How to fix it?

Figma's **Replace Content** feature automates content population, helping designers quickly fill repetitive structures with varied text and images. Here's how to use it effectively:

1. **Prepare Your Layout**: Ensure you are using **Auto Layout** with properly structured (ideally components).
2. **Select Multiple Items**: Highlight the elements you want to populate with different content.
3. **Use the Replace Content Feature**: Let Figma's AI generate varied entries for your design.
4. **Review and Refine**: Adjust or swap out any content that doesn't align with your intended output.

While this feature can speed up initial design iterations, it currently lacks contextual awareness, leading to less-than-ideal results. For more control and flexibility, dedicated plugins such as **data.to.design** or **Content Reel** (covered in *Chapter 2*) provide more reliable and customizable content generation options.

First Draft

The promise of generating entire interfaces with a simple text prompt is enticing. Figma's First Draft feature attempts to turn this vision into reality, though as you'll see, the results don't quite match the ambition—at least not yet.

What's the problem?

Generating a complete UI from a simple text prompt sounds like an incredible time-saver. In theory, this feature should allow designers to move from a blank canvas to a fully designed interface within seconds. However, after extensive testing in my online course, I found that the results were repetitive and lacked real usability. Regardless of the input prompt—whether "Budget app," "Bank app," or "Investment app"—the AI consistently produced nearly identical UI structures, heavily featuring cryptocurrency elements like Bitcoin. This suggests that the AI is primarily trained on freely available Figma designs, following outdated trends rather than user needs.While the idea is promising, its current execution is not practical for professional design work. The generated outputs are more suitable for quick visual placeholders than for fully functional products. Designing well-thought-out solutions involves more than arranging UI components—it requires user research, problem-solving, and contextual understanding, which AI-generated designs currently fail to deliver.

How to Do It

While Figma's First Draft feature is not yet refined for professional workflows, here's how you can experiment with it effectively:

1. **Use it for Ideation**: Treat AI-generated designs as rough starting points rather than final products.
2. **Refine the Structure**: Adjust layouts, replace irrelevant elements, and tailor components to your specific project needs.
3. **Combine it with Manual Design Work**: Leverage AI for inspiration, but rely on human insight for usability and UX improvements.
4. **Supplement with Plugins**: If you need structured templates, consider using curated design libraries instead of relying solely on AI-generated content.

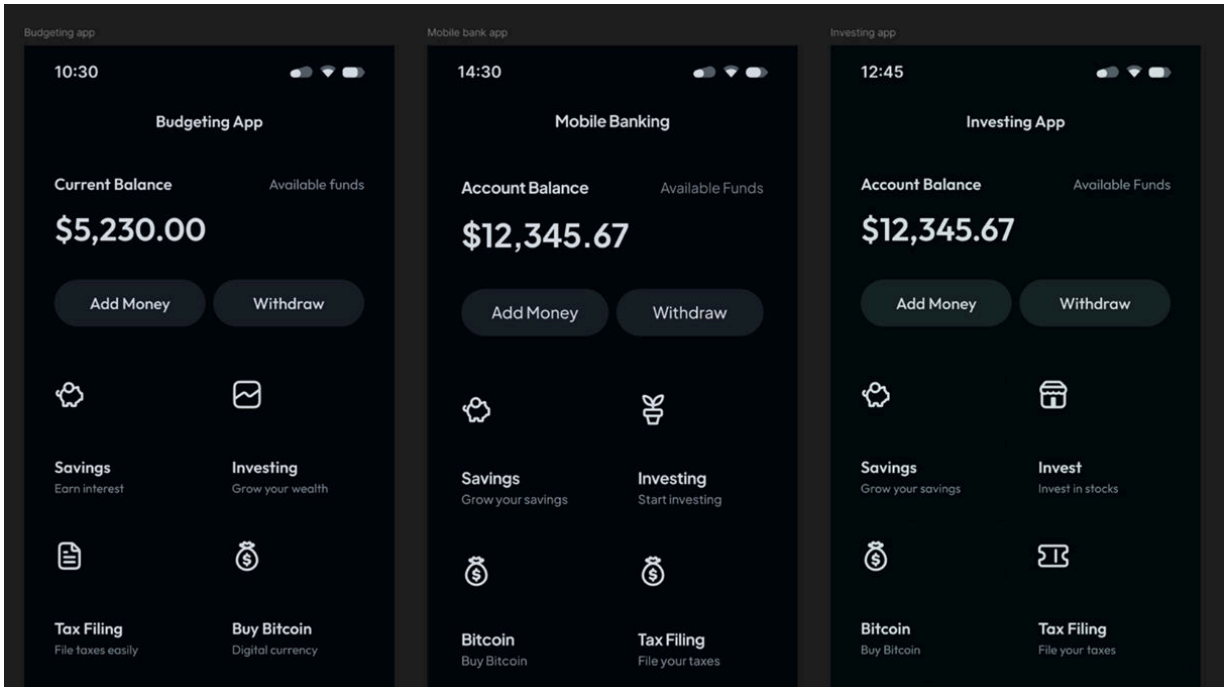


Figure 3.3 - Three different prompts, three same results.

For now, this feature serves as a brainstorming tool rather than a reliable design assistant. However, as AI evolves, it could become a valuable asset for accelerating initial design drafts while still requiring human refinement.

Riffing and Writing

Text is an essential part of every design. Thankfully, the days of relying on Lorem Ipsum are over, and AI has become a powerful tool in this field. Figma offers two features to enhance text management in design:

- Rewrite this
- Shorten
- Translate to

Rewrite this

Text quality can make or break a design. Figma's AI text rewriting capability helps designers improve copy directly within their workflow,

without needing to switch between different tools or wait for copywriter feedback on every small text element.

What's the problem?

Ensuring consistent, high-quality text across a design can be challenging, especially when working with multiple stakeholders or large projects. Designers often struggle with refining placeholder text into something more meaningful while maintaining tone, clarity, and branding consistency. Without an efficient way to edit and optimize content within Figma, teams risk spending unnecessary time rewriting copy manually.

How to fix it?

Figma's **Rewrite This** feature simplifies the process by automatically refining and enhancing text. Here's how to make the most of it:

1. **Insert Placeholder Text**: Begin with a rough draft of the text you want to improve.
2. **Select the Rewrite Option**: Let AI analyze and generate a more polished version.
3. **Adjust for Tone and Context**: Review and tweak the suggested copy to match the specific brand voice and project requirements.
4. **Use External AI for Advanced Editing**: For complex projects, integrate tools like ChatGPT to train AI with specific guidelines such as target audience, tone, and restricted words.

Additionally, ChatGPT works exceptionally well with design screenshots—simply take a snapshot of the design, provide context with placeholder text, and let AI generate refined content in seconds. This feature enhances productivity by reducing the manual effort required to refine design copy.

Shorten

Designing for multiple languages presents unique challenges, particularly when it comes to UI space constraints. This AI feature helps tackle text

expansion issues that often arise in multilingual designs, especially for languages that typically require more characters than English.

What's the problem?

When designing for multiple languages, text expansion can quickly become an issue, especially in languages like Hungarian, where words tend to be significantly longer than their English counterparts. This can cause layout breakages, misaligned elements, and readability issues. Without a proper way to test and adjust text dynamically, designers often struggle to ensure UI consistency across different languages.For example, Hungarian text often requires more space, so shortening key labels can prevent UI misalignment. This feature is a valuable tool for ensuring multilingual designs remain visually consistent without excessive manual adjustments.

English	Hungarian
Login	Bejelentkezés
Accept all	Összes elfogadása

Table 3.1 - Showcase of various lengths of commonly used texts

How to fix it?

Figma's Shorten feature helps solve this problem by providing more concise text alternatives while maintaining clarity. Here's how to use it effectively:

1. **Identify Problematic Text**: Check elements where text expansion may cause layout issues.
2. **Use the Shorten Feature**: Let AI generate a more compact version of the text while preserving meaning.
3. **Compare and Adjust**: Review the shortened text and ensure it fits within your design constraints.
4. **Test Across Languages**: If designing for multiple languages, compare different versions to confirm readability and alignment.

Expanding your product to global markets requires effective localization tools. Figma's translation feature aims to integrate language conversion directly into the design process, potentially saving rounds of back-and-forth with translation teams during the prototyping phase.

What's the problem?

Multilingual design is a necessity for many digital products, but ensuring accurate translations within the design workflow can be a challenge. While Figma's Translate to feature aims to simplify this process, its current limitations make it less effective. Not all languages are supported, meaning designers working with languages like some commonly used languages are still missing, limiting its usefulness for teams designing for global audiences.

How to fix it?

Despite its limitations, the Translate to feature can still be useful in streamlining multilingual design. Here's how to maximize its effectiveness:

1. **Check Language Availability**: Before relying on this feature, verify if your target language is supported.
2. **Use It for Common Languages**: If your project includes widely used languages, the AI-generated translations can speed up the initial localization process.
3. **Integrate External Tools**: For unsupported languages, consider using translation APIs or dedicated localization tools alongside Figma to ensure full coverage. Some plugins can help you with that.

While Figma's translation capabilities are promising, expanding language support would greatly improve its usefulness for designers working on truly international products.

Image Editing

The last group of built-in AI features focuses on image editing. There are three in total—two that I find incredibly useful and one that frustrates me.

- Remove background
- Boost resolution
- Make an Image

Remove background

Image editing was traditionally a task that required switching to dedicated software. Figma's background removal tool brings this essential capability directly into the design environment, streamlining what was once a multi-step, multi-tool process.

What's the problem?

Removing backgrounds from images has always been a tedious task for designers. Before Figma introduced this feature, many relied on third-party tools like **remove.bg** to eliminate backgrounds with a single click. While effective, these external solutions required extra steps, disrupting workflow efficiency. Designers needed a seamless, built-in way to remove backgrounds without leaving Figma.

How to fix it?

Figma's **Remove Background** feature offers a quick and efficient way to isolate subjects from their backgrounds directly within the design tool. Here's how to use it effectively:

1. **Select an Image**: Click on the image you want to edit.
2. **Apply the Remove Background Tool**: Use Figma's built-in AI to extract the subject.

Figure 3.4 - Showcase of Figma background removal

Leverage this feature for product images, marketing banners, or UI elements where background removal enhances the design. This streamlines workflows by eliminating the need for external tools. If you haven't tried it yet, I highly recommend giving it a go!

Boost resolution

Image quality issues are common when working with client-provided assets or legacy materials. This AI enhancement feature allows designers to improve low-resolution images right within Figma, eliminating the need for external image editing tools.

What's the problem?

Working with low-resolution assets has always been a major challenge for me when I was freelancing. Clients often insist on using outdated images or low-quality icons that look pixelated and unprofessional on modern high-resolution displays. Manually enhancing these assets was time-consuming, requiring external tools and meticulous adjustments to maintain quality without distortion.

How to fix it?

Figma’s **Boost Resolution** feature provides a seamless way to upscale images while preserving clarity. Here’s how to make the most of it:

1. **Select the Image**: Choose the low-resolution asset you need to enhance.
2. **Apply the Boost Resolution Tool**: Let Figma’s AI intelligently upscale the image.

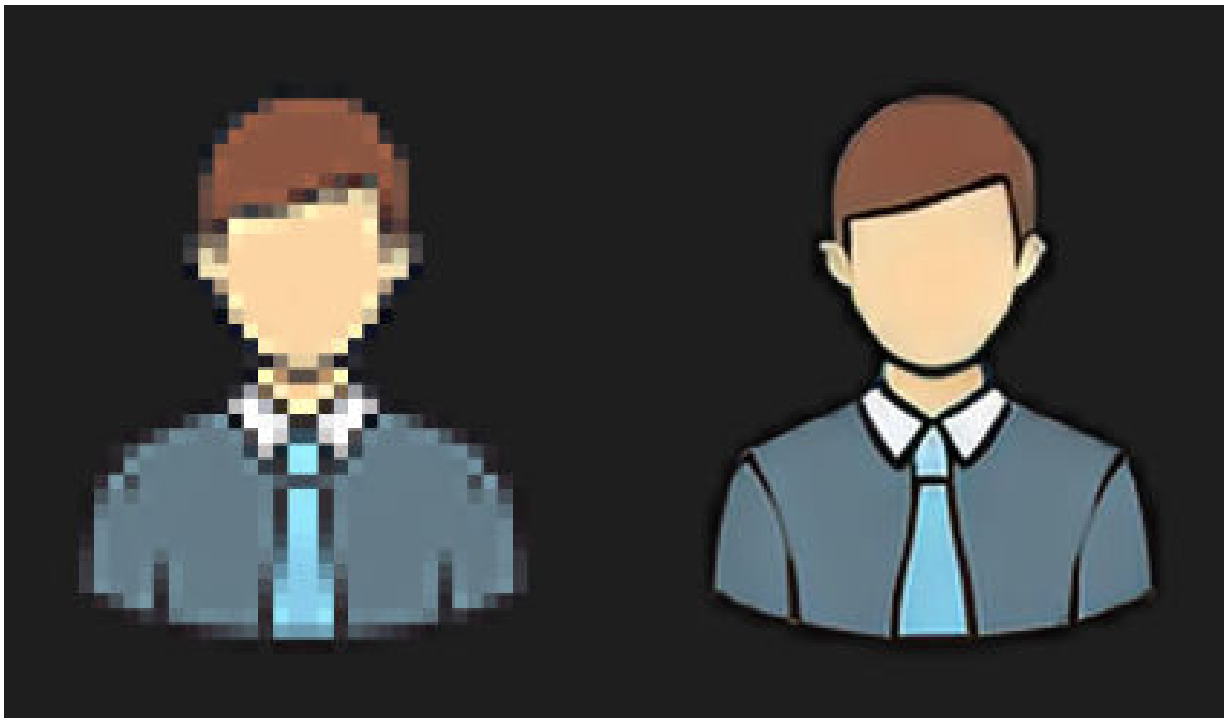


Figure 3.5 - Figma Boost resolution on old pixelated icon.

If this tool had existed during my freelance years, it would have saved me countless hours. Now, it’s an essential feature for any designer handling legacy assets or repurposing outdated graphics for modern screens.

Make an Image

AI image generation has been making waves across the design industry. Figma's implementation brings this capability directly into your design

workflow, promising to create visuals based on text descriptions without leaving the design environment.

What's the problem?

AI-powered image generation promises to revolutionize design workflows by instantly creating high-quality visuals based on text prompts. The concept is simple: type a description, and AI generates an image that seamlessly integrates into your design. However, in practice, the results often fall short. Many of the generated images are not usable.Despite my experience as a heavy AI user, I found this feature unreliable, producing generic outputs that failed to meet expectations. While this tool has potential, it currently feels more experimental than practical.

How to fix it?

If you want to experiment with AI-generated images in Figma, here's how to get the best possible results:

1. **Use Simple, Clear Prompts**: AI performs best with concise and specific input, so structure your prompts carefully.
2. **Refine the Output**: AI-generated images often require post-processing or manual adjustments to fit into a design.
3. **Combine AI with Existing Assets**: Use AI-generated images as a starting point rather than a final product.
4. **Test Different Prompts**: Small wording changes can yield significantly different results, so experiment with variations to get closer to what you need.

Although this feature is not yet reliable for production work, I am confident that it will continue to improve. AI image generation is evolving rapidly, and in the future, it could become an essential tool for designers.

Automating Routine Tasks with AI in Figma

Beyond Figma's built-in AI features, there's tremendous potential in using AI to automate repetitive design tasks. The key is identifying which parts of

your workflow can benefit from automation without sacrificing creative control or design quality.

What's the problem?

As a designer, you handle countless routine tasks daily. Over the years, as a mentor, I've met individuals who dedicate their evenings and weekends to automating not just their work but their entire lives. However, this level of thinking is rare. For the rest of you, here's my approach to thinking about automating routine tasks effectively.

How to fix it?

To successfully automate routine tasks in Figma, follow this structured approach:

1. **Schedule a Review**: Block a calendar slot every two weeks (or once a month) for two hours to analyze your workflow.
2. **Document Repetitive Tasks**: Create a simple sheet or document where you log time-consuming or repetitive tasks as you encounter them.
3. **Log in Chronological Order**: Write them down as they happen. If some tasks appear multiple times, that's a good indicator of where automation could help.
4. **Analyze Trends**: During your scheduled review, assess your notes manually or with AI tools like ChatGPT to identify the most repetitive tasks.
5. **Determine What Can Be Automated**: Not everything can or should be automated, but recognize patterns where automation makes sense.
6. **Implement and Iterate**: Apply automation techniques to the most frequent pain points and refine the process in your next scheduled session.

By incorporating this method, you can gradually optimize your workflow, eliminate redundant tasks, and free up more time for creative design work.

Integrating AI Tools for Faster Prototyping

Not everything is possible in Figma, and sometimes it's better to use other tools. In the past, we would create prototypes in Figma, but now we can generate functional code in minutes.

What's the problem?

Static designs can't explain everything. Sometimes, for user testing or to help stakeholders understand the concept, you need a functional prototype. Creating a fully functional prototype can be straightforward for simple websites but extremely challenging or nearly impossible for complex applications (especially those with interactive elements like tables).The good news is that we now have AI tools that can quickly generate code from your designs. You can publish this ready-to-use code and send it to users, stakeholders, or developers, providing a much more interactive and realistic representation of the final product.

How to fix it?

I've had good experiences with `lovable.dev`, and I've spoken with designers who have had very positive results with `v0.dev`. Both applications function on the same principle: you provide instructions along with your Figma design, and they generate ready-to-use code in seconds. You can then go step-by-step to optimize it to perfection.Here are some valuable tips I've discovered:

- **Provide comprehensive information upfront**: While you can start with simple 3-line prompts and hope for the best, in my experience, it's better to give a more complete picture of what you want to build. When I've provided minimal information initially, thinking I'd add more details later, the system sometimes got stuck in an infinite bug loop.
- **Leverage other AI code tools like ChatGPT or Claude**: Sometimes, these code generation tools create problems they can't fix themselves. When this happens, it's helpful to edit the code yourself—sync it with GitHub (in the case of `lovable.dev`, which I use), let ChatGPT or Claude fix the bug, then sync it back. This workflow is surprisingly effective.

- **Don't hesitate to start fresh**: If your project isn't progressing, sometimes it's best to start over. Rather than copying your entire original prompt, update it to incorporate the changes you've made along the way. This approach typically produces better outcomes.

Implementing AI into Your Design Workflow

While integrating AI as an individual designer is relatively straightforward, rolling out AI tools across an entire team requires a more structured approach. Success depends not just on the technology itself, but on how thoughtfully you introduce these new capabilities to your colleagues.

What's the problem?

Implementing AI into your design workflow is straightforward when working alone, but introducing AI across a team of designers and developers can be much more complex. Different team members have different needs, levels of experience, and concerns about AI's role in their work. While we've already covered how to use AI to automate individual tasks in *Automating Routine Tasks with AI in Figma*, this section focuses on integrating AI across an entire team workflow to improve efficiency and collaboration.

How to fix it?

Approach this as you would any other design problem—treat it like a structured project with clear requirements and solutions. Here's a step-by-step guide to successfully implementing AI across your design team:

1. **Define Your Target Audience**: Before implementing AI, understand who it will affect. Are you introducing AI to a small team of designers, or are you rolling it out to dozens of developers and product managers? Identifying your audience helps you craft a strategy that resonates with them and meets their specific needs.
2. **Find an Ambassador**: Once you've identified your target audience, recruit an ambassador from each key group (e.g., designers,

developers, marketing, or content). While you may lead the initiative, having a trusted advocate within each discipline ensures better adoption. For example, if you want developers to embrace AI, having a developer champion the cause will make communication and adoption smoother.

3. **Identify Pain Points**: AI should be a solution to real problems, not just a trendy addition. Identify pain points by talking to your team and gathering insights. Keep in mind that some pain points will originate from management rather than the team itself. For instance, leadership may push for faster delivery, but designers and developers may not see this as a personal issue.
4. **Workshop the Strategy**: In Dotidot, we've eliminated most unnecessary meetings, keeping schedules clear for focused work. However, a well-structured workshop can be incredibly effective in defining an AI implementation strategy. Gather the ambassadors and key stakeholders, walk them through the identified problems, and guide them toward defining solutions themselves. A well-run workshop will lead to stronger buy-in and a more tailored strategy.
5. **Create a Clear, Concise Plan**: After the workshop, document the findings in a simple, one- or two-page document outlining:
    - The target group
    - The specific problems they face
    - How AI can address these problems
    - Clear steps for implementation
6. **Run a Pilot Program**: Start with a pilot program rather than a full rollout. Set a strict evaluation deadline to assess whether the AI integration is actually improving workflows. Make it clear to the team that this isn't an after-hours side project—it's a structured initiative designed to improve efficiency and design quality.

By treating AI integration as a design challenge, rather than a forced technological shift, you can ensure smoother adoption, better team engagement, and ultimately, more impactful results in your design workflow.Not every team or individual will immediately embrace these changes, and that's exactly why Ambassadors play a crucial role. If you're implementing improvements across multiple teams, begin with the most receptive group to build a strong, data-backed case for management.

Ideally, every change would organically emerge from individual team members, but in reality, that’s not always how things work. A solid proof of concept enables you to approach management and implement changes from the top down when necessary.

Analyzing AI-Enhanced Design Workflows—Case Studies

At dotidot, we're a small team that implemented AI into our design processes not as a trend, but as a necessity to work smarter and faster. I'd like to share three key use cases where we've trained ChatGPT to enhance our daily design workflows:

Marketing Specialist Assistant

This AI assistant embodies our target persona—automation strategists who work closely with clients to implement complex automation strategies. These specialists have limited availability but possess invaluable market knowledge that designers occasionally need to tap into.Our actual marketing specialists use dotidot for several hours daily to configure client setups. Since we make nearly all functions accessible to users, these power users have a deep understanding of the product experience. The AI assistant helps bridge the gap when the human specialists aren't available, allowing designers to get perspective on user needs and market-specific knowledge without scheduling additional meetings.

UX Writer Assistant

As a B2B application, dotidot contains numerous complex elements that require precise, clear wording. While we have an excellent UX writer on staff, their limited bandwidth meant they couldn't address every small copy request.Our human UX writer became an excellent "teacher" for the AI assistant, continually helping us refine its capabilities. The assistant now handles routine copy tasks while maintaining our voice and standards, freeing our UX writer to focus on more strategic communication challenges and complex features.

Marketing Content Assistant

At dotidot, our product team is also responsible for in-app marketing materials such as promotional banners, information boxes, and feature announcements. We trained a specialized AI assistant specifically for creating marketing copy that aligns with our brand voice.This enables designers to simultaneously develop both the feature and its supporting marketing assets. Since the designers have the deepest knowledge of the new functionality, this integrated approach ensures marketing content accurately reflects the feature's capabilities and benefits while maintaining consistency across the product.

Navigating Ethical Challenges in AI-Driven Design

If you're a hardcore Figma user, you might remember when Figma announced the **Make Design AI** feature at Config—only to pull it back within days after users noticed it generated designs strikingly similar to popular apps. I won’t go too deep into this, but Figma claimed they didn’t train the model on past Figma designs. Still, as a product person, I find it hard to believe that using existing Figma data wouldn’t have been the logical starting point.AI doesn’t generate ideas from thin air—it builds on the data it’s trained with. How much we integrate AI into our design process is a decision each of us has to make. For example, I see **Renaming Layers** as a no-brainer; it saves time without compromising creativity. However, using AI to generate the first version of an app design? That’s where I draw the line. It may not be outright unethical, but it fundamentally changes the craft of design—replacing thoughtful decision-making with patterns pulled from who-knows-where on the internet.