Design Principles for Generative AI Applications

Generative AI technologies are capable of incredible feats: chatbots that speak fluently when answering our questions, image generation models that produce high-fidelity artworks and illustrations from our words, and coding assistants that help us write source code more quickly. New applications we couldn't envision just a few years ago are now being created over the course of an afternoon or weekend with state-of-the-art foundation models. Existing applications from companies like Adobe and Microsoft have also been infused with generative capabilities to provide users with new, co-creative experiences.

Given this rapid commercialization of generative AI technologies, there is an urgent need for guidance on how to design user experiences that foster *effective* and *safe* use. Generative AI technologies carry a number of risks, such as the tendency to generate plausible, but untrue information (known as hallucination), and the generation of content that reinforces stereotypes. Although much of the attention in research communities has focused on making technological advancements to address these issues, this work does not often address an important half of what Ehsan et al. call the "human-AI assemblage" — the human.

Why do we need new guidelines for generative AI?

There has been a wealth of guidelines, principles, and frameworks for the design of effective and safe computing systems developed over the past few decades. There have even been guidelines developed specifically for AI systems, including Amershi et al.'s Guidelines for Human-AI Interaction and Google's People + AI Guidebook. Why do we need yet another set of guidelines? The answer is simple: none of the existing guidelines address these unique considerations and challenges posed by generative AI:

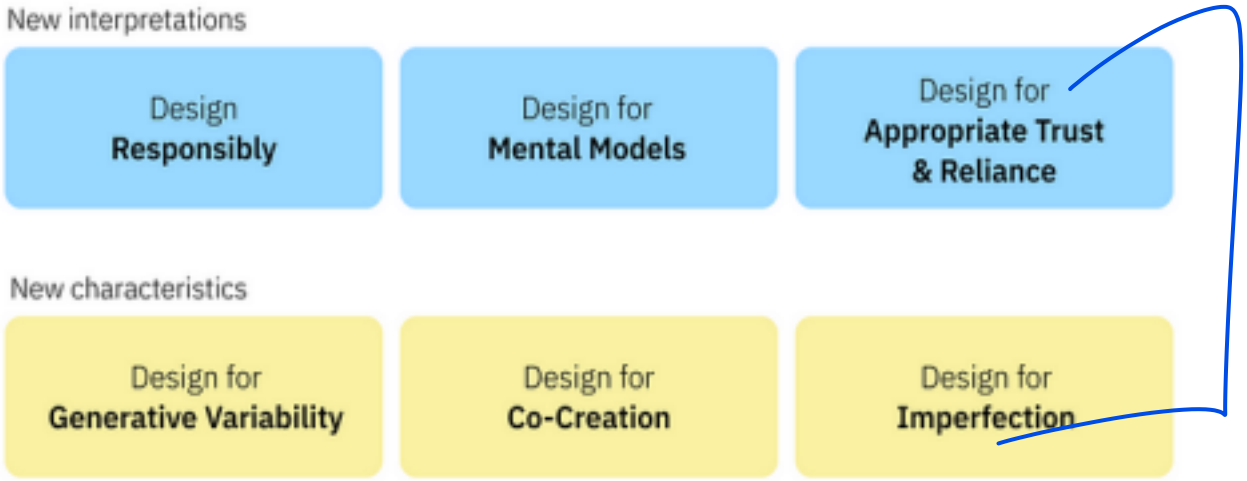
1. Generative AI introduces a new interaction paradigm. Jakob Nielsen recently argued that generative AI has enabled a new form of human-computer interaction, which he called *intent-based outcome specification*. Instead of typing, clicking, or touching our devices to get them to do what we want, we can now write natural language specifications to instruct the computer in *what* we want, but not *how* it should be produced — the AI model takes care of that. How

can we design AI user
experiences where users learn how to effectively specify the *what* and deal with cases when the outputs aren't what they expect?

2. Generative variability is the idea that every time the user clicks the "generate" button, they get different results. This behavior counters traditional UX guidelines that user interfaces should operate in consistent and predictable ways. How can we help users form the right mental models of these inconsistent, probabilistic systems?
3. There are new risks & potential user harms when dealing with generative AI. Hallucinations, toxic language, copyright infringement, and personal information leakage are just some of the risks of generative AI. How can designers navigate these tricky issues?

What are our design principles for Generative AI?

We developed six principles to help design practitioners create generative AI user experiences (UX). We call them *principles*, rather than guidelines, to highlight their fundamental nature to the design of generative AI UX. Three principles focus on unique characteristics of generative AI systems, and three principles offer new interpretations of existing issues with AI systems when viewed through the lens of generative AI. These principles are not hard rules that you must follow when designing generative AI UX. Rather, it is up to you — the designer — to use your best judgment of whether and how a principle applies to their particular use case. To make the principles *actionable*, we coupled each with four specific design strategies that exemplify how to implement the principle (through UX capabilities or the design process itself). We also identified real world examples of each design strategy in action.



Design principles for generative AI applications

We developed the principles through an iterative process that involved reviewing recent literature, evaluating commercial examples of generative AI applications, and working with multiple internal teams to assess the principles by applying them within their design process.

Principle 1: Design Responsibly

The most important principle to follow when designing generative AI systems is to design responsibly. The use of all AI systems, including those that incorporate generative capabilities, may unfortunately lead to diverse forms of harm, especially for people in vulnerable situations. As designers, it is imperative that we adopt a socio

technical perspective toward designing responsibly: when technologists recommend new technical mechanisms to incorporate into a generative AI system, we should question how those mechanisms will improve the user's experience, provide them with new capabilities, or address their pain points.

Strategy 1: Use a human-centered approach. Design for the user by understanding their needs and pain points, and not for the technology or its capabilities. → ***Example:*** *Human-centered approaches such as design thinking and participatory methods allow you to observe users' workflows and pain points to ensure proposed uses of generative AI are aligned with users' actual needs.* **Strategy 2:** Identify and resolve value tensions. Consider and balance different values across people involved in the creation, adoption, and usage of the AI system. → ***Example:*** *Value Sensitive*

Design (VSD) is a method that can help designers identify who the important stakeholders are and navigate the value tensions that exist across them.

Strategy 3: Expose or limit emergent behaviors. Determine whether generative capabilities beyond the intended use case should be surfaced to the user or restricted.

→ ***Example:*** *Conversational interfaces that enable open-ended interactions will allow such emergent behaviors to surface. For example, a user may discover that ChatGPT can perform sentiment analysis, a task that it (likely) wasn't explicitly trained to do. By contrast, graphical user interfaces (GUIs), such as AIVA, can place limits on the ways a user can interact with the underlying generative model by only exposing selected functionality.*

Strategy 4: Test & monitor for user harms. Identify relevant user harms (e.g. bias, toxic content, misinformation) and include mechanisms that test and monitor for them.

→ ***Example:*** *One way to test for harms is by benchmarking models on known data sets of hate speech and bias. After deploying an application, harms can be flagged through mechanisms that allow users to report problematic model outputs.*

Principle 2: Design for Mental Models

A mental model is a simplified representation of the world that people use to process new information and make predictions. It is their own understanding of how something works and how their actions affect it. Generative AI poses new challenges to users, and designers must carefully consider how to impart useful mental models to help users understand how a system works and how their actions affect it. Also consider users' backgrounds and goals and how to help the AI form "mental models" of their users.

Strategy 1: Orient the user to generative variability. Help the user understand the AI system's behavior and that it may produce multiple, varied outputs for the same input. → *Example: Google Gemini provides answers in the form of multiple drafts, indicating that it came up with multiple, varied answers for the same question.*

Strategy 2: Teach effective use. Help the user learn how to effectively use the AI system by providing explanations of features and examples through in-context mechanisms and documentation.

→ *Example: DALL-E provides curated examples of generated outputs and the prompts used to generate them. Adobe Photoshop provides pop-ups and tooltips to*

introduce the user to its Generative Fill feature.

Strategy 3: Understand the user's mental model. Build upon the user's existing mental models and evaluate how they think about your application: its capabilities, limitations, and how to work with it effectively.

→ Example: In evaluating a Q&A application, you might ask the user, "how did the system answer your question about who the current President is?" Answers such as, "it looked it up on the web" might indicate a need to educate users about hallucination issues. Users' existing mental models of other applications can also be useful to understand. For example, Github Copilot builds on users' mental models by following the same interaction pattern as its existing code completion features, which are familiar to many developers, hence easing their learning curve.

Strategy 4: Teach the AI system about the user. Capture the user's expectations, behaviors, and preferences to improve the AI system's interactions with them. → *Example: ChatGPT provides a form for "Custom Instructions" in which users provide answers to questions such as, "Where are you based?"", "What do you do for work?", and "What subjects can you talk about for hours?" In this way, users teach ChatGPT about themselves in order to receive more personalized responses.*

Principle 3: Design for Appropriate Trust & Reliance

Trustworthy generative AI applications are those that produce high-quality, useful, and (where applicable) factual outputs that are faithful to a source of truth. Calibrating users' trust is crucial for establishing appropriate reliance: teaching users to scrutinize a model's outputs for quality issues, inaccuracies, biases, underrepresentation, and other issues to determine whether they are acceptable (e.g. because they achieve a certain level of quality or veracity) or if they should be modified or rejected.

Strategy 1: Calibrate trust using explanations. Be clear and upfront about how well the AI system performs different tasks by explaining its capabilities and limitations. *→ Example: ChatGPT explains its capabilities (e.g. "answer questions, help you learn, write code, brainstorm together") and limitations (e.g. "ChatGPT may give you inaccurate information. It's not intended to give advice.") directly on its introduction screen.*

Strategy 2: Provide rationales for outputs. Show the user why a particular output was generated by identifying the source materials used to generate it.

→ *Example: Google Gemini provides a list of sources it used to produce answers to questions. Adobe discloses that its Generative Fill feature was trained on "stock imagery, openly licensed work, and public domain content where the copyright has expired."*

Strategy 3: Use friction to avoid overreliance. Encourage the user to review and think critically about outputs by designing mechanisms that slow them down at key decision making points.

→ *Example: Google Gemini displays multiple drafts for the user to review, which can encourage them to slow down and consider which drafts may be of lower or higher quality.*

Strategy 4: Signify the role of the AI. Determine the role the AI system will take within the user's workflow.

→ *Example: Github Copilot's tagline is "Your AI pair programmer," which elicits the role of a partner. Copilot fulfills this role by proactively making suggestions as the user writes code. It also possesses a limited form of agency by making autocompletion suggestions directly in the user's code editor, although it requires the user to explicitly accept or reject those suggestions (e.g. by pressing tab or escape).*

Principle 4: Design for Generative Variability

One distinguishing characteristic of generative AI systems is that they can produce multiple outputs that vary in character or quality, even when the user's input does not change. This characteristic raises important design considerations: to what extent should multiple outputs be visible to users, and how might we help users organize and select amongst varied outputs?

Strategy 1: Leverage multiple outputs. Generate multiple outputs that are either hidden or visible to the user in order to increase the chance of producing one that fits their need.

→ *Example: DreamStudio, DALL-E, and Midjourney all generate multiple distinct outputs for a given prompt; for example, DreamStudio produces four images by default and can be configured to produce up to 10. ChatGPT allows the user to regenerate a response to see more options.*

Strategy 2: Visualize the user's journey. Show the user the outputs they have created and guide them to new output possibilities.

→ *Example: DreamStudio, DALL-E, and Midjourney all show a history of the user's inputs and resulting image outputs. A research prototype extends the idea of "visualizing the user's journey" by showing a 2D visualization of parameter configuration options with indicators of which combinations the user has tried.*

Strategy 3: Enable curation & annotation. Design user-driven or automated mechanisms for organizing, labeling, filtering, and/or sorting outputs. → *Example: DALL-E allows the user to mark images as favorites and store them within groups called collections. Users may create and name multiple public or private collections to organize their work.*

Strategy 4: Draw attention to differences or variations across outputs. Help the user identify how outputs generated from the same prompt differ from each other. → *Example: DreamStudio, DALL-E, and Midjourney all display multiple outputs in a grid-like fashion to allow the user to identify differences, but fine-grained differences*
between outputs are not explicitly highlighted. A prototype source code translation interface visualizes the differences across multiple generated code translations through granular highlights and a list of alternate translations.

Principle 5: Design for Co-Creation

Generative AI offers new co-creative capabilities. Help the user create outputs that meet their needs by providing controls that enable them to influence the generative process and work collaboratively with the AI.

Strategy 1: Help the user craft effective outcome specifications. Assist the user in prompting effectively to produce outputs that fit their needs.

→ *Example: The IBM watsonx.ai Prompt Lab documentation includes a set of tips and examples to help the user understand how to improve their prompts.* Strategy 2: Provide generic input parameters. Let the user control generic aspects of the generative process such as the number of outputs and the random seed used to produce those outputs.

→ *Example: DreamStudio provides a slider for users to indicate the number of images they want to produce for a given prompt, along with an input field for random seed.*

Strategy 3: Provide controls relevant to the use case and technology. Let the user control parameters specific to their use case, domain, or the generative AI's model

architecture.

→ *Example: AIVA allows the user to customize domain-specific characteristics of the musical compositions it generates, such as the type of ensemble and emotion.*

Strategy 4: Support co-editing of generated outputs. Allow both the user and the AI system to improve generated outputs.

→ *Example: Adobe Photoshop exposes generative AI capabilities within the same design surface as its other image editing tools, enabling both the user and the generative AI model to co-edit an image.*

Principle 6: Design for Imperfection

Users must understand that generative model outputs may be imperfect according to objective metrics (e.g. untruthful or misleading answers, violations of prompt specifications) or subjective metrics (e.g. the user doesn't like the output). Provide transparency by identifying or highlighting possible imperfections, and help the user understand and work with outputs that may not align with their expectations.

Strategy 1: Make uncertainty visible. Caution the user that outputs may not align with their expectations and identify detectable uncertainties or flaws.

→ *Example: Google Gemini's interface states, "Gemini may display inaccurate info, including about people, so double-check its responses." This disclaimer alerts the user to the possibility of uncertainties or imperfections in its outputs. A prototype source code translation interface makes the generative model's uncertainty visible to the user by highlighting source code tokens based on the degree to which the underlying model is confident that they were correctly translated.*

Strategy 2: Evaluate outputs using domain-specific metrics. Help the user identify outputs that satisfy measurable quality criteria.

→ *Example: Molecular candidates generated by CogMol, a prototype generative application for drug design, are evaluated with a molecular simulator to compute domain-specific attributes such as molecular weight, water solubility, and toxicity.*

Strategy 3: Offer ways to improve outputs. Provide ways for the user to fix flaws and improve output quality, such as editing, regenerating, or providing alternatives. → *Example: DALL-E and DreamStudio allow users to refine outputs by erasing and regenerating parts of an image (inpainting) or generating new parts of the image beyond its boundaries (outpainting). Google Gemini offers options for the user to modify outputs to be shorter, longer, simpler, more casual, or more professional.*

Strategy 4: Provide feedback mechanisms. Collect user feedback to improve the training of the AI system.

→ *Example: ChatGPT offers an option for the user to provide a thumbs up or thumbs down rating for its responses, along with open-ended textual feedback.*

Tools and platforms that simplify the integration of GenAI into user interfaces

While Artificial Intelligence (AI) tools cannot replace deep qualitative user experience research, UI designers can leverage AI tools to build an effective, data-driven, and inclusive interface. AI equips UI designers with handy tools that make the design process smoother, improve user interaction, and create personalized experiences.

Of course, some handy tools can help UX designers improve usability and functionality rather than aesthetics, while AI tools for UI are useful in adding color, logos, and typography to their designs. AI can help designers pick colors, shapes, or typefaces to drive an intuitive user experience.

In this article, we understand how AI can make UI more intuitive, the value it adds to UI design, and its overarching role in ensuring a friction-free user experience – by exploring a few interesting tools in the market.

Benefits of Using AI in UI Design

Role of AI in UI

UI design centers on the look and feel of a digital interface, including the arrangement, font styles, color palette, and overall visual appeal. In general, being a UI designer needs creativity and flexibility for various tasks. Hence, there are specific tasks where AI can assist designers.

- Make repetitive tasks easier and faster.
- Perform localization of designs through AI-powered translation · Segment audiences and provide the right product with an interface · Give insight into what elements users interact with the most and which require the most attention
- Provide consistency between users and products by using wireframing and prototyping tools.

How does AI enhance UI design

1. User Understanding with AI

AI tools analyze how people use digital things, learning their habits and

preferences. This information helps tweak the look and feel of screens to suit each person better.

2. Simplifying User Interfaces with AI

AI makes things easier by doing repetitive jobs and making complex tasks simpler. Smart language understanding helps chatbots and virtual helpers respond to user questions. AI-driven chatbots guide users, answering questions, and helping with tasks, making everything smoother for users and making the overall experience better.

3. Voice User Interfaces (VUI) and Natural Language Processing (NLP)

The emergence of voice-controlled devices and virtual assistants brings new possibilities in UI design. AI-powered voice interfaces (VUI) use smart language understanding (NLP) to get what you say and give helpful responses. This technology allows users to interact with applications using voice commands, enhancing accessibility and convenience.

4. Visual Design Assistance

AI becomes a design buddy, suggesting creative ideas and handling repetitive tasks. Design tools, powered by smart algorithms, study patterns, colors, and fonts to offer valuable insights.This combination between AI and designers accelerates the design process while maintaining creative control and innovation.

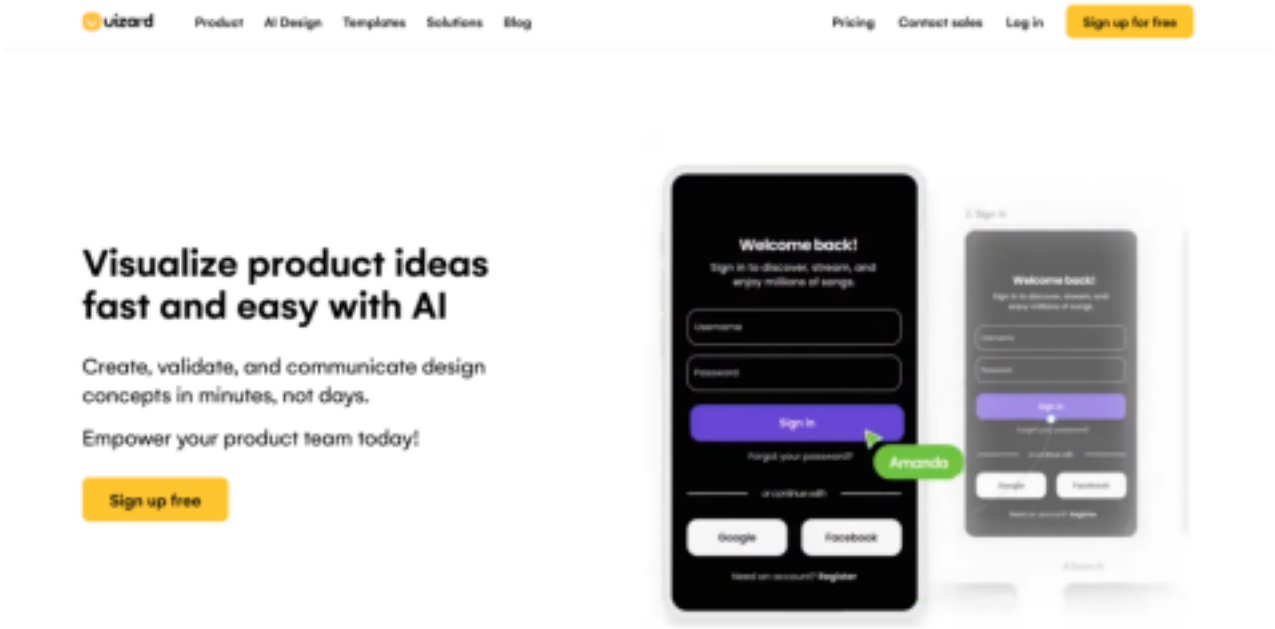
5. Automated Testing and Quality Assurance

Regular testing and quality checks can take a lot of time and effort. AI powered automation tools make this much easier by testing UI parts automatically, finding problems, and creating reports. This saves time and resources and guarantees a smoother, error-free user experience.

New-age AI Tools for UI Design

Uizard

This AI-powered design tool generates UI designs from text inputs, converts hand-drawn sketches into wireframes, and transforms screenshots into editable designs.

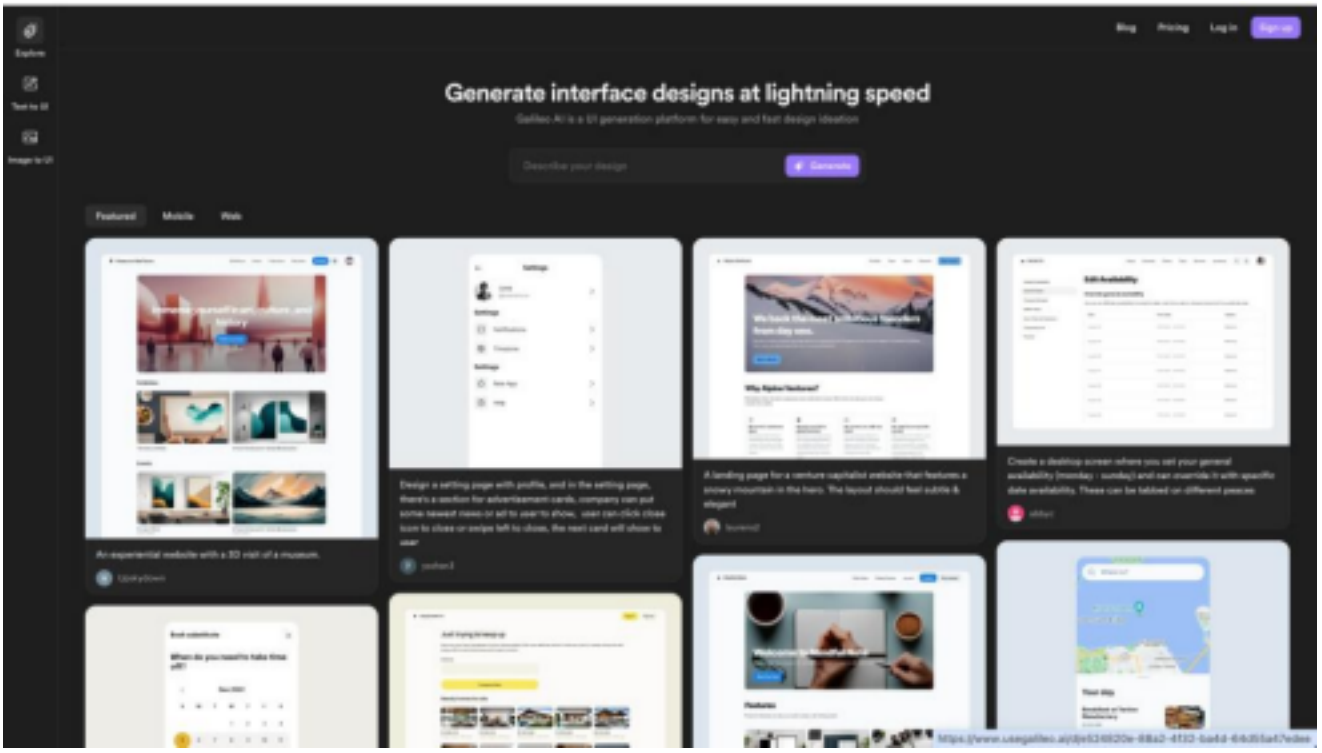


Khroma

Khroma is a color palette generator tool for designers, using AI to help them discover and save color combos they like. It learns from your choices and creates endless palettes right in your browser. You can train it by picking colors you prefer and exploring countless combinations for text, gradients, palettes, or even your own images.

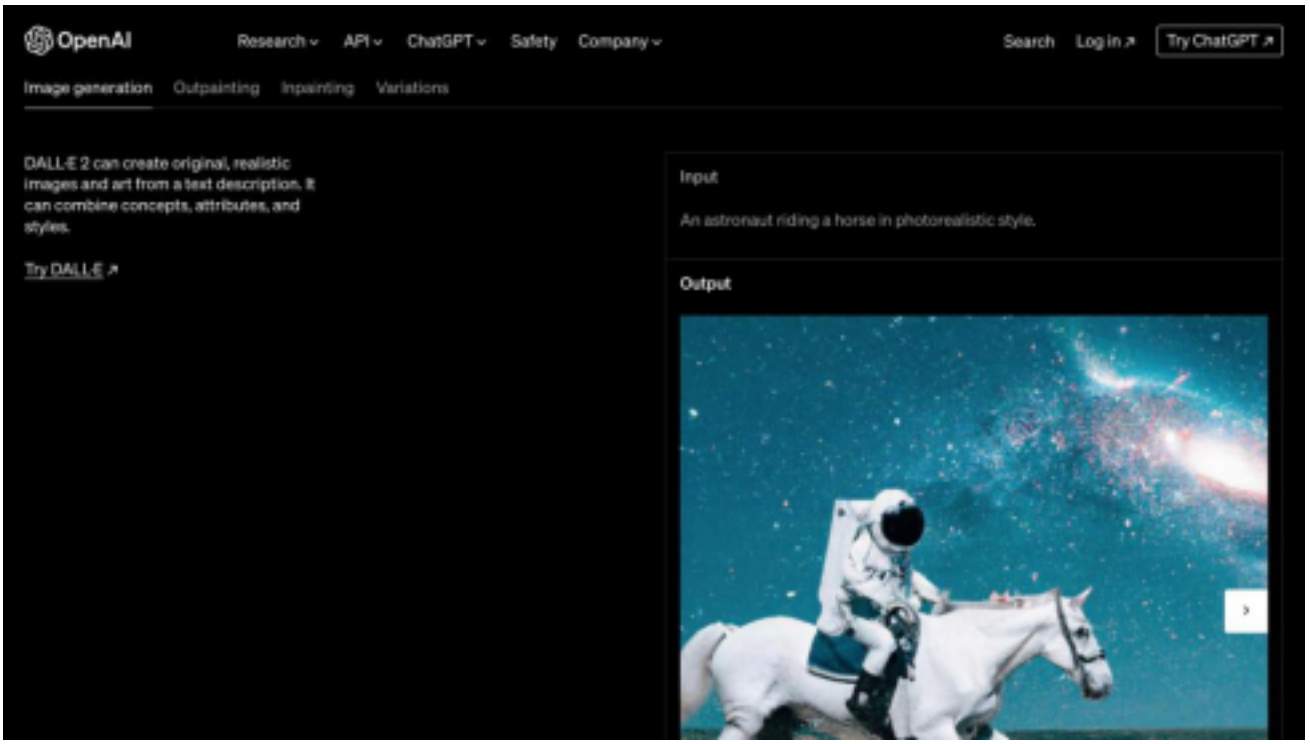
Galileo AI

Galileo AI is a generative AI tool that lets users create editable UI designs faster. It uses advanced AI algorithms to generate designs from text prompts.



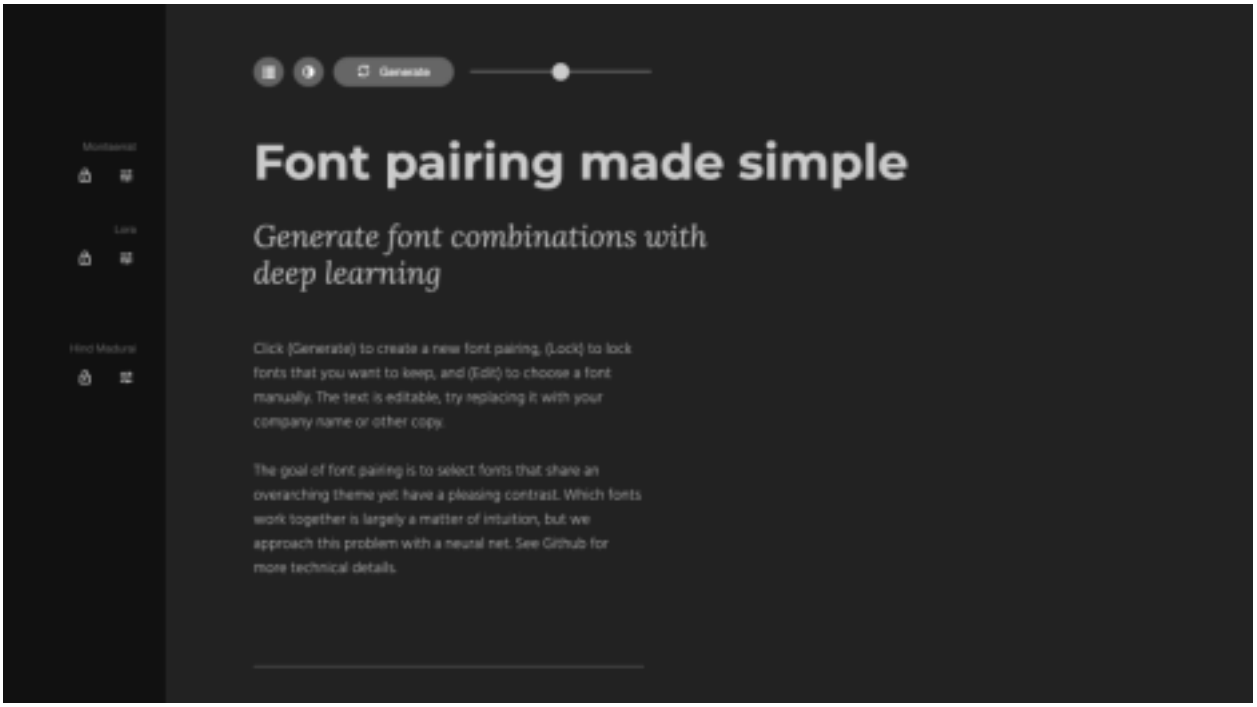
DALL·E 2

DALL·E 2 is a text-to-image AI tool that can generate realistic images from text descriptions. It turns your words into visuals.



Fontjoy

Fontjoy is a tool for pairing fonts. Fontjoy integrates with AI-powered tools, to help designers find the perfect fonts for their designs.rs find



Let’s Enhance

This AI tool automatically enhances and upscales images, allowing users to improve resolution, colors, and lighting.