ChatGPT

Napkin.ai – Case Study on Technical Implementation

Introduction

Napkin.ai is an AI-driven platform that **transforms written text into visual graphics** almost instantly. Launched in 2024 by Pramod Sharma and Jérôme Scholler (founders of the educational game Osmo), Napkin aims to automate the creation of charts, diagrams, and infographics for business storytelling [1] [2]. The idea was born from the founders' frustration with long documents and slide decks; they envisioned a tool where a user can *"paste your content, hit a button, and watch your text transform into a complete visual story in under five seconds"* [3]. In essence, Napkin uses **generative AI** to interpret textual input and produce **insightful visuals** such as flowcharts, mind maps, Venn diagrams, pie charts, and more [4]. This report delves into Napkin's technical underpinnings – how it processes text with AI, generates editable graphics, and what technologies enable this – and also examines real-world use cases and user feedback on the platform.

How Napkin.ai Turns Text into Visual Graphs

1. Natural Language Processing (NLP) with LLMs: Napkin's core engine begins by *"deeply understanding text content"* [5]. The introduction of modern **Large Language Models (LLMs)** was a turning point in Napkin's development, allowing a viable text-first approach to diagram creation [6]. When a user inputs text (e.g. a paragraph or list describing a process or concept), Napkin feeds it into an NLP model – reportedly a **generative AI** model – to parse the meaning and identify key elements. In fact, Napkin's founders credit LLMs with *"transforming text into visuals automatically, meeting users where they are most comfortable – writing text"* [6]. This suggests Napkin likely leverages a large pre-trained model (such as GPT-type models) to interpret the user's text and even to **generate intermediate representations** of the visual. For example, testers observed that Napkin will summarize input and sometimes even produce **ASCII/ markdown sketch diagrams** or an outline as an intermediate step [7]. This implies the backend might prompt an LLM to propose a structure (like a list of points, relationships or a simple text-art flowchart) which Napkin can then convert into a graphic.

2. ML Models Trained on Visual Data: Beyond language understanding, Napkin **combines NLP with computer vision techniques** to create the visuals [8]. According to one review, *"Napkin AI utilizes advanced machine learning models trained on extensive sketch data and visual representations"*, meaning the system was likely fine-tuned on a specialized dataset of diagrams and sketches [8]. This training would help the AI map textual concepts to visual motifs. Napkin's technology can recognize when text describes a timeline, a hierarchical process, pros-and-cons, a statistical breakdown, etc., and then choose an appropriate visual format [9]. The platform is essentially pioneering a new "visual generative AI" domain [10], where the AI doesn't generate a fanciful image (like DALL·E would) but **designs an informative diagram**. Notably, Napkin's co-founder emphasizes it's *"not Salvador Dali"* – the goal is not abstract art but **visualizing information** clearly [11].

Under the hood, Napkin's pipeline likely works as follows: the LLM/NLP component parses the text and decides on a **visual framework** (Napkin has a library of over *"30 kinds of graphics, including mind maps, flow charts, Venn diagrams, bar charts, pie charts and more"* [4]). Earlier iterations of Napkin actually relied on a fixed "database of visual frameworks" like Venn diagrams and pie charts, but required the user to pick one [12] . In the current version, the AI automatically suggests the relevant visual format. The model may output a structured representation of the content – for instance, a JSON or XML describing nodes and connections, or even code for a chart. It might, for example, detect a sequence in the text and choose a flowchart, or detect comparative language and choose a pros/cons table. If numerical data or a table is detected in the input, Napkin will switch to a **chart generation mode**, offering graphs like bar, line, pie, or gauge charts [13] [14] . *(In fact, the Napkin team confirmed that if your input "matches any of the relevant visuals" for data, the AI will surface appropriate charts* [15] *.)*

3. Visual Graph Generation and Layout: Once a visual format is chosen, Napkin generates the actual graphic. The platform's approach here is to produce **fully editable vector graphics** rather than static images. Napkin creates a *"Notion-like canvas"* with the input text, and inserts a **"spark" icon** next to each paragraph or section [16] . When clicked, this triggers the AI to generate a visual for that section of text. Napkin then renders the output on the canvas as a diagram or infographic which the user can refine. Internally, this likely involves **graph layout algorithms**: for example, arranging flowchart boxes in sequence with arrow connectors, or laying out a mind map radially. The team has put emphasis on *generating "high-quality visuals" and then enabling editing* [5] . Napkin's generated diagrams come with sensible defaults – it *"automatically arranges"* elements and chooses a pleasing color scheme and icons relevant to the text [17] [18] . To do this, Napkin probably harnesses a combination of **template-based design and AI image generation**. A Reddit user who tested Napkin noted that *"it's just a text to template converter really"*, suggesting that Napkin often populates pre-defined diagram templates with the user's content [19] . Indeed, Napkin offers multiple style options (e.g. *"ladders, arrows, process funnels"* as visual styles [20]), indicating a set of predefined layout patterns. The AI's role is to pick the right pattern and map the text into it.

However, Napkin is not limited to rigid templates – it also employs **dynamic graphic generation**. The platform highlights features like *"dynamic connectors that can connect anything"*, *"smart decorators"* (e.g. highlights, emphasis shapes), and an *"extensive icon library"* to illustrate concepts [21] [22] . These suggest that after the AI decides on a diagram type, Napkin's backend (or front-end) programmatically draws the shapes and connectors. It may use vector graphics libraries or custom algorithms (for instance, a force-directed layout for concept maps, or hierarchical tree layout for flowcharts). The output is vector-based (users can export to **PNG, PDF, or SVG** formats [23] [24]), which implies Napkin likely uses **SVG under the hood** for rendering. The editing interface in the browser is interactive – implemented with modern web technologies (very likely **React/TypeScript** given the complex UI). Users can drag elements, edit text labels, swap icons, adjust colors and fonts, etc., so the diagram model must be maintained as an object graph (nodes/edges/properties) rather than a flattened image.

4. Backend Architecture and Tech Stack: Napkin is offered as a **cloud-based web application** (there's no install – it's "strictly web-based for now" [25]). The backend is presumably hosted on cloud infrastructure and orchestrates the AI and generation tasks. While the company hasn't publicly detailed its full stack, we can infer some components: The **AI models** (both NLP and any image-generation models) likely run on servers (possibly using Python with frameworks like PyTorch or TensorFlow, as is common for ML inference). For NLP, Napkin could be leveraging an API to a large model (e.g. OpenAI's GPT-4 or similar) or a fine-tuned model they host. The TechCrunch author's attempt to prompt Napkin with disallowed content hints that

Napkin's AI has a built-in moderation filter (it refused a "how to commit murder" request and added a disclaimer to offensive content) [26] [27]. This behavior closely resembles OpenAI's content policies, which suggests Napkin might be using a GPT-series model behind the scenes. It's also possible they fine-tuned a model specifically on diagram-related tasks, given the mention of training on "sketch data." On the **visual generation side**, Napkin's CEO noted that the system *"doesn't use any public or IP-protected data to generate pictures"*, meaning any imagery or illustrations it produces are based on either internally generated graphics or fully licensed data [28]. This could imply Napkin built its own small generative image model (trained on open or proprietary diagram illustrations) to create decorative artwork when needed, or it strictly uses its own graphic assets (icons, shapes, etc.). In either case, users *"don't have to worry about rights"* for the generated visuals [28].

For the web front-end, Napkin's interface is described as a canvas that mixes text and graphics (similar to Notion or Canva). Implementing this likely involves a combination of **HTML5 Canvas or SVG** for the drawing area and standard web tech for text editing. The ability to click a spark icon in a text paragraph to generate a visual suggests Napkin's front-end monitors the document structure and calls backend services for each section. This asynchronous generation flow (text -> AI call -> return visual) requires a robust web application setup. The site's responsiveness (users report that *"it's fast, and the results can easily be copied/downloaded"* [29]) indicates they optimized the pipeline, possibly caching common patterns or using efficient algorithms for layout. Napkin integrates with other tools mainly by allowing easy export/copy of images – e.g. you can paste results into Google Slides, Docs, PowerPoint, Notion, etc. [30] – so the integration is more on the user's side than via APIs at this point.

In summary, **Napkin's technical implementation** marries a language-understanding AI with a graphic generation engine. The NLP/LLM brain interprets and generates a visualization plan, and a rendering module instantiates that plan with vector graphics, using a palette of templates, icons and connectors. The system is trained and tuned specifically for business diagrams: *"visuals for information,"* not general artistic images [11]. By focusing on this niche, Napkin can use more structured approaches (like selecting from known diagram types) combined with AI creativity, rather than relying solely on unpredictable image diffusion models. The result is a controlled yet flexible generator that produces diagrams which are **immediately editable** by the user – a key differentiator that required building custom graphics software in addition to the AI.

Technologies and Frameworks Used

While Napkin.ai has not published a detailed tech stack, we can enumerate the likely **technologies and frameworks** based on its features and the team's background:

- **Large Language Models (LLMs):** Used for NLP tasks – likely GPT-3.5/GPT-4 or similar via API, or a fine-tuned transformer model. The AI's behavior (content filtering and adding an "educational purposes" note to insults) suggests an OpenAI model with moderation layer [26] [27]. The LLM generates structured outputs (summaries, lists, ASCII diagrams) that Napkin converts into visuals.

- **Custom Visual ML Model:** Napkin mentions models *"trained on sketch data and visual representations"* [8]. This could be a computer-vision model that either (a) generates simple line-art illustrations or (b) evaluates/optimizes the diagram layout. It might also refer to a multi-modal embedding model (like CLIP) used to match text concepts with iconography or design styles. Given

the founders' prior work in computer vision (at Osmo, detecting game pieces in real-time [31]), they likely brought CV expertise to analyze and generate visuals.

- **Web Front-End:** A rich web app that supports drag-and-drop editing, so frameworks like **React** (with possibly Redux or similar for state) are probable. The interface behaves much like a document editor with embedded graphics, which could be implemented with libraries for rich text (maybe ProseMirror or a custom editor) and for graphics. Graphics could be rendered using **SVG** elements manipulated via D3.js or a diagramming library (Napkin might have custom code, but conceptually it's similar to how **Mermaid.js** or **draw.io** render diagrams in SVG). The *connectors* and *layout* might use algorithms from graph visualization toolkits (e.g., Dagre or Elk for automatic layout of nodes, to avoid overlaps).

- **Backend and Cloud:** The service likely runs on a cloud platform (the company is based in California; possibly using AWS or GCP for scalability). The back end may use **Python** (commonly for AI pipelines) or **Node.js** for coordinating between the web front-end and AI services. Given Napkin's need to handle image generation and possibly heavy ML inference, a microservice architecture is plausible: one service for handling user requests and authentication, one for calling the LLM, one for generating the graphic (which might call further internal APIs or libraries to assemble the SVG). They also integrated third-party login (e.g., Google OAuth for sign-in [32]), so standard web auth services are in play.

- **Data and Asset Libraries:** Napkin includes an *"extensive database of icons"* [33] and uses 700+ Google Fonts [34] for styling. It likely integrates libraries or APIs for these (e.g., the Google Fonts API for typography, and possibly icon packs like Font Awesome or Noun Project – or custom SVG icon sets stored on their servers). Color palettes and style templates are probably defined in JSON or CSS, and Napkin can algorithmically switch themes (including a dark mode optimized palette [35]).

- **Export/Rendering Tools:** To export PDF and PNG, Napkin may utilize headless browser rendering or an SVG-to-PDF library on the backend. The fact that users can also share a **URL link** to the canvas [36] suggests Napkin saves the projects on the server so that a recipient can view the visual in Napkin's viewer.

Overall, Napkin's stack is a combination of **AI/ML components** and **web development**. As Accel's investor noted, the challenge was taking something technically complex and making it *"easy for users"* [37] [38]. This is achieved by hiding the AI's complexity behind a simple interface: the user just sees a "generate" button (spark icon) and some style choices, but behind that, Napkin's tech stack orchestrates **natural language understanding**, **visual template selection**, and **graphic rendering** in a matter of seconds.

Real-World Use Cases and Industries

Napkin.ai is designed for anyone who needs to communicate ideas visually, and it has gained traction across various roles and industries. Some key use cases include:

- **Business & Project Management:** Professionals creating presentations, strategy documents, or project plans use Napkin to generate slides and diagrams. For example, project managers find it *"immensely helpful"* for quickly turning concepts into diagrams for decks [39]. It automates business

charts and process flows that would otherwise take hours in Visio or PowerPoint. Napkin pitches itself as *"a designer at your fingertips"*, ideal for **pitch decks, proposals, and reports** where visuals can sell an idea [40].

- **Marketing & Content Creation:** Marketers and bloggers use Napkin to produce infographics and social media graphics. The platform's **one-click branding** features (applying brand colors/fonts automatically [41]) and social-media optimized outputs make it a fit for content marketing. A user on LinkedIn noted Napkin is *"very intuitive, I'm using it mainly for newsletter posts"* to add visual appeal to written content [13]. Napkin can take a long-form blog or newsletter text and insert several illustrative charts or conceptual images, boosting reader engagement. It's also been used to create visuals for **LinkedIn articles and Medium posts**, making complex info more digestible [42].

- **Product & Engineering Teams:** For technical domains, Napkin helps in **software architecture diagrams, workflows, and mind maps**. Engineers can describe a system architecture in text and get a quick diagram of the microservices, databases, and integrations involved. In one experiment, a user described a multi-microservice system with queues and cron jobs, and Napkin generated an architecture diagram capturing the interactions [43] [7]. It can also turn tables of data or comparisons into graphs. This is valuable for IT professionals documenting systems or for startups brainstorming product features (e.g., drawing an org chart or user flow from a text outline).

- **Education & Training:** Educators and students leverage Napkin to create didactic visuals. A professor testified that Napkin made it seamless to produce visuals for a LinkedIn article *"aligned with professional standards"* [42]. Teachers have found it useful for generating diagrams that *"bring textual ideas to life"* in the classroom [44]. Essentially, Napkin lowers the barrier for non-designers (like many educators) to include diagrams in curricula, lecture slides, or handouts, enhancing clarity and student engagement.

- **Consulting & Strategy:** Consultants and analysts often need to convey frameworks, SWOT analyses, process diagrams, etc. Napkin can quickly generate these from a written analysis. One growth marketing leader remarked he'd *"been waiting years for a tool like Napkin"* because he loves using business diagrams but isn't skilled at making them from scratch [45]. Now, even a rough written outline of a strategy can be turned into a polished graphic. Napkin is used for **mind maps, decision trees, flow charts** illustrating business strategies or product roadmaps, which is especially handy in client deliverables and workshops.

- **General Office Communication:** Many professionals use Napkin in day-to-day documentation – from summarizing meeting notes into a visual diagram, to adding a quick graphic in an email for clarity. Since Napkin works with common platforms (users can copy visuals into **Google Docs, Slides, Word, PowerPoint, Notion, Confluence, Slack** etc. [46]), it fits into existing workflows. It essentially gives any employee the power to create an infographic or diagram without formal design tools.

Napkin's mission is to bring visual communication to any context where people typically only use text [47]. By targeting **"marketers, content creators, engineers and professionals in the business of selling ideas"** [1], Napkin spans tech and non-tech industries alike. Early adopters include startup teams (for internal documentation and investor presentations), educators (for teaching materials), content marketers (for blogs and social posts), and corporate strategists. The common thread is the need to explain something

clearly and attractively – Napkin provides a shortcut to do exactly that, whether the content is a business plan, a product concept, research data, or a training module [48] [49].

User Experience and Feedback

Napkin.ai's user feedback has been generally positive, especially regarding its **speed and ease of use**, with some caveats noted around its limitations given the early stage. Here is a summary of user experiences:

- **Dramatic Time Savings:** Many users highlight how quick and effortless it is to produce visuals. *"I often use Napkin to create presentations or simply to generate visual ideas… it's been a game-changer for me,"* says one project manager [39]. Tasks that used to require a skilled designer or tedious manual work in tools like PowerPoint can now be done in seconds. A reviewer noted you can *"generate visuals in seconds, drastically reducing the time and effort typically required"* for design [50]. This speed is repeatedly praised – *"remarkably quick and easy to produce a plausible article and decent looking visuals,"* as one experimenter put it [51]. Even Napkin's own demo promises transformation of text to graphics *"in under five seconds"* [52], and users generally confirm it's fast.

- **Low Skill Barrier:** Napkin succeeds in empowering non-designers. Its AI handles layout, icon selection, and styling automatically. *"No design experience needed – the tool's smart templates and automation mean anyone can create stunning visuals,"* one review concludes [53]. Users appreciate that the interface is **intuitive** and the suggestions make it easy to get started. For instance, a marketing professional on LinkedIn described Napkin as *"Great tool, very intuitive"* for creating newsletter graphics [13]. The editing interface also receives positive mention – every element can be adjusted, which gives users confidence that even if the AI output isn't perfect, they can fix it. Being able to tweak colors, swap icons, or edit text on the graphic means the output can meet professional standards (as noted by a professor who was *"impressed by how seamless the process was"* and that results *"aligned with the professional standards I aim to maintain"* [42]).

- **Quality of Visuals:** The consensus is that Napkin produces clean, presentable graphics suitable for business use. Napkin *"generates exactly what I'm looking for in seconds,"* said one startup COO, adding that before Napkin they were very limited in what graphics they could easily create [40]. The style is deliberately simple and polished (using consistent color schemes, simple icons, and readable layouts). Some tech journalists have noted the design language is *"pretty generic"*, comparing it to a basic PowerPoint style [54]. This is sometimes a **trade-off**: the visuals won't be highly customized or artistic, but they will be neat and clear. Early users of Microsoft's similar AI PowerPoint tool made *"high school-level"* comments, and Napkin's output could evoke that sentiment [54]. However, for most business purposes, a clean and generic style is acceptable or even preferred. Napkin's team is continually adding new styles and design options to avoid monotony [55].

- **Usefulness vs. Complex Input:** One pattern in feedback is that Napkin performs best with **well-structured, concrete input**, whereas very abstract or complex text may confuse it. A user on Reddit observed: *"It's good at transforming text or concepts into various charts, especially if the concept is well structured (headings, paragraphs, lists)…. If you want to visualize a well-described idea, this is a straightforward and fast way."* [56] [57] On the flip side, *"it does not do well with abstract inputs"* [57]. The TechCrunch test echoed this: *"Napkin does best with simple descriptions, broad strokes of ideas… When the text is a bit more nebulous, Napkin grasps at straws, sometimes generating visuals that aren't grounded in the text at all."* [58]. For example, the AI might force an inappropriate graphic if it can't

parse the text's structure, resulting in a nonsense diagram. Users have learned that providing clear sectioned input (like a list of pros and cons, or step-by-step process) yields much better visuals than a vague essay.

- **Generative AI Quirks (Hallucination):** Because Napkin's engine is generative, it may occasionally **invent content** that wasn't in the original text. The TechCrunch reviewer noted Napkin *"invented pros and cons out of whole cloth"* in one of his tests – the AI added points about privacy issues and learning curve that the user never wrote [59]. This kind of hallucination is a known behavior of language models trying to fill gaps. It means users need to review the AI's output; Napkin gives full editing control, so users can delete or correct any AI-added text in the visual. Napkin is designed to assist, not to have final say – one user pointed out that it *"encourages laziness"* if you accept visuals without thinking, because part of creating good visuals is refining your own ideas [60]. Napkin's team will likely improve the fidelity of text-to-visual mapping over time, but for now, users should see the AI's suggestions as a first draft.

- **Limitations and Requests:** Some advanced users have bumped into Napkin's current limitations. **Custom layouts** outside the provided templates can be hard. For instance, a user wanted a specific arrangement (four equally sized tiles connected in a certain way) that Napkin couldn't produce [19]. He concluded *"it's just applying text and color to a library [of templates]"* and not truly creating novel layouts [19]. As a result, very specific graphic designs might still need manual tools. Users have also requested more support for data visuals – e.g. the ability to input their own data tables and generate charts on the fly [13] [61]. Napkin does some of this (it can detect simple tables and suggest charts [62]), but it doesn't yet let the user manually tweak data points through the interface. The Napkin team responded that they already support many chart types (line, bar, pie, etc.) when the AI deems it relevant [15], implying that as the AI improves at recognizing data in text, this will become smoother.

- **User Satisfaction:** By and large, those who fit Napkin into their workflow report high satisfaction. It's telling that users from diverse fields – a professor, a marketing leader, a project manager, a startup COO – all gave enthusiastic endorsements, calling Napkin *"invaluable"*, *"a game-changer"*, and something they would promote to colleagues [44] [39] [40]. The ability for "everyone to become a visual storyteller" is being realized; even those who *"dislike most of the PowerPoint options"* for diagrams now have a tool that does the heavy lifting for them [45]. Some criticisms (generic style, occasional nonsense outputs) are acknowledged, but these are often outweighed by the benefits of speed and convenience, especially given Napkin is still in beta and improving. The product was free (or had a free tier) during beta [63], which also positively influenced user sentiment.

In conclusion, **Napkin.ai's users generally find it a powerful assistive tool** that saves time and helps visualize ideas without design skills. It lowers the barrier to creating infographics and diagrams, which many find liberating. As one early user succinctly put it: *"Before Napkin, we were constrained regarding what graphics we could access, but now we have a designer at our fingertips."* [40] The excitement is tempered with realistic understanding that it's not perfect – yet, as the team expands and refines the AI, users expect the quality and variety of visuals to keep improving.

Conclusion

Napkin.ai represents a compelling case of **AI applied to visual communication**. Technically, it stands at the intersection of NLP and graphic design: it reads like an AI writing assistant and outputs like a graphics

editor. The platform's implementation showcases how large language models can be guided to perform specialized tasks – in this case, turning a chunk of text into an appropriate visualization. By training on sketch and diagram data and leveraging a library of visual formats, Napkin strikes a balance between generative creativity and template reliability. The backend's clever design (fusing text analysis with vector graphic generation) enables a seamless user experience where *"editing is added to complement the generation"* [64] rather than the other way around.

Napkin's emergence also highlights a growing demand across industries for quick visual content. From our exploration, it's clear that **marketers, educators, business leaders, and developers** alike have embraced the tool to cut down the friction of creating diagrams. In real-world usage, Napkin has already begun to transform workflows – pitches are more visual, documents more skimmable, ideas more shareable. The user testimonials underscore that Napkin delivers real value by democratizing design capabilities.

Looking ahead, Napkin.ai's approach might foretell the next generation of productivity software, where *"generative-first"* interfaces become the norm [64]. Its technical journey (from initial sketching tools to LLM-powered automation) shows how advances in AI can unlock entirely new features. The company has secured significant funding to grow its team and refine the product [65] [37]. We can expect improvements like richer visual styles, smarter handling of complex input, and possibly interactive or animated visuals. But even in its current state, Napkin.ai stands as an innovative marriage of AI and UI: a system that **reads**, **imagines**, and **draws** – turning our napkin scribbles (or lengthy memos) into pictures worth a thousand words.

Sources: References are cited in text, including Napkin's official site and blog, tech journalism (TechCrunch, VentureBeat, SiliconANGLE), and user-contributed reviews and discussions. All citations follow the format `【source†lines】`. For example, Napkin's use of ML models trained on sketch data is described in a review [8] and the variety of supported graphics (30+ types) is noted in SiliconANGLE's coverage [4]. These and other cited lines provide the factual backbone for this case study's analysis.

---

[1] [9] [16] [25] [26] [27] [28] [37] [38] [54] [58] [59] [64] [65] Napkin turns text into visuals with a bit of generative AI | TechCrunch
https://techcrunch.com/2024/08/07/napkin-turns-text-into-visuals-with-a-bit-of-generative-ai/

[2] [5] [6] [11] [30] [55] Napkin uses visual AI to automate business storytelling | VentureBeat
https://venturebeat.com/ai/napkin-uses-visual-ai-to-automate-business-storytelling/

[3] [20] [52] Napkin AI Review - The Ultimate Design Hack: Create Stunning Visuals in Seconds! | by Natalie Saint | Medium
https://medium.com/@lilnat1981/napkin-ai-review-the-ultimate-design-hack-create-stunning-visuals-in-seconds-db517fee4576

[4] [10] [40] [46] [48] [49] Visual AI startup Napkin.ai launches graphics generator for business storytelling - SiliconANGLE
https://siliconangle.com/2024/08/07/visual-ai-startup-napkin-ai-launches-graphics-generator-business-storytelling/

[7] [32] [43] [62] [63] AI Tools: Napkin AI and Gamma - Fresh/Brewed
https://freshbrewed.science/2025/01/14/napkinai.html

[8] [17] [18] [41] [50] [53] Napkin AI Review: A Design Tool for Everyone
https://doneforyou.com/napkin-ai-review/

12 Napkin AI - Visual AI for Business Storytelling
https://www.napkin.ai/blog/introducing-napkin/

13 14 15 61 Napkin AI Great tool, very intuitive, I'm using it mainly for newsletter... | Gabriel Botsie
https://www.linkedin.com/posts/gabrielbotsie_napkin-ai-great-tool-very-intuitive-im-activity-7251004727734206464-I0kS

19 29 56 57 Napkin ai review? Has anyone used it? Any experience to share? : r/ObsidianMD
https://www.reddit.com/r/ObsidianMD/comments/1gkjoip/napkin_ai_review_has_anyone_used_it_any/

21 22 23 24 33 35 36 39 42 44 45 Napkin AI - The visual AI for business storytelling
https://www.napkin.ai/

31 47 About us - Napkin AI
https://www.napkin.ai/about-us/

34 Napkin AI Review: Turn Text Into Visuals With AI in Seconds
https://freshvanroot.com/blog/text-to-visuals-ai-napkin/

51 60 Napkin AI: An Experiment
https://www.uptechstudio.com/blog/an-experiment-with-napkin-ai