AI-driven design tools for academic websites: a comparative analysis and a Figma Make workflow

Herramientas de diseño basadas en IA para sitios web académicos: un análisis comparativo y el flujo de trabajo de Figma Make

Chebotarova Iryna[1], Kriachko Maryna[1]
[1] Department of Media Systems and Technologies, Kharkiv National University of Radio Electronics
iryna.chebotarova@nure.ua, maryna.kriachko@nure.ua

Abstract

The paper compares contemporary AI-driven design software and justifies the selection of Figma Make for a developing a multi-page academic department website built by a single student-designer. A two-tier review covers both classical UI/UX editors (Adobe XD, Sketch, Penpot, the classic Figma editor) and the new generation of AI-driven tools (v0 by Vercel, Lovable, Bolt.new, Pencil, and Figma Make). Eight practical requirements for an academic single-designer project are formulated and applied as comparison criteria. The methodological context of prompt-first design is examined, and a four-stage workflow — prompt, generate, refine, publish — is described and applied to the case of the Media Systems and Technologies Department website at Kharkiv National University of Radio Electronics. The study demonstrates that AI-driven design tools have become part of the standard professional toolkit, and that, among them, Figma Make uniquely combines a designer-friendly interface, a choice of language models, three complementary ways of editing, and built-in publishing.

Keywords: AI-driven design, Figma Make, prompt-first design, vibe coding, academic website.

Introduction

Generative artificial intelligence has rapidly transformed the toolset of contemporary web and interface design. Throughout 2024–2026, a new generation of AI-powered design platforms entered the market and shifted the field from static mock-ups towards prompt-first design — an approach in which functional web interfaces are generated directly from natural-language descriptions. The choice of an appropriate design tool has therefore become a key methodological decision for any web project, and especially for small teams or single-designer projects that previously could not realistically deliver a complete, publicly accessible website without a separate front-end development stage.

The task addressed in this study is the design of a multi-page website dedicated to the history and present-day life of the Media Systems and Technologies Department at Kharkiv National University of Radio Electronics. The goal of the project is non-trivial. It is not merely to produce a static prototype, but to deliver a fully working, publicly accessible website. The project must be built by a single student-designer who has only basic programming skills, without delegating implementation to a separate development team. These constraints narrow the range of suitable tools significantly and require a structured comparison of available options against well-formulated requirements.

The aim of this paper is therefore to compare the contemporary landscape of design software — both classical UI/UX editors and the recent generation of AI-driven tools — against the practical requirements of an academic single-designer project, and to justify the choice of Figma Make as the optimal environment for the described case (Chebotarova & Kriachko, 2026). The paper also describes the prompt-first methodology that the chosen tool supports and the four-stage practical workflow that has been applied to the project.

The shift to prompt-first design

The methodological context for the selection of any contemporary design tool is the shift from a classical, mock-up-centred workflow to prompt-first design. In the classical model, work was structured as a sequence of analysis, mock-up production, and hand-off to a developer, and the designer's main artefact was a static visual representation of the interface. With the emergence of generative AI tools, the designer can obtain a functional interface directly from a textual description, called a prompt, and a significant portion of routine layout work is delegated to the model.

The essence of prompt-first design lies in shifting the centre of the project activity away from the static mock-up and towards the formulation of a high-quality prompt. The prompt is no longer an auxiliary command; it becomes a central project artefact that partially performs the functions of a technical specification, a design concept, and a behavioural description of the interface (Owen & Roberts, 2024). The role of the designer also changes. From an executor who places every visual element by hand, the designer becomes a conductor who sets the direction, evaluates the generated result, and corrects it through detailed prompts, reference screenshots, and targeted manual edits.

An effective prompt for the generation of a web interface is best constructed as a multi-component structure. Current recommendations in prompt engineering agree that a high-quality request should contain a clearly defined task, context, tone, detailed rules of execution, and explicit criteria of acceptance. For web design, this means that the prompt should describe the purpose of the interface (for example, an educational landing page or a historical archive), the target audience, the page structure (hero block, navigation, cards, timeline, gallery, footer), visual and stylistic parameters, behavioral and adaptive requirements, and quality constraints such as readability, contrast, and stylistic consistency. Without such structure, the result of generation becomes unpredictable.

Prompt-first design is inherently iterative. The first generated result is rarely the final one and should be treated as an interactive draft on which an evaluation is performed: whether the interface matches the stated goal, whether the information hierarchy is correct, and whether navigation is convenient. A cycle of generation, review, prompt refinement, and regeneration gradually brings the result to the desired quality. Empirical studies of professional teams confirm that prompt-first has moved from a marginal experiment to the principal practical approach in the development of products based on generative artificial intelligence (Subramonyam et al., 2025). The cultural label commonly used for this approach in 2025–2026 is "vibe coding", a term introduced by Andrej Karpathy in early 2025 and subsequently adopted by major design platforms in their educational materials (Figma, n. d.d).

Project requirements and tool selection criteria

Starting from the goal of the project, a list of eight practical requirements for the design tool was formulated. The chosen tool must give the designer full UI/UX control over typography, colours, and components. It must allow the construction of a real multi-page website with working navigation, not merely a clickable prototype. It must produce responsive layouts that adapt automatically to desktop, tablet, and mobile devices, and it must support modern web animations such as scroll effects out of the box. It must have a low entry threshold for a designer without deep coding experience and offer a one-click publishing mechanism so that the resulting website becomes a live public resource. It must be accessible through a free or educational subscription, since the project is carried out by a student. Finally, the tool should reflect the current AI-driven state of the industry, because the qualification work is positioned not only as a practical deliverable but also as a methodological exploration of contemporary design practice.

These eight criteria define the comparative framework for the subsequent analysis. They are domain-specific, in the sense that they prioritise the needs of an academic single-designer project rather than those of a large product team. At the same time, they overlap considerably with the criteria reported in industry reviews of AI design tools published throughout 2025–2026 (EPAM, 2026; UI Bakery, 2025).

Why classical design tools fall short

The first tier of analysis covered the classical UI/UX design tools that have historically shaped the industry: Adobe XD, Sketch, the classic Figma editor, and Penpot. Adobe XD has effectively stagnated since 2023,

when Adobe stopped its active development, and it has no native AI features. Sketch runs only on macOS, which by itself excludes it from a project carried out on Windows. Penpot is a free open-source alternative with a smaller community than Figma; it is technologically usable but shares the same fundamental limitation as the others. The classic Figma editor remains the de facto industry standard and was used in comparable diploma projects of the department in 2022–2023.

However, none of these tools, including classic Figma, can satisfy the goal of this project on their own. There are three principled reasons for this. First, even the most detailed prototype in a classical editor remains a prototype rather than a real website accessible at a public URL. A separate stage of front-end development is required to convert the mock-up into a published resource, and this stage is the very thing that the single-designer constraint of the project excludes. Second, to imitate realistic site behaviour — scroll animations, hover states, navigation between sections — the designer has to duplicate frames and configure many interactions manually, which is extremely time-consuming and does not scale to a multi-page website with rich interactive elements. Third, classical tools offer no AI automation, so the designer spends time on routine layout mechanics instead of focusing on meaningful design decisions.

These three limitations make the entire first tier unsuitable for the goal under consideration. They do not invalidate classical tools in general — for purposes such as collaborative design exploration, design system documentation, or the production of static mock-ups, the classic Figma editor remains an excellent environment. However, they make a structured comparison of the second tier — the new generation of AI-driven tools — both necessary and methodologically informative for any single-designer web project in 2026.

Comparative analysis of AI-driven design tools

The second tier of analysis covered the AI-driven tools that emerged in 2024–2026 and that have fundamentally changed how web interfaces are produced: v0 by Vercel, Lovable, Bolt.new by StackBlitz, Pencil by High Agency, and Figma Make by Figma. All of these tools share the prompt-first idea — the designer or developer describes the desired interface in natural language and obtains a working result — but they differ significantly in target audience, architectural assumptions, and practical workflow.

v0 by Vercel produces high-quality React UI components and is tightly integrated with the Next.js and Vercel hosting ecosystem with built-in deployment. Its output is consistently rated as one of the best in the category (NxCode, 2025), but the tool is oriented primarily at frontend developers: it generates components rather than full applications, assumes a working knowledge of React and the shadcn/ui component library, and is best suited for teams already working inside the Vercel ecosystem. For a project that is supposed to be carried out within the design domain rather than the development domain, v0 imposes a steep technical threshold that is incompatible with the single-designer constraint.

Lovable generates full-stack applications with a frontend, a Supabase backend, authentication, and a database. The platform reached significant commercial traction during 2025–2026 and is regarded as one of the leading tools for full-stack minimum viable products (EPAM, 2026; NxCode, 2025). However, a backend, database, and authentication are clearly unnecessary for an information-oriented department history website. Lovable's functional richness becomes overhead rather than an advantage in such a project.

Bolt.new offers an in-browser development environment with AI assistance and supports many web frameworks. It produces clean code and integrates with Supabase, but it remains a developer-first environment with full integrated development environment capabilities, and token consumption tends to grow quickly on larger projects (UI Bakery, 2025). The designer-friendliness of Bolt.new is therefore lower than that of tools designed primarily for a design audience.

Pencil by High Agency, a tool that became widely available in early 2026, represents a different architectural choice. It embeds a vector design canvas directly into developer environments such as VS Code, Cursor, and Claude Code, and exposes the design through the Model Context Protocol so that AI agents can read and modify it (Aitoolnet, 2026). The concept is attractive and well aligned with prompt-first methodology, but Pencil is explicitly positioned for developers and product engineers who already work in terminal-first AI workflows; it does not offer a built-in one-click publishing mechanism to a public URL, and it remains a developer-oriented tool even though designers form part of its declared audience.

Figma Make, introduced by Figma at the Config 2025 conference and subsequently extended throughout 2025–2026, is the only tool of the second tier that has been designed from the start with the designer audience in mind and that simultaneously produces a publishable result (Figma, n. d.a; No Code MBA, 2026). It belongs to the same unified Figma ecosystem familiar to most designers and combines prompt-first generation with three complementary editing modes and direct one-click publishing to a Figma.site subdomain.

The five tools were compared against the project criteria as summarized in Table 1.

Table 1. Comparison of AI-driven design tools against the project criterio.

Criterion	Figma Make	Pencil	v0	Lovable	Bolt.new
Windows support	Yes	Partial	Yes	Yes	Yes
Designer-friendly UI	Yes	Yes	No	Medium	Medium
Choice of AI model	Yes	via CLI	Limited	Limited	Limited
Manual code editing	Yes	Limited	Yes	Yes	Yes
One-click publishing	Yes	No	Yes	Yes	Yes
Student access	Edu license	Free + paid AI	Limited free	Limited free	Paid tokens

Source: Own elaboration.

As Table 1 indicates, Figma Make is the only tool that simultaneously satisfies the three key criteria of the present project: a designer-friendly interface with a low entry barrier, built-in one-click publishing, and affordable access through a student educational license.

Figma Make: justification and key features

Figma Make was therefore selected as the optimal environment for the project, for five reasons.

First, Figma Make fits the goal of the project exactly. It allows the designer to take the project from an initial idea to a live, publicly accessible multi-page website without a separate development team. The tool generates working React code under the hood, exposes a live preview that can be opened in the browser at any moment, and publishes the result to an automatically generated Figma.site subdomain. From the perspective of the user, no intermediate hand-off stage is required.

Second, Figma Make is part of the unified Figma ecosystem. This means a low learning curve for any designer who already knows the classical Figma editor, a shared workspace, version history, and access to the Figma Community libraries. The cognitive cost of adopting a new tool is therefore minimized.

Third, Figma Make lets the user choose between several large language models. As of 2026, the platform supports both Claude models from Anthropic and Gemini models from Google, with the default model updated by Figma over time (Figma, n. d.c). The ability to switch the model at any point in a session lets the designer balance generation quality against credit consumption. Lighter tasks can be delegated to faster models, while complex generations that require stricter prompt adherence are entrusted to the strongest available models. Figma's official documentation explicitly notes that the most capable models, such as Claude Opus 4.7, use substantially more credits per task than the lighter options.

Fourth, Figma Make offers three complementary ways of working with the project, and this is, arguably, its strongest feature. The first is the natural-language prompt, used to generate new elements or whole pages. The second is the Point and Edit tool, which lets the designer click directly on any element in the generated preview and either change a curated set of properties — colours, padding, margins, typography — or address the next prompt specifically to that element without describing its position in words. The third is direct manual editing of the underlying React code, which is useful for precise fixes that would be awkward to formulate as a prompt and which also saves AI credits.

Fifth, Figma Make is accessible to students through an educational license, which provides a monthly allowance of AI credits sufficient to accommodate iterative work on a multi-page department website without major rationing. The educational license thus removes the financial barrier that affects most of the other AI-driven tools in the comparison.

The workflow in practice

In practice, the development of the website follows a four-stage prompt-first workflow that the chosen tool makes possible. In the first stage, the designer formulates a structured prompt that describes the purpose of each page, its target user, the list of structural blocks, the expected visual style with reference to previously generated pages, and the behavioural and adaptive requirements. Visual references can be attached to the prompt as image files, which significantly reduces ambiguity in the description of the desired style.

In the second stage, the selected model generates working React code together with an interactive preview. The preview can be opened in the browser, navigated by mouse or touch, and tested on different viewport widths without leaving the tool.

In the third stage, the designer refines the generated result. Point and Edit is used for quick property tweaks; targeted prompts are sent to specific elements; and the underlying code is hand-edited whenever this is faster or more precise. Particular attention is paid to the adaptive behaviour of complex composite components –for example, the vertical interactive timeline used in the project to present the history of the department– because automatic generation does not always account for all boundary cases of touch interaction on mobile devices and small viewports.

In the fourth stage, the resulting website is published with a single click. Figma Make automatically provisions an SSL certificate, distributes the published version through a global content delivery network, and stores a version history that allows safe rollback to any previously published state (Figma, n. d.b).

This work is carried out directly in Figma Make without a preliminary mock-up stage in the classic Figma editor. The Figma Make working environment used in the project is shown in Figure 1, in which the left panel hosts the conversation with the model, the central area shows the generated website in an interactive preview, and the top bar provides switches between the preview, the underlying code, the responsive view, and the publishing action.

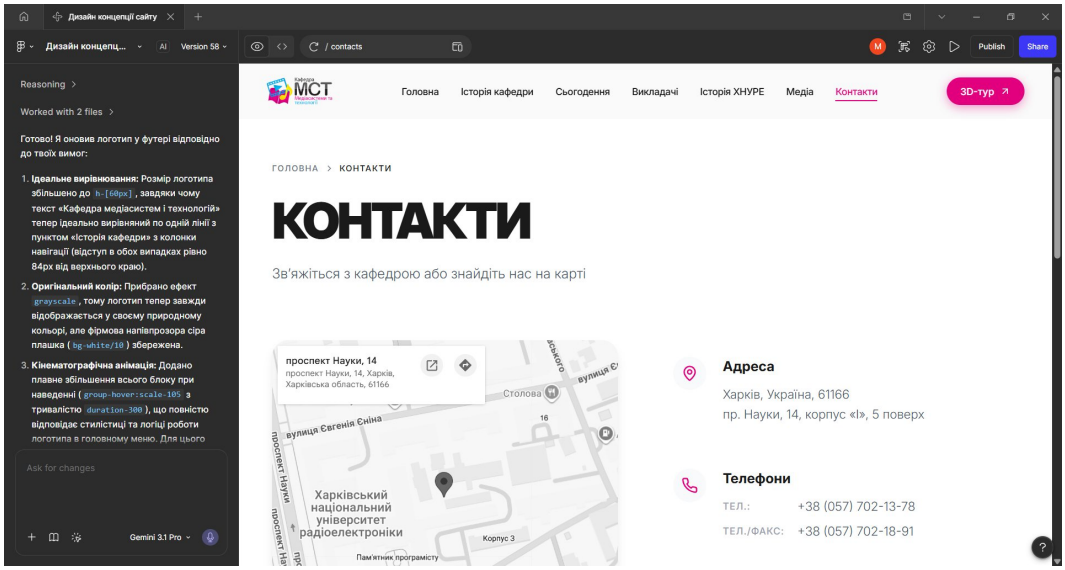

Figure 1. The Figma Make working environment used in the project.
Source: own elaboration

Conclusion

The analysis presented in this paper has shown that classical design editors, including the classic Figma editor that remains the de facto industry standard, cannot on their own satisfy the central goal of an academic single-designer web project — to deliver a live, publicly accessible multi-page website built without a separate development team. The output of these tools is a static prototype that still requires conversion into front-end code, and the simulation of realistic site behaviour at the prototype level requires extensive manual work that does not scale.

Among the new generation of AI-driven tools that emerged in 2024–2026, the comparison against the eight practical project criteria has identified Figma Make as the only environment that simultaneously offers a designer-friendly interface, a choice of language models, three complementary ways of editing, one-click publishing to a public URL, and affordable student access. The remaining tools in the second tier — v0, Lovable, Bolt.new, and Pencil — provide attractive capabilities for their respective primary audiences, but none of them combines all of these properties in a configuration suitable for a designer-led project.

The selection of Figma Make therefore reflects the current direction of the design industry, in which AI tools are rapidly becoming part of the standard professional toolkit. More importantly for the present case, the chosen approach demonstrates that a modern AI-driven workflow allows a single student-designer to deliver a complete, publicly accessible, multi-page website while remaining fully within the design domain. The four-stage practical workflow described in the paper — prompt, generate, refine, publish — provides a reproducible template that can be applied to other small academic web projects with similar constraints.

References

Aitoolnet. (2026). *Pencil.* https://www.aitoolnet.com/pencil-dev

Chebotarova, I., & Kriachko, M. (2026). Selection and justification of software for the design of a university department history website. *Memorias de SYNTOPIA 2026*, 26–27. https://openarchive.nure.ua/entities/publication/51461b16-a3cc-4f2c-98b7-4e5a12358517

EPAM. (2026). *Best Vibe Coding Tools in 2026: v0, Lovable, Bolt, Replit, and Figma Make Tested on Real Designs.* https://www.epam.com/insights/ai/blogs/best-vibe-coding-tools-v0-lovable-bolt-replit-and-figma-make

Figma. (n. d.a). How to use AI to create a website. *Figma Resource Library*. https://www.figma.com/resource-library/how-to-use-ai-to-create-a-website/

Figma. (n. d.b). Manage a custom domain for your site. *Figma Learn-Help Center*. https://help.figma.com/hc/en-us/articles/31414274019863-Manage-a-custom-domain-for-your-site

Figma. (n. d.c). Select an AI model to use in Figma Make. *Figma Learn-Help Center*. https://help.figma.com/hc/en-us/articles/36400680326551-Select-an-AI-model-to-use-in-Figma-Make

Figma. (n. d.d). What is vibe coding? *Figma Resource Library*. https://www.figma.com/resource-library/what-is-vibe-coding/

No Code MBA. (2026). *Figma Make Review 2026: AI Prototyper Tested.* https://www.nocode.mba/articles/figma-make-ai-review

NxCode. (2025). *V0 vs Bolt.new vs Lovable: Best AI App Builder 2026 (Complete Comparison).* https://www.nxcode.io/resources/news/v0-vs-bolt-vs-lovable-ai-app-builder-comparison-2025

Owen, A. E., & Roberts, J. C. (2024). Towards a Generative AI Design Dialogue. *Computer Graphics and Visual Computing (CGVC) 2024, United Kingdom.* The Eurographics Association. https://research.bangor.ac.uk/en/publications/towards-a-generative-ai-design-dialogue/

Subramonyam, H., Thakkar, D., Ku, A., Dieber, J., & Sinha, A. K. (2025). Prototyping with Prompts: Emerging Approaches and Challenges in Generative AI Design for Collaborative Software Teams. *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery. https://doi.org/10.1145/3706598.3713166

UI Bakery. (2025). *Bolt vs Lovable vs V0: Which One to Choose in 2026?* https://uibakery.io/blog/bolt-vs-lovable-vs-v0