

[Why no-code tools are not the right choice No-code App Development: The Future, The Limitations, and The Possibilities](https://aveo-software.tumblr.com/post/683136681575071744/why-no-code-tools-are-not-the-right-choice-no-code)



No-code app creation has sparked numerous disputes in various tech forums among programmers, designers, entrepreneurs, and employees all over the world. Is it, however, the best alternative for you?

Coupled with the widespread use of [**minimum viable product (MVP)**](https://href.li/?https://aveosoftware.ca/mvp-development/) development methodologies, most of these online communities promote no code’s quick validation capabilities as they look forward to the no-bright code’s future.

So, the most important question to ask yourself is not whether or whether you should use no-code app development, but rather why you shouldn’t. Many people see the benefits of no-code, but the disadvantages are less obvious, despite their importance.

Sure, you’ve probably heard one or two names of folks who have successfully started no-code projects. However, countless others have become stuck, and their projects have never materialized, despite having brilliant ideas.

So there’s still a lot to debate here, but first, let’s define the term “no code"—what does it mean?

**What Is No-Code?**

No-code is a software development strategy that uses visual flows and tools with features like drag-and-drop, templates, and custom resources like graphics to allow programmers and non-programmers with little or no coding abilities to quickly construct apps.

No-code platforms like **[Webflow,](https://href.li/?https://webflow.com/)**[**Bubble**](https://t.umblr.com/redirect?z=https%3A%2F%2Fbubble.io%2F&t=NDczNWY2NzVlNDBmNDY1NzFkNzQwMGQyYzY4NDMzMzYxNjJiMTdlOSxoY2F4eTVQaw%3D%3D&b=t%3AeZhBdYvF1KqudlVAuffGeg&p=https%3A%2F%2Faveo-software.tumblr.com%2Fpost%2F683136681575071744%2Fwhy-no-code-tools-are-not-the-right-choice-no-code&m=1&ts=1655301226), and **[Zoho Creator](https://href.li/?https://www.zoho.com/creator/)** have a large following among non-techies, designers, and developers because to their broad appeal. Users of no-code platforms are no longer burdened by long hours of app development, arduous development duties, or[**intolerable coding methods**](https://href.li/?https://en.wikipedia.org/wiki/Coding_best_practices)that they couldn’t keep up with.

Low code is commonly confused with no code, which is a similar but not identical word. In contrast to the low-code software development strategy, which requires basic coding knowledge, no code allows you to construct a working application without writing a single line of code.



**No code’s possibilities: things that can be done with no code**

That would be so wrong—for sure—to discard no code and declare it useless. But how many things can you truly construct without using code? Is it ten, twenty, or one hundred? Or even more? Certainly, there are a plethora of them.

It’s critical to note that, while there are a lot of options with no code, as your programme grows, you may run into a lot of no-code limits. So, let’s go into the specifics of what’s 10 times easier to build without coding.

Apps that are well-suited to no-code platforms include those that demand minimal interaction, as well as those that are basic with predictable and finite app features, as well as those with one-way data flow. Some examples of these apps are as follows:

* Product prototypes
* Landing pages
* Single-page sites
* MVPs
* Landing pages
* Micro SaaS
* Communities
* Simple eCommerce solutions
* Simple automation
* Knowledge base
* Journaling app



**No-Code’s Limitations: The Things That Cannot Be Done With No Code**

While many organizations and aspiring developers appreciate the enticing benefits of the no-code method, it’s crucial to be aware of the restrictions and drawbacks that come with it.

If you need to create[**scalable and sophisticated software solutions**](https://href.li/?https://aveosoftware.ca/mobile-development/) for your organization, no-code development becomes less appealing. Custom app solutions necessitate one-of-a-kind apps that can adapt to shifting needs and [**trends in app development.**](https://href.li/?https://aveosoftware.ca/the-ten-top-software-development-trends-in-2022/)

So, if you’re curious about what you can’t achieve with no-code technology, have a look at the following examples.

**Apps with Complex Functionality**

No code is not an option if you need a high-performing software with advanced user experiences. Because of its limits, this technology may not allow you to create a fully personalized product.

The more you use no-code platforms to build and extend your product, the more custom elements your app will require, and the less you’ll need their all-in-one solutions, which were formerly useful when the app was still young.

Consider[**employing specialist software developers**](https://href.li/?https://aveosoftware.ca/a-comprehensive-guide-how-to-hire-the-best-app-developer/) for your project if you need to create an app with a lot of interactivity and feature.

**Highly scalable apps**

You won’t be able to design an app with no code if you want it to increase in features over time. If you create an app with a feature that isn’t available in your no-code platform’s toolkit, you’ll become stranded, making it more difficult to expand.

While some of these limits may have workarounds available with higher subscription options, they are frequently unavailable, complicated, or expensive. Because your scalability is more dependent on your vendor than on your demands, no-code will continue to be a pain as your project outgrows your platform.

**Apps that require huge integration options**

Another major drawback of no-code platforms is their restricted integration capabilities, making it impossible to create an app that requires several bespoke connectors with various apps and services.

Even with the most basic functionality, many no-code systems do not provide smooth interfaces. That’s a logical assumption. The main goal of such platforms was to allow non-technical consumers to utilize them, and integrations may complicate their system, therefore it was never a possibility for them.

If you choose with no-code, your platform might not support integrations, which means you won’t be able to link your favorite apps and services. If it does, you may need to pay a developer to create a complicated and [**specific API**](https://href.li/?https://docs.oracle.com/en/cloud/paas/mobile-autonomous-cloud/get-started-custom-apis/index.html) for your app, which could be costly.

**Apps with full code ownership**

With no code, you do not own the assets. Even if your platform may allow you to export your application’s assets, it often comes with added costs and challenges like unreadable code.

Some core features of your application may be tied to your platform, which may make your application’s assets useless without using the platform.

This inability to move freely across platforms has become a great disadvantage of no code. Thus, the main limitation of no code is [**vendor locks**](https://href.li/?https://www.cloudflare.com/learning/cloud/what-is-vendor-lock-in/)**:** the dependence on a platform throughout the lifetime of your app.

**Apps with great user experiences**

User experiences are minimal or terrible with no-code app development alternatives. The majority of these systems are focused on delivering speedy results, which may significantly impact the user experience.

These platforms may be useful yet restricted if you wish to plan and create complicated user interfaces. As a result, you’ll require developers at some points during the design process if you want a robust and complicated application with a great user experience.

**High-end prototyping**

[**Prototyping**](https://href.li/?https://www.interaction-design.org/literature/article/design-thinking-get-started-with-prototyping) is a feature of many no-code tools. While they are suitable for small apps and some MVPs, they lack the ability to simulate the real-world experiences of your desired product.

Furthermore, combining design and development in one location may not be the best option for creating complicated apps that require deliberate and compelling user experiences. So, unless your prototype design requires a high level of intricacy, you won’t need to use any code.

**Apps with sensitive data**

In a no-code platform, your data is usually hosted on the vendor’s server. As a result, you have no control over the security of your source code or databases. When you don’t have control over your data, you don’t have control over how it’s accessed or stored.

You’re pinning your hopes on your vendor, who may decide to stop performing upgrades and maintenance, resulting in security breaches and data loss. If ensuring the privacy and security of your data is a top priority, these platforms may not be the best choice.

**Apps with full control**

Building your app on a proprietary platform, unlike [**custom software development**](https://href.li/?https://aveosoftware.ca/custom-software-development/), does not give you inside-out expertise of your app. This limitation is common because your app’s whole implementation is secret and restricted.

Even if the source code for your programme is available, it lacks sufficient documentation. Their coding structure might be complicated and unmaintained, making it difficult to grasp.