



The **Nocoded** guide  
to **launching** your  
**MVP** without **coding**

 **nocoded**

# Intro

There is nothing worse than going to google, typing a question, and finding hundreds and hundreds of answers. You open the first three links and realize that they don't help you at all, and the more you read, the more questions you have. In the end, you end up in a vicious circle of questions and answers about the same topic again and again.

This situation we just described is a fairly common problem for startup founders, especially those with little or no business experience. **They just come up with the idea that it will change the world but don't know where to start to see it through.**

We want to save you hours of google searches, Youtube videos, and possible mistakes starting your project and launching your MVP. So we have decided to use our experience working with startups and no-code platforms to summarize everything in one e-book: **The Nocoded guide to launching your MVP without coding.**

Now, let's get down to business.

## What will you find in this ebook?

We will present you with a complete road map from having your brilliant idea to launching the first version of your MVP to the market. At the beginning of this book will find a definition of what an MVP is and why it's necessary. This is probably something you have already read a lot of times. Still, **we will give you examples of other ways to do an MVP, validate your idea, and collect feedback.**

**The second part will answer HOW to design your MVP.** We will give you a step-by-step guide of all you must do before sitting down to develop. **For example: what platform will the application be for?** What will the workflow be, and what are the key features you must include to generate value for early adopters.

Finally, the third part will be an impartial comparison of different no-code platforms to create MVPs. **We will tell you the advantages and disadvantages of each one, their characteristics, and what platforms they work best for.**

This way, you will have all the information you need to start developing your project without needing a professional team or knowing code.



# What's an MVP?

Every startup founder knows the answer to this question. However, that answer from a google search does not explain anything in detail and leaves much room for interpretation.

To start talking about the subject, we have to mention the **Lean Startup Methodology of Eric Reis**. He was the first to talk about this concept and revolutionized how technology companies bring new products to market.

**Reis discovered this methodology after years of working with technology companies seeking to innovate and bring new products to market.** Before, when a company wanted to create a new product, they talked with their marketing experts and team and decided what they believed the market wanted and how they would do it.

After making this decision, the technical team spent time and money developing the final application to be released when finished. **The problem was that if the application didn't please the customers, failure was resounding.** A clear example of this was the second life-type chat platform **IMVU**.

## What's an MVP?

---



Let's take a trip back in time to 2004. Then, chat platforms were the closest thing to today's social networks, and everyone had an account in at least one of them. Parallel to this, Second Life, the precursor platform of the metaverse, was also in its prime. So the people of IMVU came up with an idea: **What if chat platforms could be combined with Second Life 3D avatars to create a chat where you can interact with your friends digitally?**

So far, the story doesn't sound too bad. The executives liked the idea, and they started to develop the platform. The work took six months, and the most common chat platforms were integrated. Finally, the application was launched, and it failed miserably. No one understood why. People liked to chat, and Second Life was trendy. IMVU decided to call in some users and test them while interacting with the platform to discover what had gone wrong. **The result was that users wanted to talk to everyone within the platform.**

They were more interested in talking to strangers than their friends using their 3D avatars. Finally, after the feedback, they pivoted and became an open world chat with 3D avatars.

**After IMVU, Reis realized that there had to be an efficient way to validate a product without spending so much time and money developing the final version. The MVP concept was born.**



# What is the **Lean Startup Methodology?**

Reis defines this methodology as

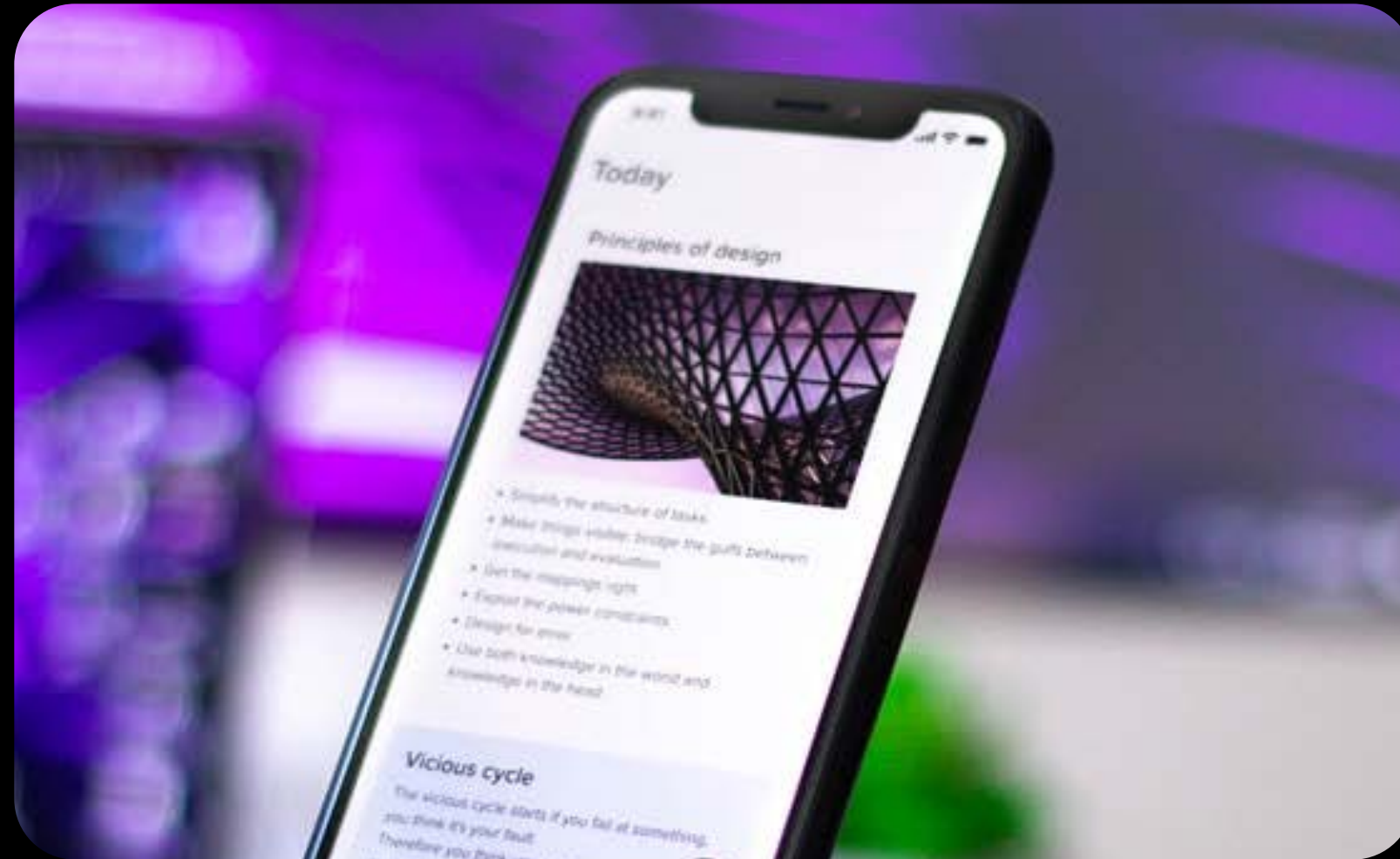


**"a set of practices for helping entrepreneurs increase their odds of building a successful startup."**

Lean Startup allows you to create a product or business through experimentation, validating the hypotheses proposed to achieve the final product. This way, you will be able to define and **shorten the development cycles by obtaining the target audience's opinion and adapting the project to their needs.** In other words, Instead of making complex plans based on premises, it allows continuous adjustments.

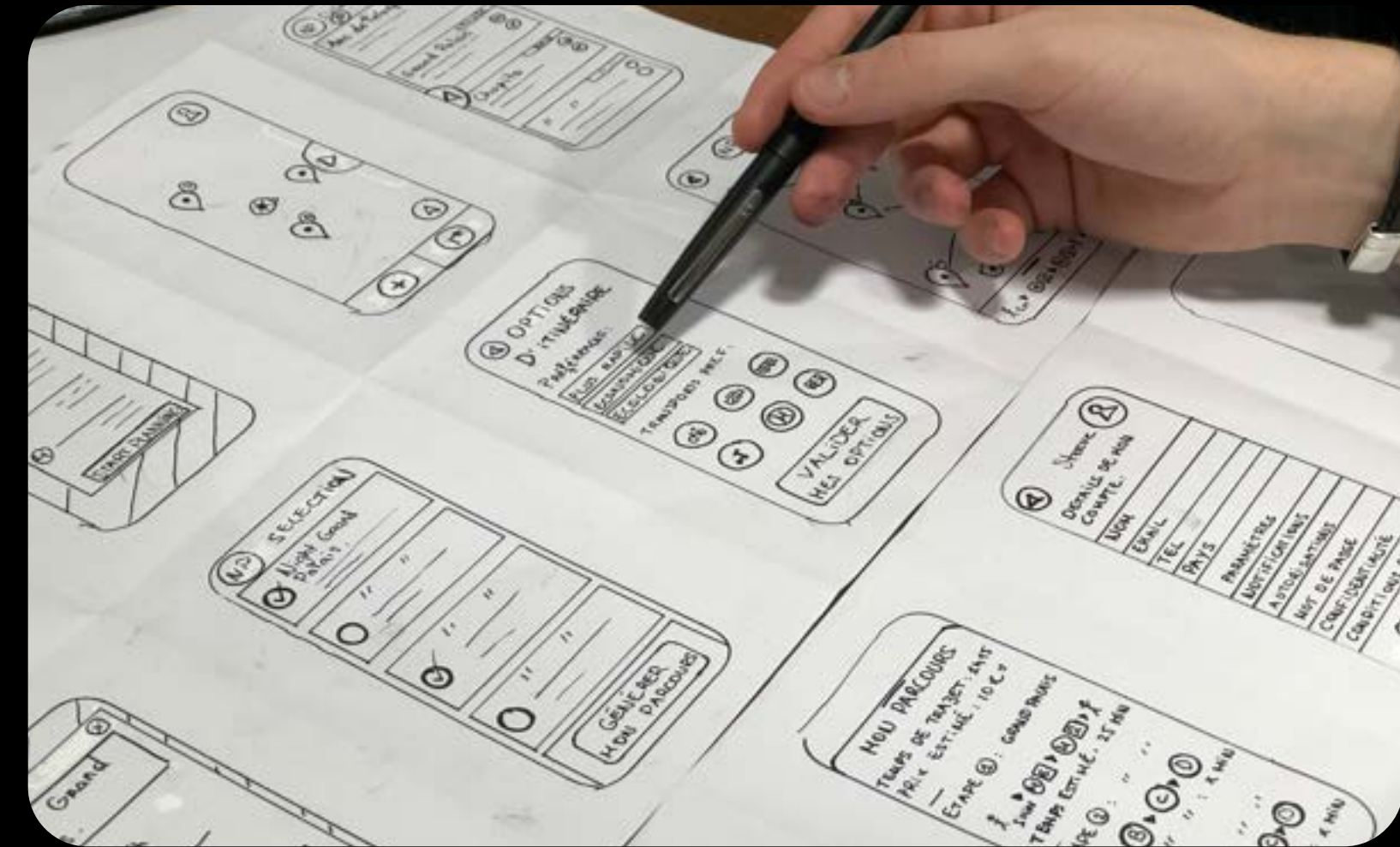
However, the Lean Startup Methodology is not a success guarantee. Instead, it makes failure much cheaper and the risk lower by testing simpler versions of your solution. These prototypes exposed to the client are called **Minimum Viable Product (MVP).**

# The MVP



It's a basic version of your product with essential functionalities to test the public's reaction concerning the idea, product, or service. Knowing the Key features to make your minimum viable product **helps development teams avoid wasting resources on something no one wants or needs.**

This process shouldn't require too much effort. Thus, if the hypothesis is wrong, it's easier to pivot (change the business strategy) from the initial idea, proposing alternatives closer to what the public wants. So here we share some aspects that you should consider when **designing your MVP.**



## Design

It can be simple, but the user experience must delight customers and make them feel comfortable using the app and coming back.

## Usability

Your app needs to give enough value that people are willing to use it or buy it. In other words, it has to be useful for your target audience.

## Functionality

Your minimum viable product must have all the necessary features to solve a specific consumer problem. This is the hardest part to achieve because it's normal to not know when is enough and want to keep adding things to make your solution better.

# Different types of MVPs

There are different ways to make an **MVP**. The model you choose depends on factors such as how advanced your business model is, the number of resources you have, and how sure you are that your idea will sell.

## 1. Low Fidelity MVP

Lo-Fi minimum viable products tend to be easy to **develop and deliver basic results that help you understand consumer needs**. Your goal is to explore what type of solution would be most effective for customers.

In most cases, these **MVPs** do not even remotely resemble the final solution. **For example, a well-known case of a Lo-Fi MVP is DropBox.**



Before even developing, the founders **made a video explaining to their potential users what the platform's functionalities would be to validate their idea.**

## 2. High Fidelity MVP

Hi-Fi minimum requires **further development and resembles the final solution**. However, these MVPs have already validated most of the business model and are close to becoming the real thing. Their function is to **validate if the client is willing to pay for the solution and how much**. They also allow defining and optimizing marketing strategies in the value proposition and communication channels.

This minimum viable product **requires a lot more time and effort, but it is an effective way to check if you have a desirable product or service** before you build it completely, for example, with digital prototypes.

**An excellent example of this is Zappos.**

Nick Swinmurn knew that people would buy shoes, but I wasn't 100% sure that HIS business would work. So instead of buying stock...



**He created a web page and uploaded the photos of the shoes.**

Then, every time someone asked him for a pair, he would go to the store himself, buy them, and then send them to his customer.

**Finally, remember that an MVP seeks to verify that the product effectively solves a market need before investing too many resources in its development.**



# Step to step **MVP development without coding**

Contrary to what everyone believes, it is not necessary to know how to code for founding a startup, not is it to beg your developer friend to become a co-founder and help you with making your application.

Reid Hoffman said:

**“If you are not embarrassed by your first product, you launched too late.”**

The idea of going to market quickly, testing, pivoting, etc., is repeated again and again everywhere. Still, it is not that simple. In most cases, founders don't know how to do something simple. **So we will explain in detail all the steps you must follow to develop your MVP without having to write a single line of code.**



# Steps to develop your MVP without code

## 1. Comprehensive research

Before embarking on the development of your **MVP**, you should ensure that your idea fulfills the target users' needs. **The best way to do this is to do product/market research that gives you information on the possible customers and their unmet needs.** This can be accomplished by conducting surveys or market research.

The more information you have about the field, the higher the chances of success.

### Some of the questions you need to answer at this stage include:

#### What specific problems should your MVP solve?

The idea is to understand the product's value and how customers will benefit from it.

#### Are there other solutions to the problem, and what are they?

This will allow you to understand the direct and indirect competitors and find some added value you can give to your product, so the public prefers it.

#### What is the ideal user that will be interested in your product?

There are always specific categories of users with custom needs and requirements to help you make the product as user-oriented as possible.



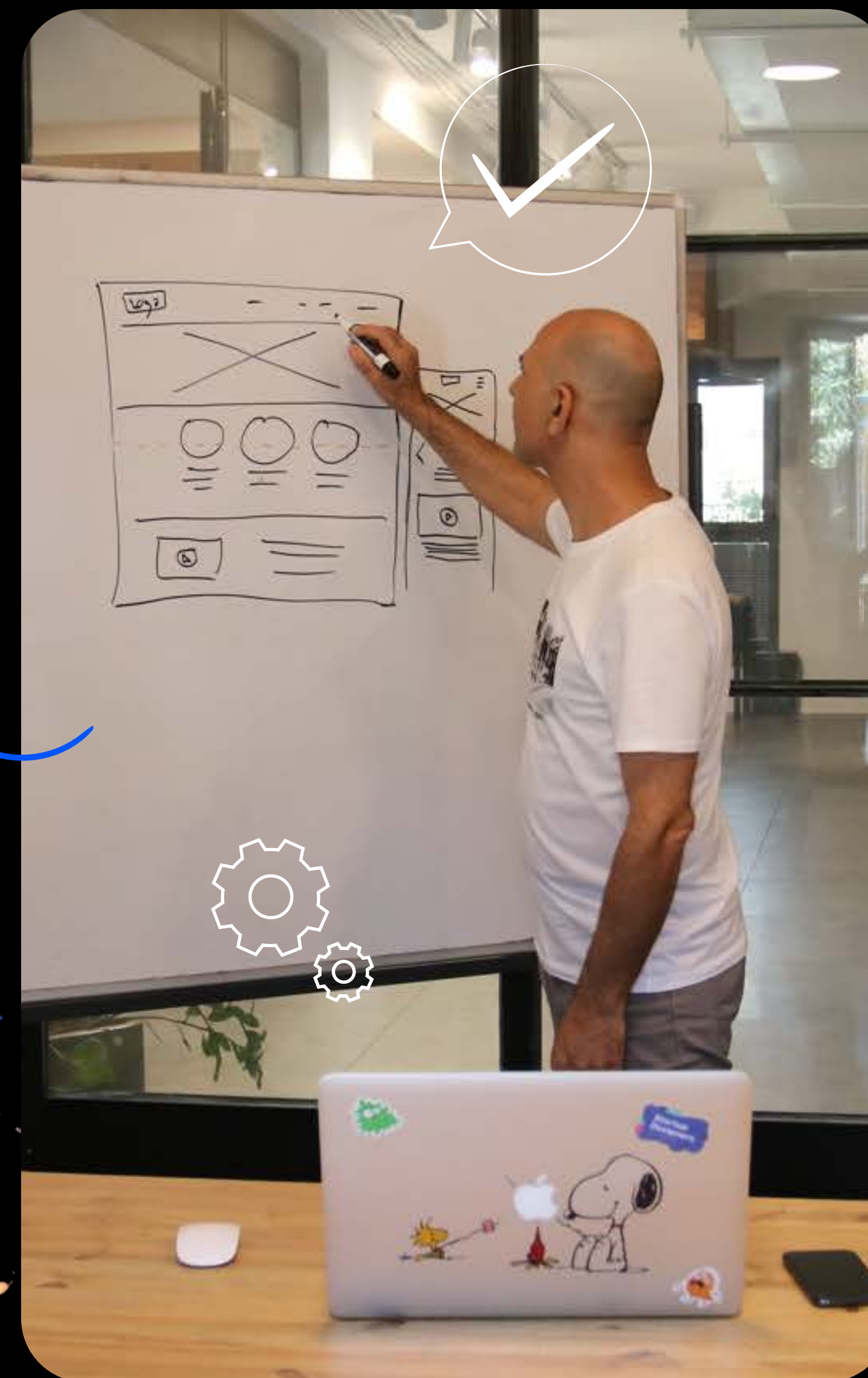
## 2. Identify and prioritize features

Once the research is done, the next step is to identify and prioritize the features you want to have based on their importance. Your MVP should have only one top-priority feature that conveys the product's core value.

To know what features to prioritize, ask yourself questions such as:

**What do the users want? Is this product offering them something beneficial? Etc.**

Finally, you need to arrange all features in the product backlog (priority-wise). Then, When building your **MVP**, you start with the top priority functions.



### 3. Select an approach for your MVP

At this point, you need to think about what approach is most appropriate for your project.

#### Here is a list:

##### The prototype approach

The idea is to create a first version product to help you understand the target market's requirements. It will help you know if people are willing to subscribe to this idea and pay for it. With this kind of MVP, you can reach a maximum of people with minimum effort.

##### Visualization of ideas

In this approach, you test your business hypothesis using targeted marketing campaigns. The idea here is to represent or explain what the product will look like and what it will do.

##### Sell first, build later

The premise of this approach is based on pre-selling a product before building it. This can be done by launching a crowdfunding or pre-order campaign. If this method succeeds, it will allow you to validate your idea and raise funds, which allows you to envision multiple revenue-generating possibilities for your product.

##### Single feature

As the name suggests, the product is developed based on its most essential feature. Since customers need to understand what it is intended for, the focus should be entirely on core functionality. The idea here is to build an MVP that can reduce user efforts by **60-80%**.

##### Chat concierge

Concierge MVP it's a real-time chat that allows you to talk to your users and learn about their needs and opinions regarding your product. This approach aims to validate whether they need the product you're offering.

##### Landing page

As its name indicates, this **MVP** type is a web page where you fully describe your product or service and some of the benefits of using it. In addition, it features a CTA-like button that allows interested visitors to click to read more, join a mailing list, buy products, or take some other action.

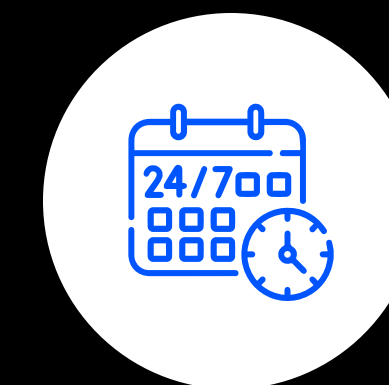
##### Email MVP

You can start by manually creating a few emails to check if the response is favorable. In case of a positive answer, you can proceed to develop the related product.

### The choice of the type of MVP depends on several factors, including:



**Needs of  
your users**



**Time  
available**



**Budget  
you have**

### 4. Develop a road map

After investigating, prioritizing features, and deciding on an approach to your **MVP**, **preparing a roadmap is the next critical step**. Ideally, the product roadmap should consist of four components, including:

- **Objectives:** To underline the fundamental vision of a product.
- **Activities:** The goals specified above can be achieved by performing specific actions.
- **User Stories & Tasks:** The activities require the implementation of tasks and features that can be converted into user stories.

### This design process is a vital MVP stage

You need to look at the app from the users' perspective, starting from opening the app to the final process. In addition, **user flow is an important aspect to consider** because it ensures nothing will be missed while keeping the future product and its user satisfaction in mind.

The focus should be on basic tasks such as finding and buying something or managing and receiving orders. These are the end-users goals while using the solution.



### 5. Identify the success criteria



Very well, almost everything is ready to start the most fun part of the process: developing. But first, you need to establish the success criteria of your application. You have to specify the most actionable metrics and success criteria.

**Here is are some examples:**

#### **Number of downloads**

Although this metric can be misleading at times and should be accompanied by others, it's essential for mobile apps. The simple logic here is that the more downloads your app has, the more popular it will be.

#### **Percentage of active users**

The number of downloads can be deceiving. Instead, you should check how many users are active and think about how to turn passive users into engaged ones.

#### **User engagement**

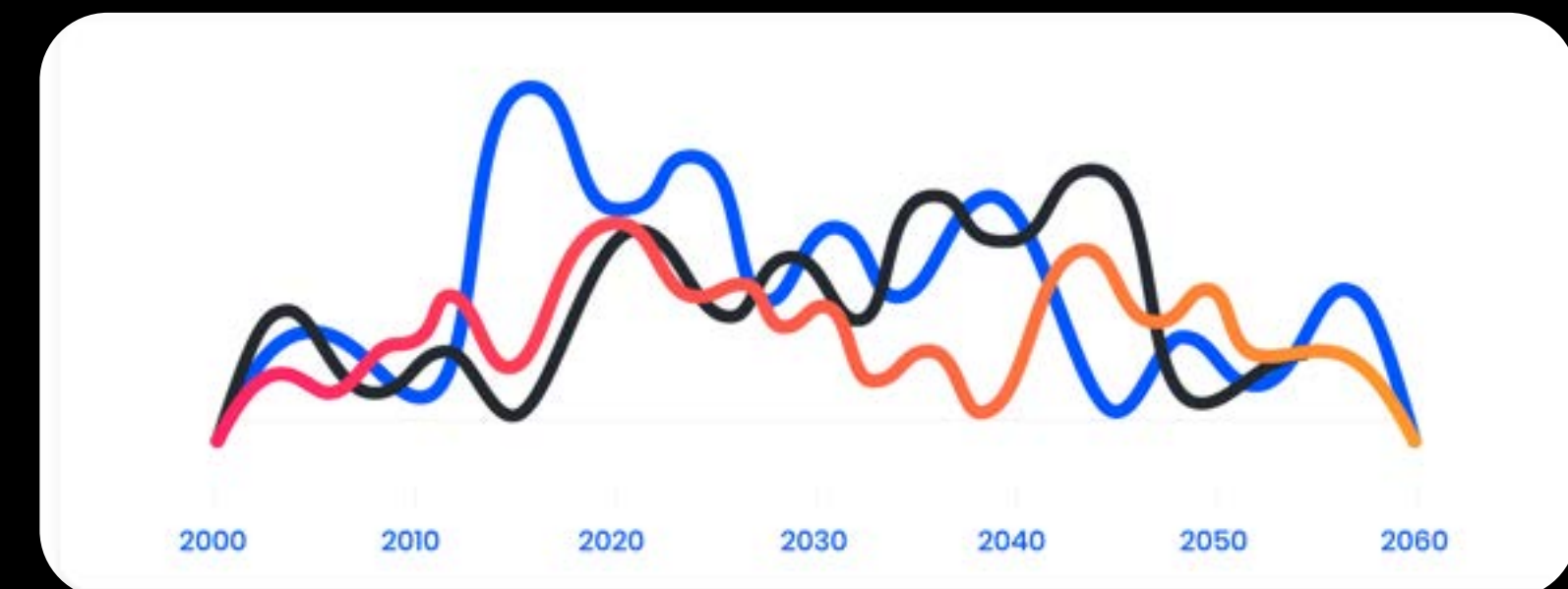
This is especially important as it symbolizes how much time users spend in the app, if it fulfills its objective, and helps assess its true potential. It can also indicate how well designed your UX is and if it needs modifications.

#### **Customer life**

This metric helps you understand how long users spend using your software before removing it.

#### **NPS**

NPS or Net Promoter Score is essentially a survey-based metric where you have to ask users directly about the usefulness of the MVP.



**Now that you have an idea of what criteria you're going to use to evaluate the success of your solution, it's time to move on to the best part, development**

## 6. Put the methodology into practice: Build, Measure, Learn

It's time to start with the development. Don't run after perfection. **Fast to market is better than perfect to market.**

After having your first MVP ready, the iteration process begins. Now you have to measure the impact of your product to determine the high and low points of your launch. Remember that you already set up success criteria to see if your product is working.

This process is repeated many times, as many times as necessary.

**You analyze the data, make the required improvements, and repeat.** Sometimes you may need to re-evaluate some or even all of the 6 points above during this process.

Not all **MVPs** are first-time hits, and not all business models are the right approach to a problem. That is why it is essential that you do not fall in love with your developments and do not insist on them if you see that the data is not giving results.

**Instead, the best thing to do is fall in love with the problem, pivot, and look for the solution in another direction.**





# What tool will **match my MVP needs?**

We come to the essential part of this e-book.

Everything we have told you so far is part of the methodology that every startup should know and apply regardless of if they have funds for a development team, technical knowledge, or not.

**This section is dedicated exclusively to these entrepreneurs who don't have the technical knowledge or funds to hire a company to do this job. Here we will talk about the no-code platforms we suggest to build an MVP.**

As its name suggests, no-code is a visual approach to developing digital products that allow you to make applications by dragging and dropping elements and doesn't require any development experience. **Those platforms also speed up the process up to 40%**, and what could take months can be done within days.

Okay, all that sounds fantastic, and I imagine that now you are about to close this e-book and go to google and search "**no-code tool to develop an MVP**"; well, this is not that simple. There are dozens and maybe hundreds of no-code platforms on the internet. They all have different focuses and strengths and will serve you more or less depending on the needs of your **MVP**. **Therefore, we will save you hours of research or a mistake in choosing your platform by giving you here a recommendation of six no-code platforms to develop MVP and its features.**

# Glide

**Web:** [www.glideapps.com](http://www.glideapps.com)  
**What can you make:** Mobile apps and web pages



Our first platform for building a no-code solution is Glide. This platform specializes in creating PWA mobile applications and web pages.

This platform makes a big promise to its users **"create an app from a Google Sheet in five minutes, for free."** But is it really true? This platform is considered one of the simplest of all. That is why it is highly recommended for founders with no experience and little time to learn.

The learning curve is probably Glide's greatest strength. It has a visual interface where you can see all the changes made while designing the application. Therefore, if you are not one to watch tutorials and like to "play around" with the app, you will understand how it works pretty quickly. In addition, Glide offers many free and paid app templates you can customize with a few clicks.

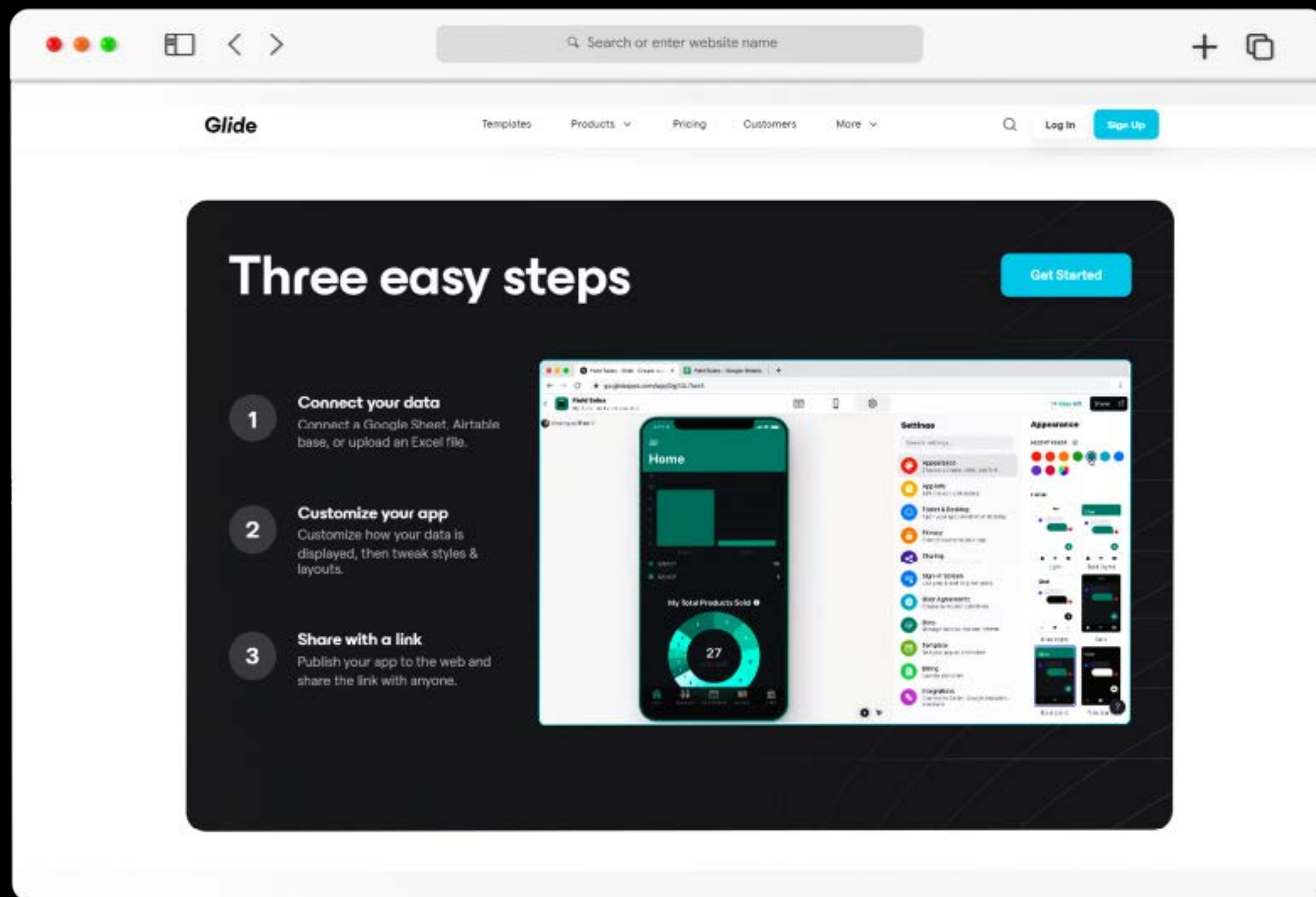
Similarly, you only need a google spreadsheet as a database when creating an application without using a template. **Glide scans it and gives you a draft of the app that you can easily edit.**

The builders' interface is not "drag and drop," making it even easier for people with no experience to get used to the platform. You can modify the widgets and functions by clicking on a series of established features. **At the same time, the changes are applied on the main screen, where we can see what our application looks like.**

This is the platform with the shortest learning curve of all that we will list, and it's also the fastest when it comes to launching a functional application. **The average time to make an app is 40 minutes, but it can take shorter if you use pre-existing templates or a very well-organized spreadsheet.**

# Glide

**Web:** [www.glideapps.com](http://www.glideapps.com)  
**What can you make:** Mobile apps and web pages



Although the learning curve is short and the possibilities of creating different apps are good, the scalability offered by the platform is very limited. For example, the maximum number of data entries in a free plan is 500 rows, and the Pro App plan only allows us 25000. **As a result, if the app grows too much, you will quickly be forced to migrate it to another platform.**

Another of Glide's low points is its customization possibilities. Although it's possible to change the colors, you can only choose from already pre-selected ones and upload your logo. **Not even with the payment plans can you make big changes, which affects your users' experience.**

Likewise, there is no way to create costume features or functionalities, but it does allow integrations with Zapier and access to APIs.

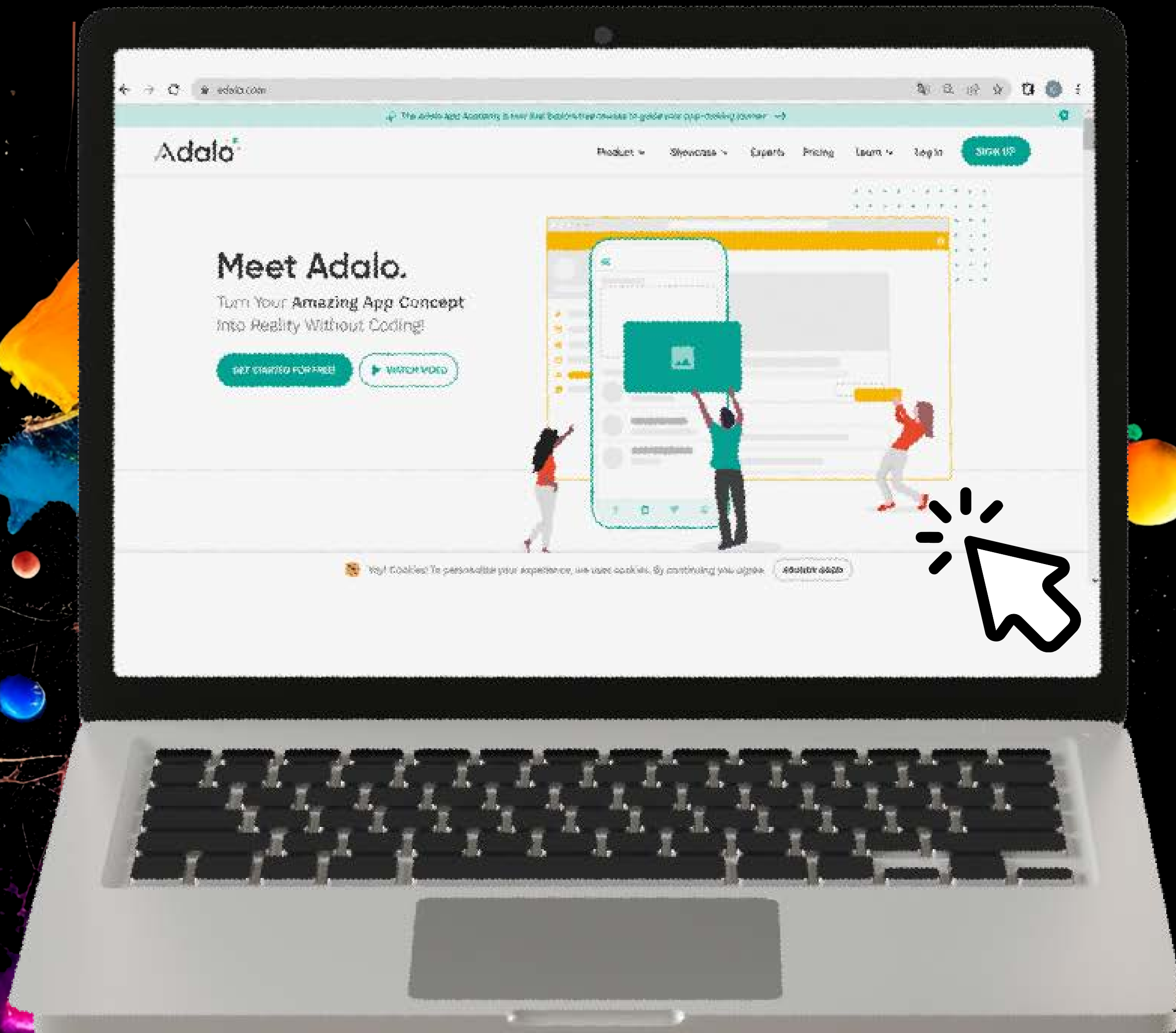
On the economic side, Glide is not expensive, but you need to evaluate its limitations before engaging economically. Its free version offers all the necessary features to create an app. The paid versions only offer upgrades that can improve the scalability of your solution, such as more storage space, data rows, and integrations. **There are 3 payment plans: Pro App: \$32 per month, Private App: \$40 plus \$2 per extra user per month, and Enterprise Apps, where the price is calculated according to your needs.**

Although they have **several tutorials on their page and youtube channel** about how to use the platform, the different citizen developers' communities are constantly publishing new content and tutorials in different languages, a good alternative for those who have difficulties understanding English. **It's even possible to find free and paid courses about how to use the platform on Youtube, Platzi, and Udemy.**

# Adalo

**Web:** [www.adalo.com](http://www.adalo.com)

**What can you make:** Native Mobile Apps and web Apps



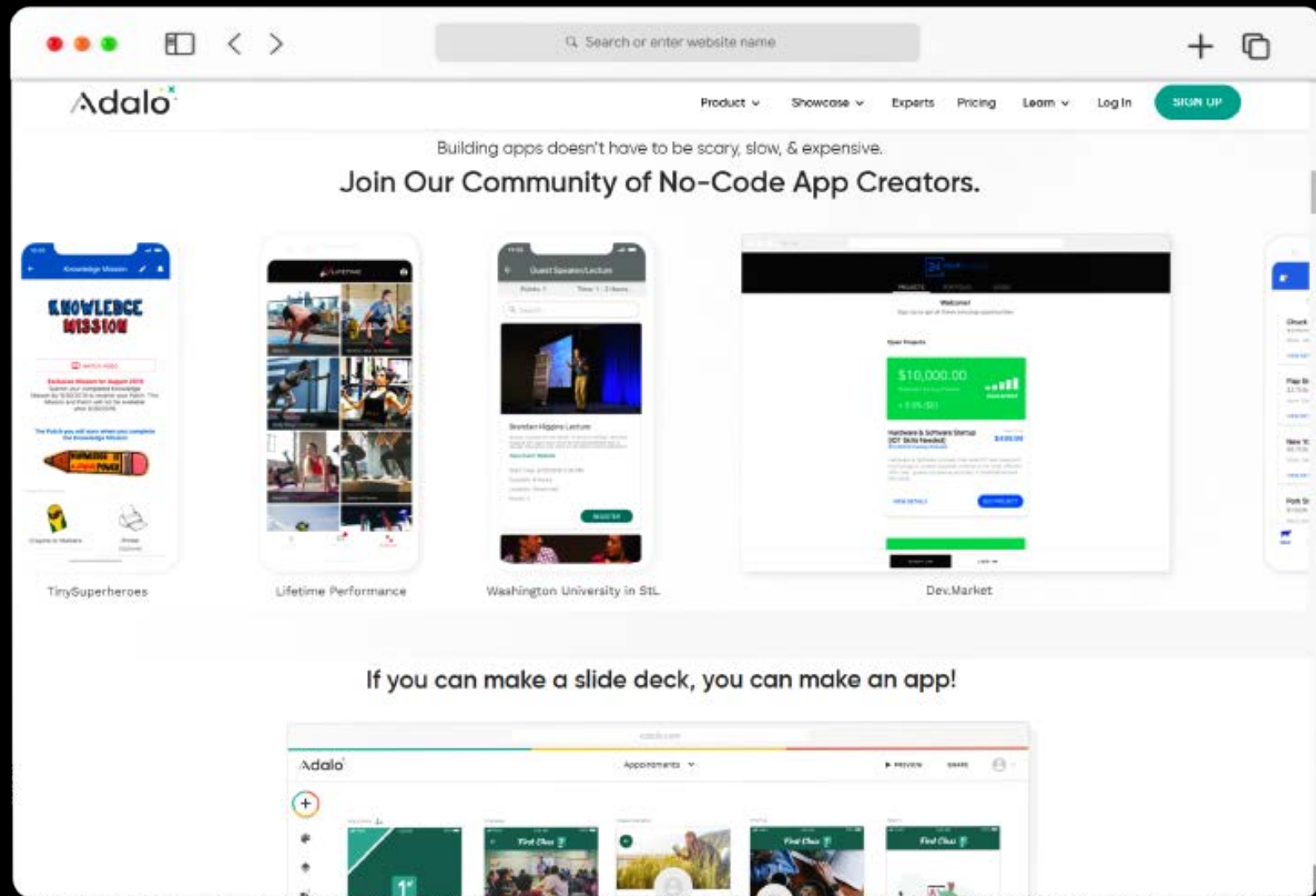
Adalo is a no-code platform with a drag and drop interface to create native mobile and web applications. This platform is ideal for launching your MVP and even iterating a couple of times until you have a solid product. However, it falls a bit short in terms of integrations and customization. So there will come the point where you want something more customizable.

Adalo has a relatively short learning curve. **The drag and drop interface is quite intuitive.** The databases are similar to Google Sheets or Excel, and thanks to the builder, everything is done very simply.

As a positive point, Adalo is very interested in the possibility of escalating the applications created within the platform. Furthermore, **they allow you to save different versions, which is advantageous since you can keep iterating your MVPs and improving.**

# Adalo

**Web:** [www.adalo.com](http://www.adalo.com)  
**What can you make:** Native Mobile Apps and web Apps



The next plan is the **professional plan or Pro Plan**, worth \$50 a month. It already allows you to publish your app in the Android and IOS stores, have your own domain, more storage space, connect to external databases, and access integrations that perform automatic actions.

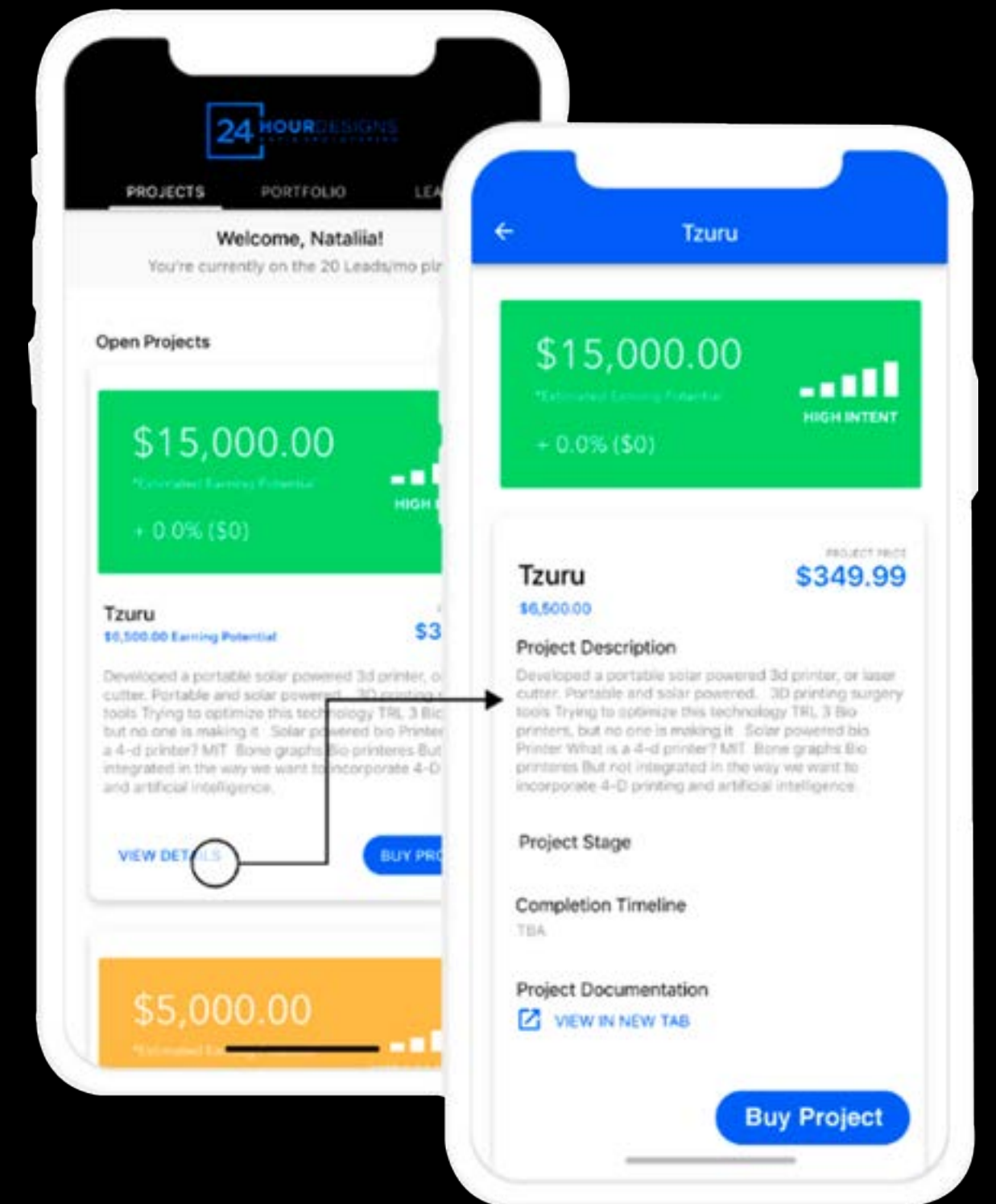
The platform also has a plan for companies or agencies called **Business Plan**, which costs \$200 per month. It has practically the same features as the pro plan; the only difference is an increased storage capacity of 20 GB and the possibility to have up to 5 app builders. **You also have a dedicated account manager.**

In terms of tutorials and community, they have **documentation and tutorials on their web and youtube channel**. Some of those were made by them and others by citizen developer communities.

There is a forum where you can ask for help, advice, and more from the community, but a lot of people complain about the lack of attention to the user and how slow they are to launch new features.

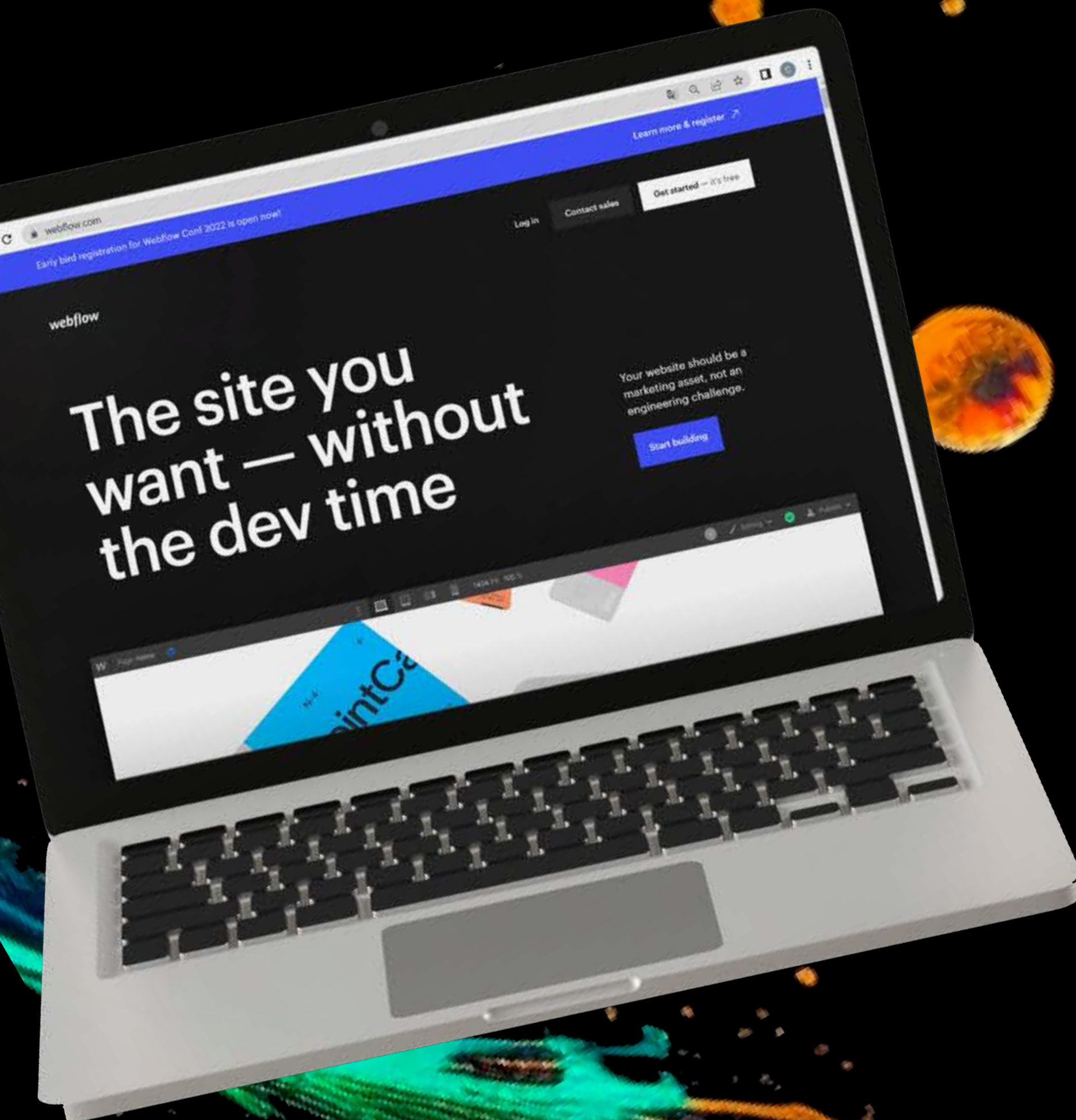
Adalo's prices are somewhat high compared to its competitors.

Although the **free account** has some good things, like the possibility of creating unlimited apps, you can only have 50 data lines per application. They also have all the branding of Adalo and their domain.



# Webflow

**Web:** webflow.com  
**What can you make:** Web pages



Webflow is a platform that has been growing since it came out in 2012 and has become the direct competition of WordPress. If we consider that 30% of the web pages in the world are made in WordPress, we are talking about the big leagues of web development without code.

It uses a visual interface; however, it's nothing like other builders like Squarespace or Wix. **Instead, Webflow pages are highly customizable and can be made from scratch by adding all the boxes of text and images and editing them yourself.** It's a platform that allows you the flexibility of front-end development without code, and all changes made are reflected on the screen and generate HTML, CSS, and JavaScript code that can be used and hosted outside the platform ecosystem.

**This platform is made especially for designers with experience in web page development.** This allows the complete edition and customization of the page the same way it would if code were used. But this also becomes its biggest weakness, as the learning curve can take too long for most people. In addition, it requires specialized knowledge in web design that takes time to learn.

The use of the platform is not as intuitive as other no-code development pages. Still, **all these difficulties are compensated by the ability of the platform to do practically everything you can think of natively.** It does not need plugins of any kind, and it allows you to do awesome things like animations within the platform.

**It is also compatible with mockups made in Figma, Sketch, and other design platforms. And for those who care about the code, the generated code is totally clean, which speeds up the site's loading.**

# Webflow

**Web:** webflow.com  
**What can you make:** Web pages

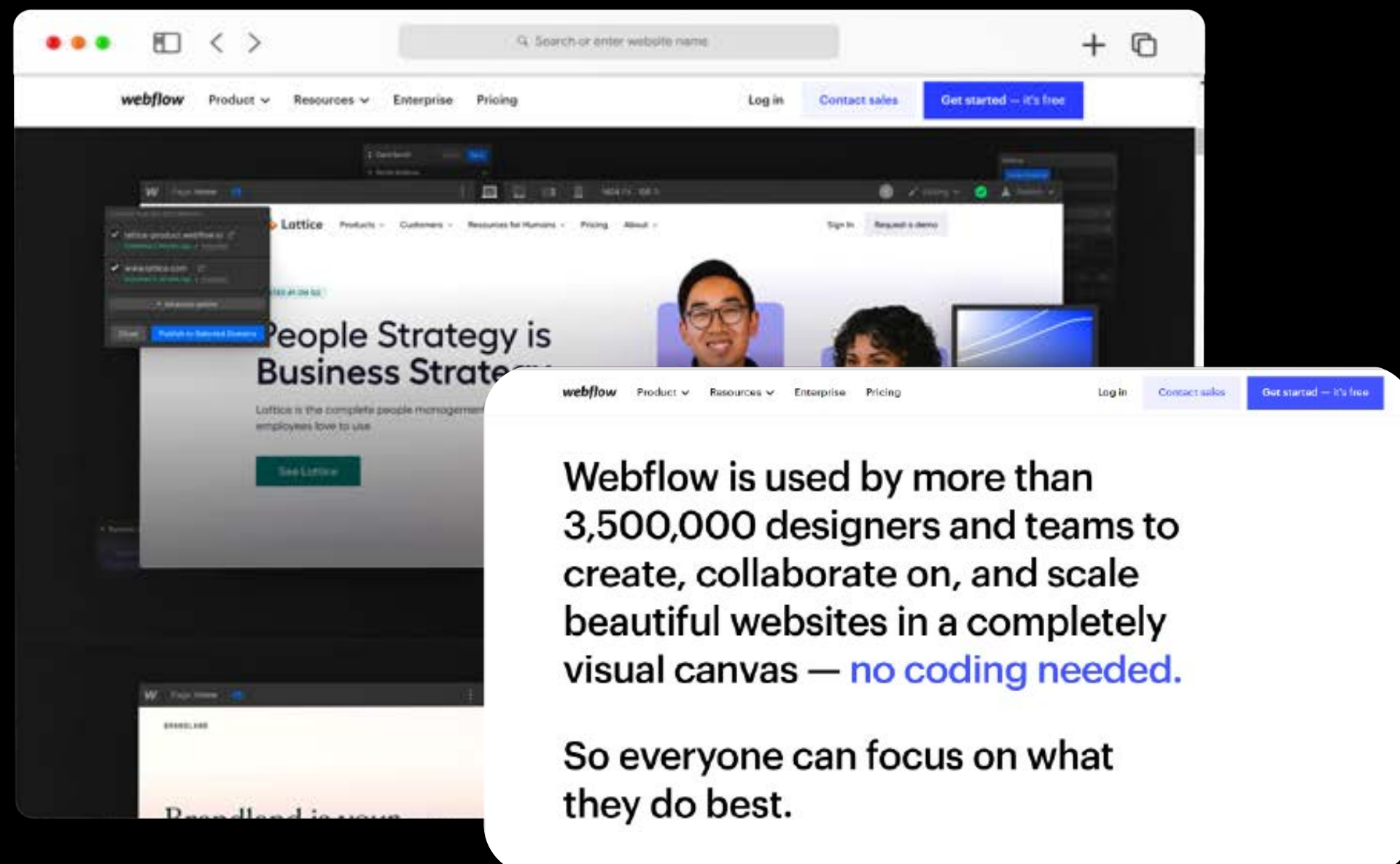
The scalability of the page is also excellent. It can be done from a very simple page to a totally professional one with animated transitions, payment gateways, etc. **The size of the and all the features may have depended a lot on the plan you have.**

They have plans that can range from 15 to 45 dollars per month. They also have a free plan, but this would be more to familiarize yourself with the platform and learn how to use it. These plans are more expensive than most platforms. **Still, they offer you a greater possibility of customizing your page, and you have access to all the platform's functionalities.**

As we already mentioned, Webflow is the direct competition of WordPress. Consequently, many web page designers are switching platforms, thanks to its advantages. Therefore, the community is growing rapidly, and it's possible to find tutorials and courses on how to use Webflow. Likewise, the platform offers a section called "University," where you can find tutorials and courses made by them so that you can learn and discover all the advantages of the platform.

In addition, on their website, Webflow gives you the possibility of hiring an expert if you cannot continue the development yourself. They also have a "wish list" of features that users consider necessary to develop and add to the list.

Finally, it must be said that the only difficulty that Webflow presents is the creation of large e-commerces. In this sense, platforms such as Shopify continue to lead. However, small boutiques and stores that do not have a wide variety of products can use Webflow without any difficulty.



In conclusion, Webflow is an excellent platform if you are looking for a very professional page with unique features and the ability to customize it 100% to your liking. Of course, you need to keep in mind the learning curve and that you will need time to master the platform. However, the results are very satisfactory in terms of users' experience and the look and feel of the final result.



However, Webflow is a perfect tool if you want a page that prioritizes the design, whether a website to sell services, a portfolio, or a collaborative blog-type page.

# Experts Opinion

**Diego Barrera**  
(Head of Design)

“

**Webflow** is the ideal platform for designers and people to work on their web without writing a line of code. The platform allows you to transfer your web project's entire graphic and interactive design vision without needing a template to start. Likewise, **Webflow generates clean and structured code without creating extra content.** This greatly helps the SEO of the page without the need for any external plug-ins.

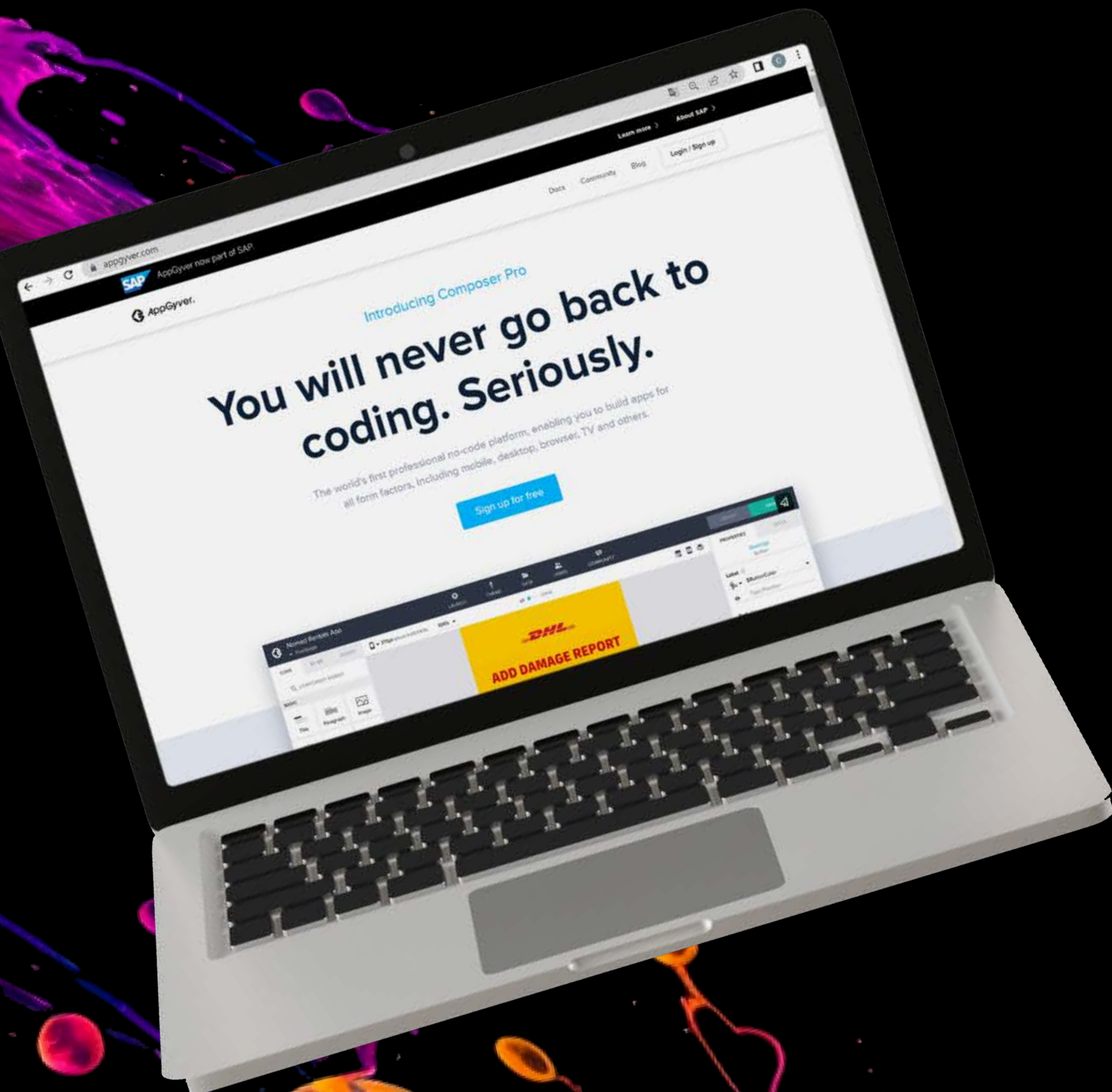
There can be a difficulty when it comes to having a page in **Webflow**, and it's the **lack of knowledge of marketing and SEO agencies of no-code tools and how to use them**, which is why they prefer much more traditional ones such as WordPress.

”

# AppGyver

**Web:** [www.appgyver.com](http://www.appgyver.com)

**What can you make:** Apps for mobile, desktop, browser, TV, and others



AppGyver is a European platform that promises a lot to its users. Its value proposition states that **"You will never go back to coding. Seriously."** Ensuring that you can do absolutely anything with the platform and its drag and drop interface.

Like other no-code platforms, AppGyver has a very attractive visual interface and easy to understand and use; even the interactions and workflow of your app can be designed visually, **which is very helpful for users without any experience.**

AppGyver can develop applications for all types of equipment, from native applications for Android and IOS cell phones to apps for smart TV and desktop. **In addition, they ensure that you can convert your application to any operating system, so you only need to develop your application once, which saves a lot of time.** Another advantage for users is the possibility offered by the platform to publish your applications in the google stores and the Appstore.

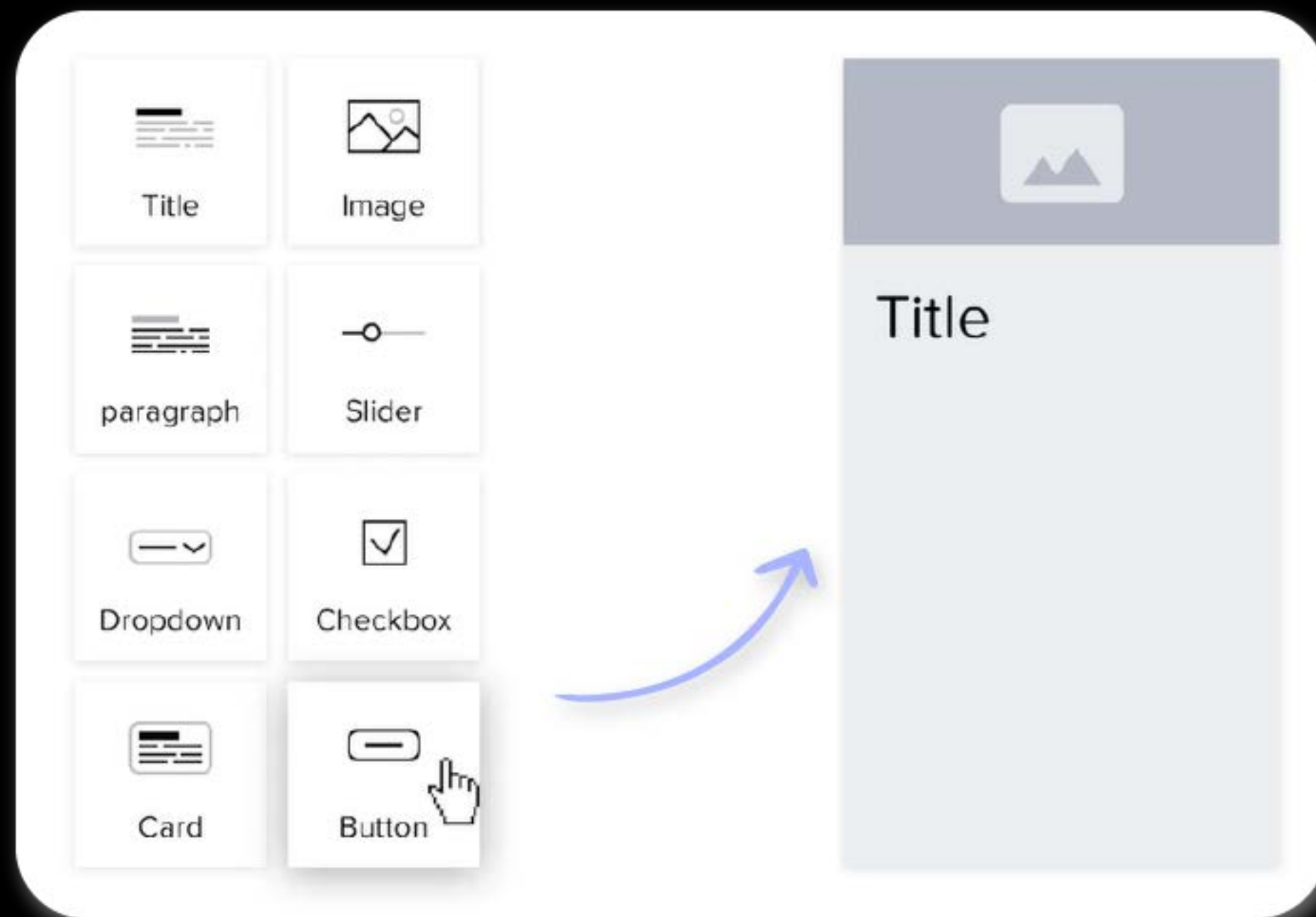
The platform also offers a vast library of layouts and allows unlimited integration with third-party software and APIs. If you have some development experience, it also enables you to make some customizations to both the back and front end using custom code.

Now, everything is fantastic, right? The question is, how much does this platform cost? That's the best part. AppGyver is **FREE** for companies with profits below 10 million dollars. You have access to the entire platform just by signing up.

# AppGyver

**Web:** [www.appgyver.com](http://www.appgyver.com)

**What can you make:** Apps for mobile, desktop, browser, TV, and others



It is interesting to mention that this platform also allows customization through code. **So if you have experience in low-code or are a professional developer, you will probably get even more out of everything that can be done with AppGyver.**

On the negative side, there are no templates. So you must start each of your projects from a blank canvas and do everything yourself. **So if you are a complete beginner, it can take some time to get familiar with everything on the platform.** Likewise, the customization capacity of the builder is limited. To achieve full customization of the front end, you must code it yourself. This, however, is quite normal in almost all no- and low-code editors.

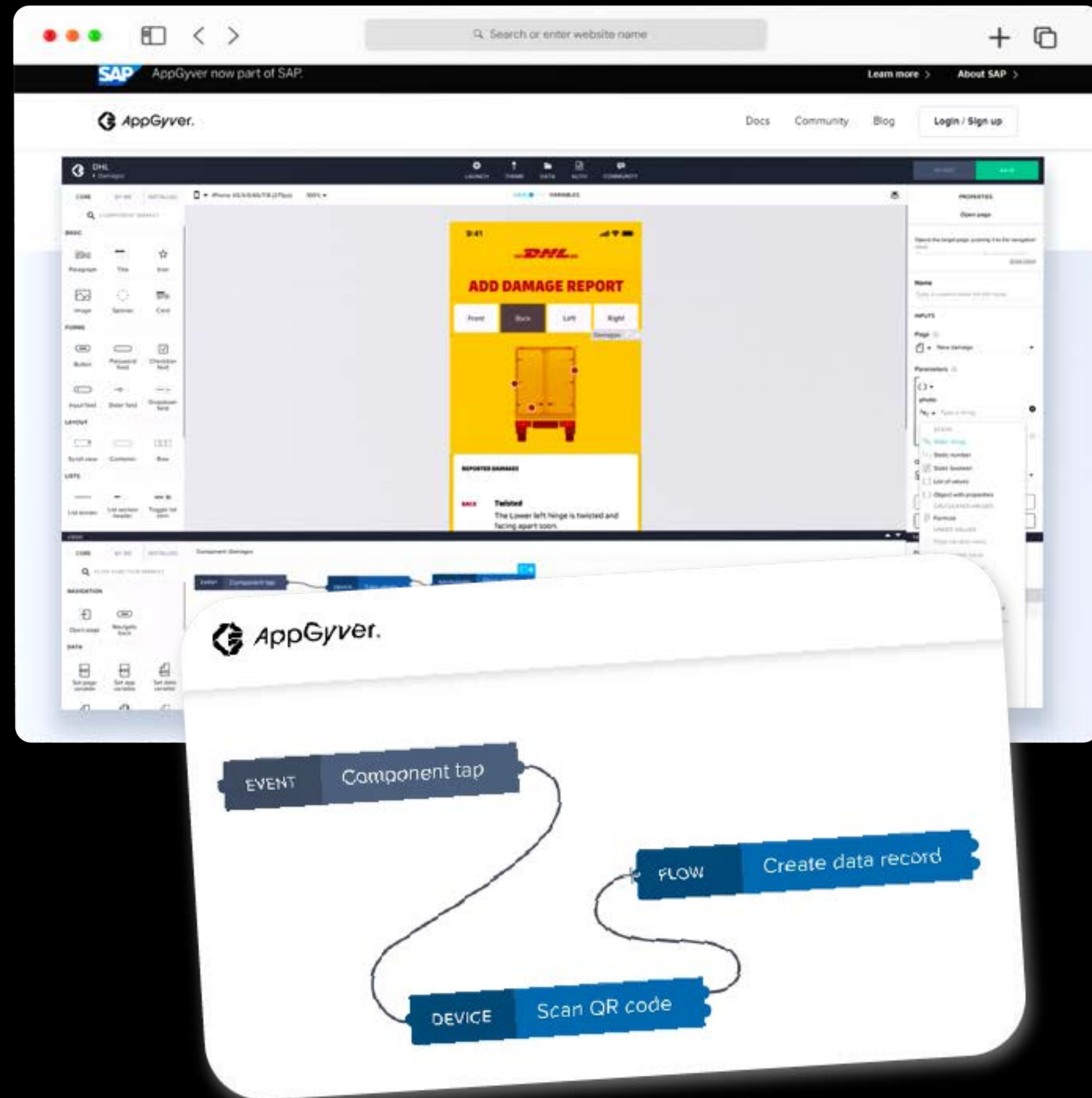
The learning curve of the platform is relative. Learning to master the editor and making an app with the plugins offered by the platform is quite simple. **You can even do it by playing around a bit without the need for a tutorial.** However, if you want to do something more complicated, do some integrations, or use third-party software, you may need a video, a course, or a tutorial. Still, those are available on its platform as official teaching material and documentation, and on youtube, you can find videos made by the community.



# AppGyver

**Web:** [www.appgyver.com](http://www.appgyver.com)

**What can you make:** Apps for mobile, desktop, browser, TV, and others



There is also another downside that may be a problem for some developments. AppGyver doesn't have cloud storage or a file manager. **You need an external provider to add those functionalities to your app; however, most of these services are paid.**

The scalability of the applications is incredible. **You can go from MVP to a fully functional app but just adding and changing things.** You will never need to pay to escalate your app within the platform. This rule does not apply to third-party software, where you need to pay their plans.

Another good thing about this platform is its **community**. Customer service is good, and the forum is active. Most users claim to have **received answers to their questions in less than 24 hours**, either from the staff or other users.

In conclusion, AppGyver is a powerful platform to make your MVP and scale it without code. The fact that you do not have to pay anything unless you have high revenues makes it super attractive for founders who don't have or don't want to invest a lot of money in something they are not sure will be successful. **It also allows you to add some custom code and doesn't limit you in how many integrations you can have.** In short, AppGyver is a platform to consider when thinking about the future of your startup.

# FlutterFlow

**Web:** flutterflow.io

**What can you make:** Native mobile apps



FlutterFlow is a no-code platform for designing and developing native Android and iOS applications. It's based on Flutter, Google's mobile software development kit, and works with Flutter's existing codebases, which are open source. Thanks to Flutter's constant updates every week, **the FlutterFlow team adds updates based on the new features created by the Flutter users and Dart programming language.**

Even though FlutterFlow is only about a year old, it has already managed to form an active community of users. Moreover, it seems to be growing thanks to the constant release of new features for the platform.

Like Web Flow, **they have a strong interest in designing and creating an application with an excellent look and feel that offers an incredible UX and the possibility of using custom code.**

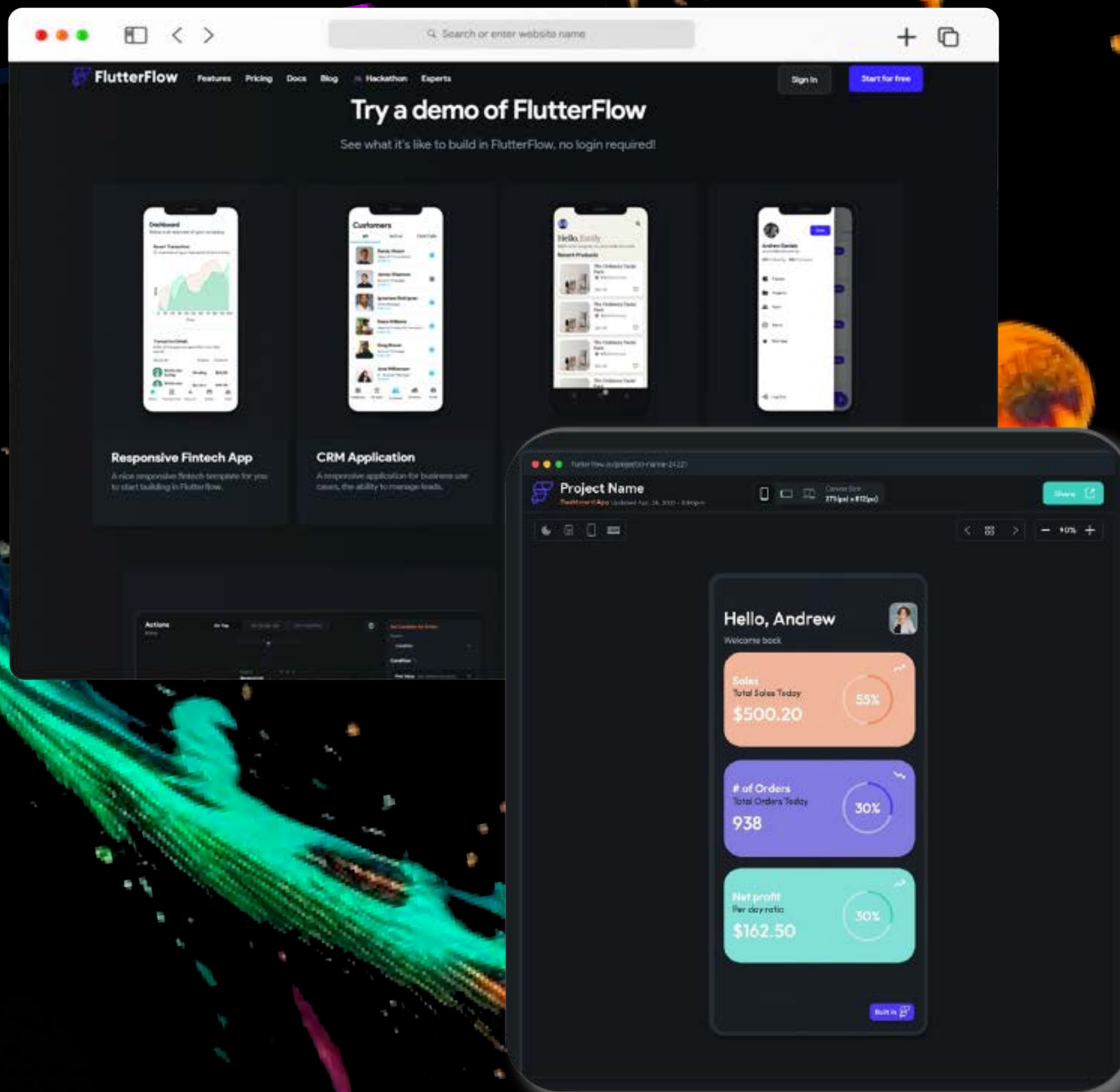
Before going into the details of the platform, I would like to highlight two very positive aspects of FlutterFlow. **The first is the possibility to export your Dart code and use it outside of the platform ecosystem.** A great advantage if you want to scale your platform or work with a team of professional developers. On the other hand, **they allow collaborative work even on its free plan. You can work with a team on the same project and the changes made don't affect the others' work.**

In general, FlutterFlow is very similar to any other no-code platform. **It has a drag and drop interface, templates, integration with third-party software, and the possibility of developing an app from scratch using a Blanc.** However, the customization possibilities set them apart from the other platforms.

# FlutterFlow

**Web:** flutterflow.io

**What can you make:** Native mobile apps



Unlike other highly intuitive platforms, FlutterFlow is not that simple to use. They put a lot of effort into making it appealing for people with design experience. Therefore, designers with some experience can more easily adapt to the wide offer of widgets and platform components without forgetting the functionality.

**This is not the simplest platform on the market for non-coders. It has a long learning curve, and it's necessary to use the tutorials and documentation to understand and use the app's full potential.** It's good that they have a pretty active community creating content. They are releasing tutorials and courses to learn how to use the platform. It is also recommended to have some experience with similar platforms or application designs.

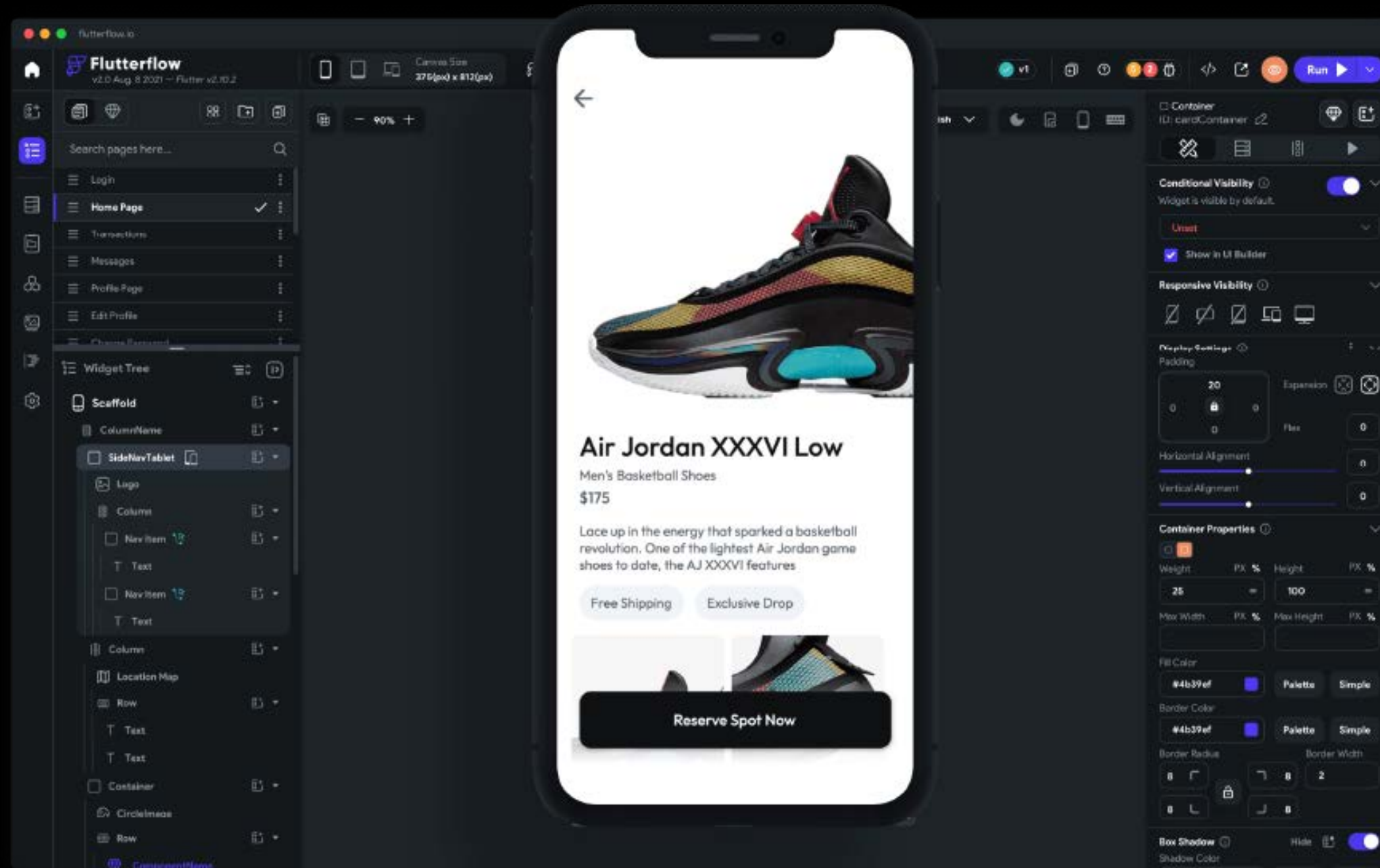
**The prices of the platform plans are also quite favorable if we compare them with their competition.** In addition, if you are not located in the USA or Europe, you can have a regional discount on the different plans.

**The free plan** gives you access to the entire platform, third-party software integrations, customized features and widgets, and the possibility to work simultaneously with a team, among other things. **Compared to other no-code platforms, the free plan allows you a wide range of action.**

# FlutterFlow

Web: flutterflow.io

What can you make: Native mobile apps

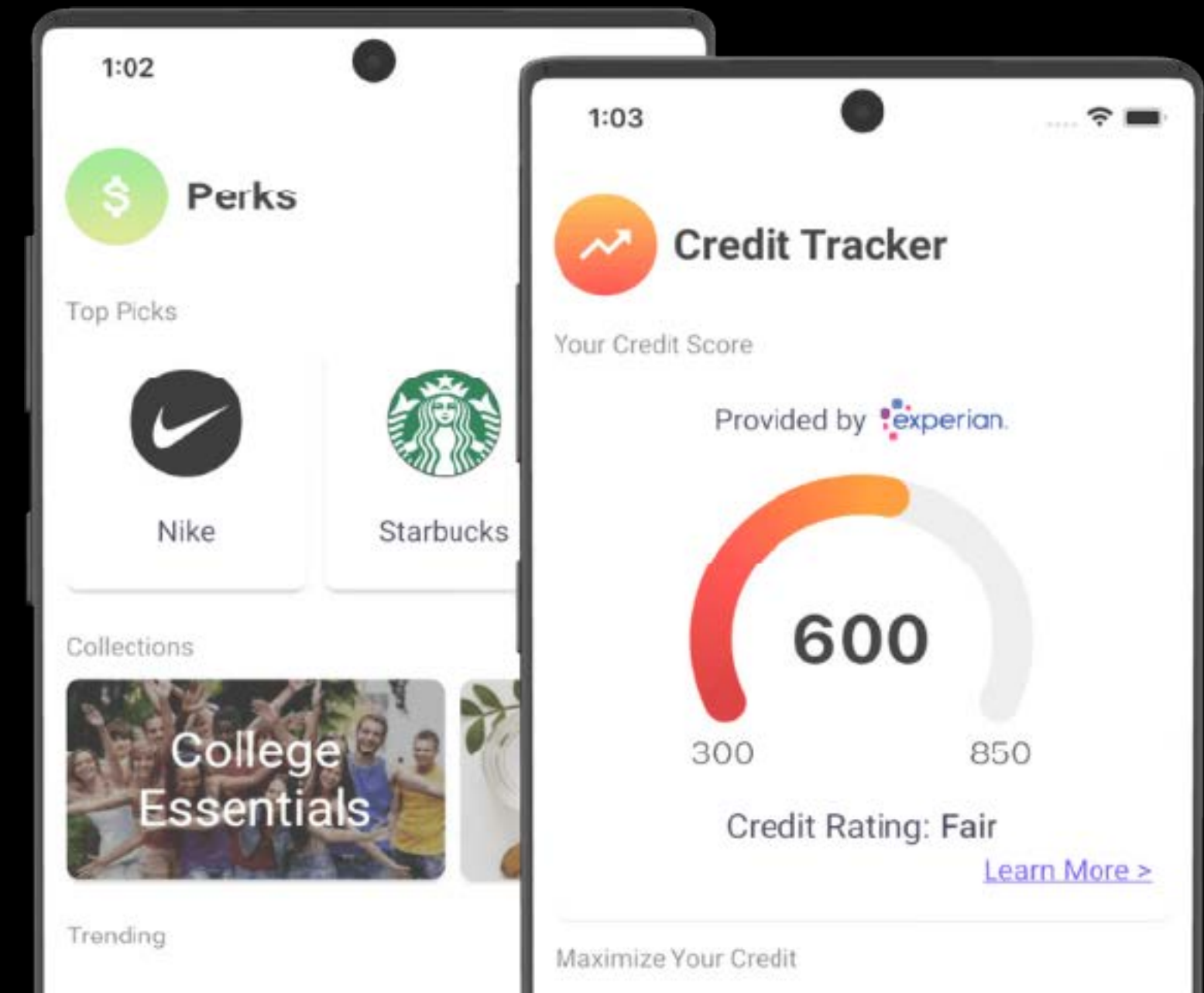


Finally, as I already mentioned, FlutterFlow is a relatively new tool launched in May 2021, but that has managed to get a big community derived from its direct connection with Flutter and Google.

This community is active, and although they do not have a forum, it is possible to find many sources of information and tutorials on the internet. On the other hand, **they offer the services of experts that can help you build your app if you are stuck, or you can contact them, and they answer all your questions.**

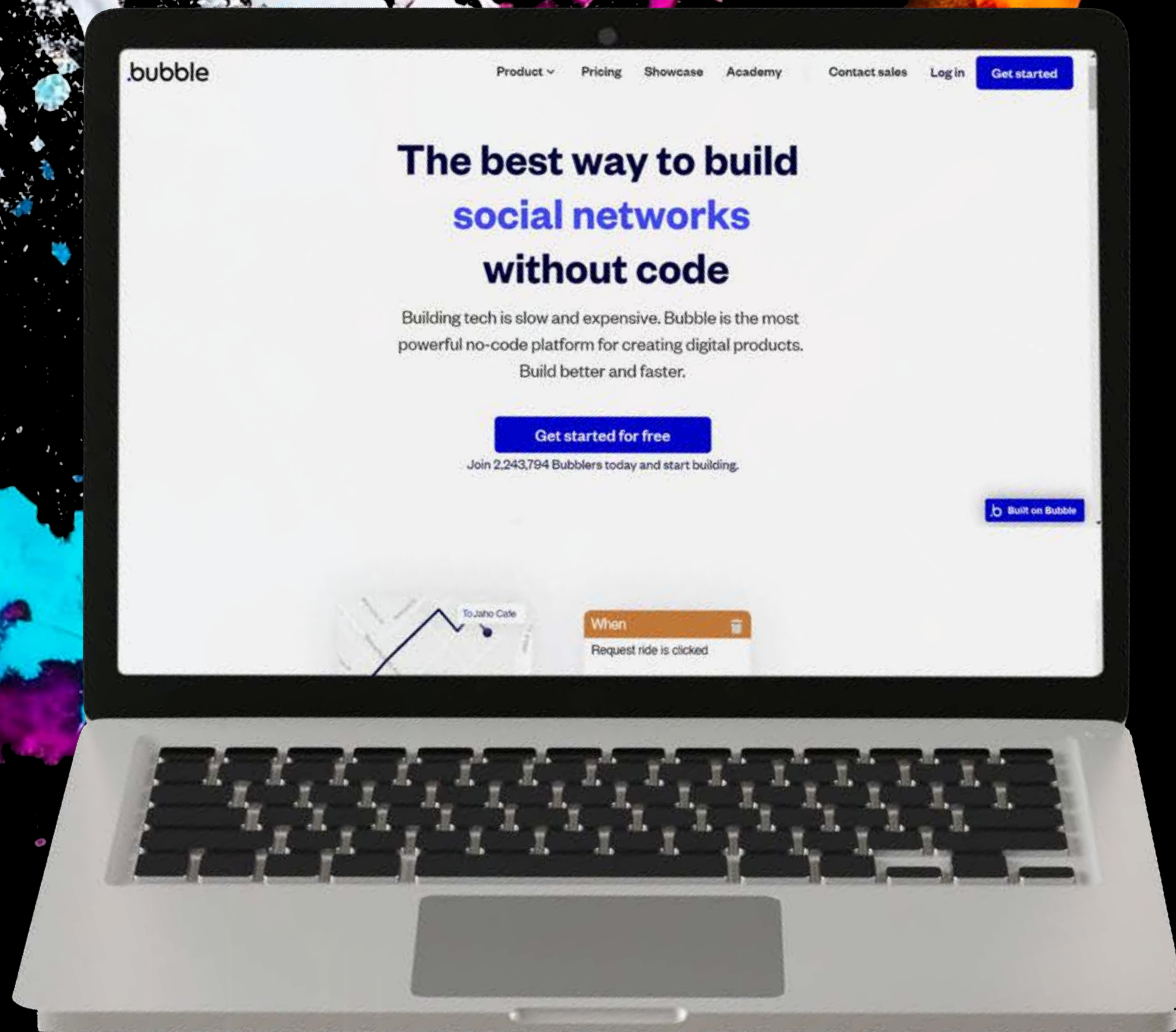
**The second plan costs 30 dollars per month.** It has everything on the free version and allows you to generate the APK of your apps and download the source code. To use it outside the platform ecosystem.

The third plan costs 70 dollars and allows you to deploy your app to the App Store and Play Store, integrate with Custom APIs and Github, and have a Firebase Content Manager. In addition, **the platform offers a 14-day free trial of any of the two payment plans, something that other platforms on the market don't have.**



# Bubble

Web: bubble.io  
What can you make: Apps for desktop and browser, no native apps



The last platform we will look at is bubble.io. Many of you have likely heard it before since it is a trendy platform among startup founders.

Bubble has been around for 10 years or so, but only a few years ago, it became widely known as one of the best no-code platforms for startup founders. **All thanks to its huge number of plug-ins, its API integrations, and the possibility for users to create their own plug-ins if they do not find the right one.**

The first thing about Bubble is its robustness compared to other web application platforms and non-native mobile apps. **It currently offers the most plug-ins, and applications can even be created with cutting-edge technologies such as Ai, machine learning, and blockchain.**

Let's start by talking about the learning curve. Although Bubble is tremendously popular, it is not the simplest application on the market. On the contrary, a user with no experience with no-code platforms will need a lot of time to get used to it and understand how the platform works. Therefore, before starting to work on the platform, **it is recommended to watch tutorials, read the documentation on the page, or watch some videos from the community.**

**This can help users get a hand on the platform much sooner.**

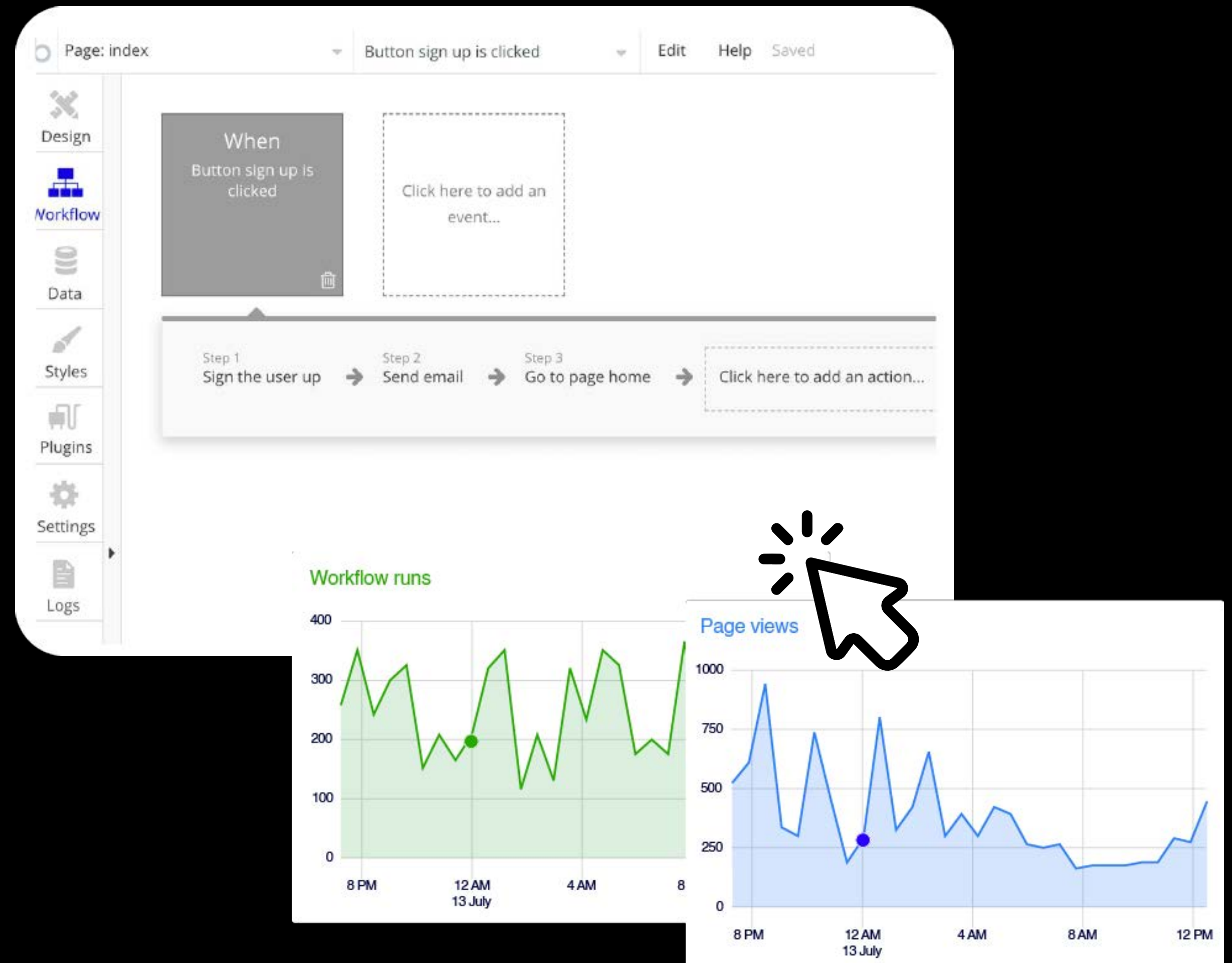
# Bubble

**Web:** bubble.io  
**What can you make:** Apps for desktop and browser, no native apps

Since we mentioned the community, let's talk about that. Bubble currently has one of the largest and most active communities in the market; hence you can learn a lot from the content they produce and post both on youtube and on the forum. **This is an advantage over other platforms that don't have as many people actively using the platform, creating content, and even creating plug-ins.**

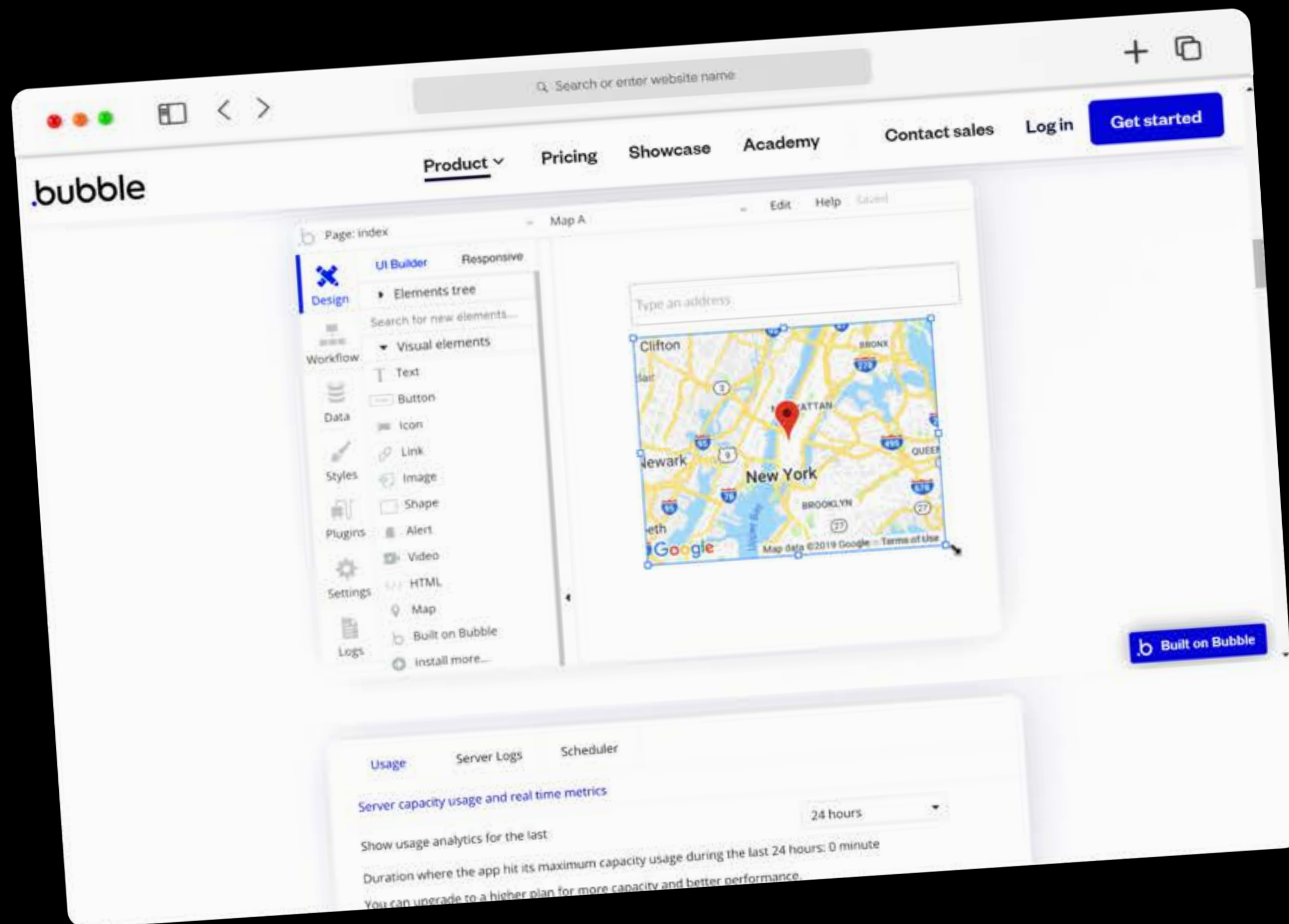
Another significant aspect of Bubble is **the ability to scale and customize your applications without using code.** Bubble is 100% no code and allows pixel-perfect editing. This makes it highly customizable but also very time-consuming if you don't have design experience.

The scalability is also incredible. Without leaving the platform, **you can create anything, from a prototype to a very robust and fully functional and robust app.** **This is one of the reasons why many founders use it.**



# Bubble

Web: bubble.io  
What can you make: Apps for desktop and browser, no native apps



Probably the negative point of Bubble is in its prices. Currently, Bubble has 4 plans. **The first one is free** and allows you to access the platform and create your application, but it does not allow you to launch it or integrate it with APIs. It also has minimal storage capacity and cannot save versions. **To launch your app to the market, you need to pay a personal plan of 29 dollars per month.** This plan already allows you MUCH more storage, integration of APIs, and attention to the platform in customer service.

So far, so good. The main problem is that, unlike other no-code platforms, **you pay per app and not per account.** **If you want to launch a second app, you have to pay for another personal plan.** Likewise, there is also no possibility of working as a team if it is not up to the professional and production plans already for companies and costs 129 dollars and 529 dollars per month, respectively. The advantage of these plans over the personal plan is increased storage, the ability **to save backups, save versions, and have more than one editor on the platform.**



As a result, Bubble really stands out whether you are just starting out with a new project or want to add more features to what you already have built.

# Experts Opinion

**Ignacio Piris**  
(Head of Development)

“

**Bubble's** intuitive platform, great UX, and constant updates make it one of the best, if not the best no-code platform I've worked on.

**Small details such as more features in their data storage and UI elements that make the development process really easy to handle, turn the no-code experience into something that anyone can pick up at any time and feel real progress with.**

**The number of plugins developed for Bubble is enormous, but its strength lies in the number of those developed for web 3 applications. This differentiates Bubble from other no- and low-code platforms that only develop for web 2.**

”



# Conclusion

**When choosing a platform to create an MVP, several important elements are to consider.** Not only does the speed to release your product matter, but also how easy it is to develop it and how fast you can iterate. Finally, you need to **analyze how much the platform will allow you to scale your project** until you need to change to a bigger platform or develop it from scratch.

**The platform you choose must be easy for you to use and offers you the basic functionalities you need for your minimum viable product beyond its first version.** It is not worth choosing a platform that is too simple if, for the second version of your MVP, you have to change it to another.

Finally, keep in mind that these are not the only platforms on the market. **We have chosen these because they are the most used by startups to create their MVPs and prototypes.** Still, if you feel that you need another one or have some coding knowledge, you can also check on low-code platforms. Generally, **these platforms have a higher learning curve than a no-code**, but many things can be done only with their drag and drop interfaces; they also allow you to customize many features through programming languages.



**nocoded**