



STAGE 0. PROTOTYPE AND TECHNICAL SPECIFICATION

Before starting development, it is important to clearly understand why you need an application or website and what tasks it should solve:

- 1 Define the full functionality.
- Collect examples of websites or applications where you are satisfied with how the functionality works (important: this refers specifically to functionality, not design).
- Discuss the vision of the final product within your team and form a shared understanding.

The more specifics and examples you provide, the easier it will be for developers to execute and understand how you envision the result.

We will explain in detail how to prepare a technical specification (TS) in Stage 5



STAGE 1. DESIGN: TEMPLATE OR CUSTOM DEVELOPMENT

At this stage, you need to decide whether to use a ready-made design template or work with a designer on a unique project.

KEY POINTS:

A frontend developer ≠ a designer

Their task is layout, functionality implementation, and connection to the backend (which is provided by us and does not need to be developed). Design (appearance) and UX, however, are the work of a designer.

Design changes = additional resources

Any adjustment to the design requires thinking through adaptability and entails modifications in layout and functionality. Therefore, what you may consider "minor edits" often lead to poor results without a designer's involvement, and you will still have to pay for them.

! IMPORTANT NOTE:

A frontend developer may promise to make design changes, but the result is rarely satisfactory.



1.1 Using a Template

A template is a fast and inexpensive way to develop the visual part of a website or application. There are many catalogs and resources where you can purchase ready-made solutions for any business needs.

Things to keep in mind with this approach:

- Making major changes to a template is difficult and costly. Blocks cannot be freely rearranged, removed, or replaced without deep layout rework.
- Since there is usually no designer involved, modifications are handled by the frontend developer. They implement them as they see fit, which leads to additional time and cost, and the final result is often worse than expected.
- If you are fully satisfied with the structure of the template and only need to change colors, banners, and add branding elements, it is best to use it without modifications.

IMPORTANT NOTE:

A template is not a constructor. Its elements cannot simply be rearranged, replaced, or removed.



1.2 Working with a Designer

If you decide to create a unique design for your website or application, at this stage you do not need a frontend developer at all. The key role here belongs to the designer, and it is especially important to pay attention to their UX expertise — the user's interaction experience with the application. The interface should not only be visually appealing but also simple, logical, and convenient to use.

Very often, businesses at this stage strive for an "ideal" result, investing huge resources into it. In practice, however, unnecessary complications often go unnoticed or unused by customers.

Instead of overspending on things that bring no real value, it is much wiser to gradually improve the product during its actual use.

Ask yourself and your team honest questions: Is this solution truly necessary? Can I work comfortably without it? This approach helps avoid unnecessary costs and focus on what really matters for the end user.

To ensure the project's success, keep two simple rules in mind. First, the product itself matters: its price, quality, and value for the client. Second, the application should be light, convenient, and easy to understand, without unnecessary complexity.

And one more important point: if the majority of your audience will use the website or application from mobile devices, start designing with the mobile version. This will help avoid many revisions and ensure the best possible user experience



STAGE 2. LAYOUT AND FINDING A FRONTEND DEVELOPER

When the design is ready, it is time to bring it to life as a working website or application. For this, you need a frontend developer. Today there are many resources where you can find talented and responsible specialists: upwork.com, freelancer.com, toptal.com, guru.com, fiverr.com.

When choosing a contractor, pay attention not only to the cost but also to experience, reviews, and real examples of completed projects.

To reduce risks, you can use intermediary platforms. They act as a guarantor between the client and the developer: funds are reserved and transferred to the developer only after the task is completed. A commission is charged for this, but you gain the assurance that the work will be delivered. Examples of such services include **toptal.com**, **lemon.io**, **arc.dev**, **upstackhq.com**, **topcoder.com**, **upwork.com**.



IT IS IMPORTANT TO UNDERSTAND:

Many studios on the market do not keep their own programmers on staff. When they receive an order, they hire the same freelancers and then bill the business five to ten times more. Essentially, **you overpay for middlemen.**

Therefore, unless you have a specific reason to work with a studio, direct collaboration with a developer through trusted platforms often proves more reliable and cost-effective.



STAGE 3. HOW TO CHOOSE THE RIGHT DEVELOPER

Once you have found a candidate, it is important to determine whether their experience matches the tasks of your project. Much depends on the scale and complexity of your website or application.

Simple application or website with minimal logic.

For example, a landing page or a service with basic functionality. In this case, you can work with a junior developer with at least one year of experience. Such specialists charge lower rates and value reviews highly, but deadlines may stretch since they require more time to complete tasks.

Medium-complexity project.

If there are many features, it is better to look for a mid-level developer with at least three years of experience. These specialists already work independently, and they complete tasks predictably and on time. Here, it is especially important to carefully review their portfolio and feedback.



Complex application or website with specific business logic.

In this case, you will need a full-stack developer (frontend + backend) or an additional backend specialist to support the frontend developer with certain tasks. This is relevant when it comes to implementing complex or specific functionality, such as currency conversion in a large catalog or complex integrations with external systems.



IMPORTANT NOTE:

«The developer's level must match the level of the tasks.

The more precisely you define requirements at the start, the lower the risks of overpayments and delays».



STAGE 4. CHOOSING THE RIGHT TECHNOLOGY STACK

Our platform is **compatible with any modern frameworks and technology stacks.** At the same time, we recommend relying on **popular and widely used solutions in the market.**

For web applications, the minimum set looks like this:



Next.js — a modern framework that allows flexible SEO optimization and simplifies the development of complex interfaces.



Tailwind CSS — one of the most popular styling tools, which speeds up work and ensures a consistent design style.



ESLint + Prettier — tools for maintaining code quality and formatting consistency.



For mobile development, there are two main paths:



React Native — optimal for less resource-intensive applications, works for both iOS and Android.



**SwiftUI + Kotlin — an option for more resource-demanding applications where native development is required.

If you choose **React Native**, don't forget the additional set: **Tailwind + ESLint + Prettier.** This ensures clean and maintainable code.



IMPORTANT NOTE:

«The recommendations are based not only on technical capabilities but also on the popularity of tools in the market. The less a technology is used, the fewer specialists know it and the higher their rates.

By using widely adopted solutions, you reduce risks and can easily replace a developer if necessary».





STAGE 5. TECHNICAL SPECIFICATION AND AGREEMENTS WITH THE DEVELOPER

Once you have chosen a developer, it is time to formalize the agreements.

The foundation of any working relationship is a properly drafted technical specification (TS) and a contract that clearly defines all key parameters of the project.

What is important to consider when drafting the TS and contract:

- Avoid vague wording.
- Clearly describe the functionality, even if it seems obvious.
- Fix the timelines: start date, completion date, and intermediate milestones.
- Specify the technology stack that will be used and include it in the contract.
- Work only through a contract, a secure transaction, or escrow services.
- Do not demand work beyond the TS without additional payment, as this preserves good relations and avoids conflicts.
- If delays occur due to the developer's fault, demand compensation in the form of additional work at their expense or a discount.

These steps will help avoid misunderstandings and conflicts and protect you from situations where the project drags on or its cost suddenly increases.



Documentation

It is very important to agree in advance: **before final payment, the developer must provide documentation** on the infrastructure and project structure, a list of the tools used, and comments in the code. This applies to both freelancers and studios.

These documents are your guarantee that the **frontend part of the application** can be maintained and further developed even by another specialist. The backend always remains under your protection within our platform, so there are no risks there. That is why it is especially important to obtain full documentation specifically for the frontend part and comments in the code: this will safeguard you and prevent unnecessary expenses when replacing a developer.

Caution with file exchange

Do not forget about security. Before the deal is concluded, be extremely careful with any files and links from contractors. Instead of a TS or documentation, you may receive a malicious file that compromises accounts or access to banking services. This is especially risky if you are being rushed or offered conditions that seem too good to be true.

To protect yourself, use licensed antivirus software, check all files before opening, and, whenever possible, work via Google Docs.



IMPORTANT NOTE:

«When registering a project on the OneEntry platform, all key data, such as phone number and email address, must initially be registered under the business owner. If this is not possible immediately, make sure it is specified in the contract and TS that all access credentials to the project's control panel (login and password) will be transferred to you before final payment, along with updating the phone number and email address».

In the event of a developer change or a team member leaving, it is necessary to promptly remove them from the list of administrators or change the access passwords. This procedure is set forth in the OneEntry offer agreement.

If unauthorized access to your application is detected, our company has the right to request original supporting documents: the domain ownership certificate, the company's founding and registration documents, an order appointing the CEO or another authorized employee, as well as confirmation from the telecom operator verifying that the phone number is indeed assigned to the authorized person.



STAGE 6. PROJECT DELIVERY AND STAFF TRAINING

The final stage is not just the handover of a finished application, but a full acceptance of the project. To ensure this process goes smoothly, include in the contract a **defined number of training hours for your employees.**Our platform is simple to use, and even staff without technical skills will be able to manage the application. However, every business has its own specifics, so it is better if the developer does not limit themselves to handing over the project, but also clearly demonstrates how everything works.

What is important to consider during project delivery:

- Obtain all documentation and comments in the code before final settlements.
- Agree on possible support hours for the frontend part of the application and clarify the cost of such work in advance. Keep in mind that support provided by an individual developer is significantly cheaper than the rates of studios.





Remember that studios often earn more from ongoing support than from the initial application development. Dishonest companies may deliberately include «hidden problems» so that the business regularly faces difficulties and pays for their resolution.

By working with our platform, you are protected from such issues: **the backend belongs only to you,** and no one can interfere with its operation or plant «time bombs» for the future. **The application will run for years without interruptions** unless you decide to improve or change something in the functionality or layout.

AND MOST IMPORTANTLY, ONCE AGAIN:

Register your OneEntry account only to your own phone number and email address.

This guarantees that you will retain full control over the project.

Follow the recommendations from Stage 5!



STAGE 7. UPDATES AND INTERFACE DEVELOPMENT

Business does not stand still, and in 3–5 years you will most likely want to update the look of your application or website or change certain interface elements. Here it is important to remember: **the business logic and backend remain the same and do not need to be rewritten.**

This can be compared to a house: the foundation and walls are already yours, but you can repaint them, replace the windows, or update the floor without destroying the entire building. The same applies to an application — the external appearance can change (the frontend part), while data storage and processing continue to work as before.

Why is this so important?

Because many developers use «rigid» application-building tools and try to convince businesses that even small interface changes will require rewriting part of the backend. In essence, this is presented as creating a new application, with costs comparable to initial development.

In our platform, such risks do not exist: the logic and infrastructure remain unchanged, and interface updates do not become an excuse for inflated budgets.



Frequently Asked Questions and Answers

Question 1. «Why do you need OneEntry? My stack is better».

Answer: The stack chosen by a developer may indeed be good, but it is important to consider not only its quality but also its prevalence in the market. If a technology is rare, there are fewer specialists who know it, their rates are higher, and replacing a developer in the future will be much more difficult.

We recommend using **the most popular and widely adopted frameworks.** Our platform is fully compatible with any stack and works without conflicts, so there are no limitations for developers.

These recommendations are given solely from the perspective of business convenience: so that, if necessary, you can quickly and easily find a new specialist. For example, technologies like Angular can be more expensive to maintain, since there are fewer developers available in the market and their hourly rates are higher.



Question 2. «There are free CMS solutions and we have our own or, for example, we can use open-source. Why pay for this at all?»

Answer: At first glance, free CMS platforms may seem like an attractive option. But it is important to understand: OneEntry is not a CMS, but a fundamentally new technology that combines cloud infrastructure, ready-to-use business logic (backend), and a content and data management system. That is why comparing it to ordinary CMS platforms is incorrect.

However, to see the real picture, you should take into account several nuances:

1. A server is still required.

Regardless of which CMS you use, you need a server to store application data. And this is never free.

2. Server and CMS maintenance costs money.

Setup, updates, backups, and storage of backup copies are all separate tasks that must be paid for. With us, all of this is already included in the price.



3. Dependence on third-party plugins.

Most of the functionality of free CMS platforms is based on additional extensions. In case of malfunctions, you are the one who pays for fixes and compatibility of these plugins, even if the CMS itself is «free».

4. Dependence on backend code and developers.

Every programmer writes code in their own way and rarely thinks about who will maintain the project in the future. A new developer almost always criticizes the code of the previous one and either demands to rewrite everything from scratch or charges a high fee for analysis. This creates a trap of dependency for the business: problems appear not immediately, but when the application already generates revenue, and the entrepreneur faces the pressure of «hidden costs».



SUMMARY:

«OneEntry provides a single solution without third-party plugins, with built-in infrastructure and support. This frees businesses from unpredictable expenses and dependency on specific developers or agencies».



Question 3. «I haven't worked with OneEntry and I'm ready to take on the project only if you pay me for the time to study the platform».

Answer: Our platform has detailed documentation and training videos. To understand the principles of working with OneEntry, a developer needs just a few working hours. Later, while working with the platform, they will have access to technical support where they can clarify any questions that arise.

In our experience, Middle-level developers almost never contact support and a basic introduction is enough for them to work confidently.

No one ever gets paid for getting acquainted with a system. **First**, you cannot verify whether the contractor actually spent time studying it or simply sent you an invoice. **Second**, a good specialist quickly adapts to a new platform and does not demand extra money. Constant counterarguments and attempts to impose payment for "studying" are a red flag regarding the developer's competence.

FROM OUR EXPERIENCE:

«A qualified specialist never argues and quickly gets into the work, while those who abuse such demands usually count on the business's lack of technical knowledge».



Question 4. «I don't trust cloud services.

Tomorrow they might disappear and all the data will be lost».

Answer: This is a common concern, but in practice it is unfounded. Here's why:

1. Data storage is not limited to the cloud only.

You can synchronize it with accounting systems and CRMs, so the information is duplicated and does not depend solely on the cloud.

2. Reliable backups.

We regularly create backup copies that guarantee the safety of your data.

3. Platform reliability.

We have invested significant resources in OneEntry and did not create it for short-term profit. The platform continues to evolve because many businesses solve their tasks on it, and that is the foundation of its stability.



IT IS IMPORTANT TO UNDERSTAND:

«The mission of OneEntry is fundamentally different from the approach of studios and developers.

They often work by the 'done and gone' model, stop answering calls, or demand extra money for fixing constantly arising issues that they themselves deliberately built in at the start. This is why, in order to keep control, such contractors often try to 'scare' businesses with distrust toward cloud solutions.

We, on the other hand, build **long-term and transparent relationships**, where data security and reliability are the foundation of our product».



Question 5. «I use another cloud backend, and my clients are satisfied».

Answer: It is important to understand that there is no ready-made solution on the market that combines backend, business logic, content management system, and cloud in one product without third-party plugins. OneEntry is unique precisely because of this: you get everything at once within a single platform.

Other cloud backends have a number of limitations:

1. Lack of a unified product.

Usually, you have to combine several services, connect third-party modules and plugins, which complicates maintenance and makes the system less stable.

2. High cost.

Most solutions are more expensive. Moreover, their pricing policy is non-transparent: you never know in advance how much exactly you will have to pay at the end of the month.



3. Multiple services lead to higher expenses.

In reality, it is often not one service but several that are used. Each of them requires separate payment and separate maintenance, which increases the total cost of ownership.



OneEntry offers a single tool where all key functions are already integrated. This reduces costs, simplifies maintenance, and frees the business from dependency on numerous third-party services.



Question 6. «OneEntry is a complex and unclear platform. I wouldn't recommend it to you».

Answer: Such statements often reflect not the platform itself but the level of the specialist making them.

OneEntry includes many features designed to make developers' work more comfortable, namely:

- An SDK that speeds up the development process.
- Detailed documentation and video tutorials with step-by-step explanations for every section of the platform.
- **Technical support** ready to answer any questions that may arise during work.

All of this is more than enough **for any qualified specialist** to master the platform and use it effectively in projects.



Question 7. «I reviewed OneEntry and it doesn't have the functionality you need. This platform is not suitable for you».

Answer: In fact, you will not find the same level of flexibility in any other solution as in OneEntry. Even if you need to implement specific business logic, this can easily be done through **the integration module** or by creating a custom module within the system.

The platform allows you to build unique scenarios while managing the entire application from a single point.



Such claims usually arise either from a specialist's unwillingness to explore the platform or from an attempt to lock the business into their own technology stack and maintain dependency on them.



Question 8. «I haven't heard of this platform, and there's little information about it online».

Answer: OneEntry is successfully used in different countries and has already proven itself over time and through real projects. Reviews and examples can be found on independent resources such as YouTube and G2.

It is important to understand that **OneEntry is a fundamentally new solution**, which is why it incorporates the most modern technologies. Older and long-standing platforms are often limited by their architecture: they were created in a different development paradigm and cannot offer the same level of flexibility and speed.



A telling fact: while development on other solutions may take three weeks, with OneEntry an equivalent project can be completed in just two days. This demonstrates **significant** optimization of development costs. At the same time, businesses gain a high level of protection, independence, and security built into the platform.



Question 9. «My solution is better for SEO».

Answer: OneEntry takes into account all the key aspects of SEO optimization.

The platform supports:

- Image optimization
- Working with thumbnails (previews)
- Flexible configuration of content storage for SEO

This makes it easy to manage all the components that affect the visibility of a website or application in search engines.



IT IS IMPORTANT TO UNDERSTAND

That most SEO optimization is performed on the frontend.

OneEntry, as a backend platform, provides all the necessary tools for developers to implement the proper mechanisms on the client side.



Therefore, statements that OneEntry is «worse for SEO» are incorrect. They mislead businesses about the division of responsibilities between the frontend and the backend. In reality, **the platform provides a complete set of tools**, while the final result depends on how SEO settings are implemented on the frontend of the application.



Question 10. «They might update something on their side, and your application will stop working».

Answer: All OneEntry updates are seamless and invisible to users.

This is ensured by a strict process:

- 1. Each update is first tested on staging applications to rule out inconsistencies with previous versions.
- 2. Data migration undergoes multi-level audits, guaranteeing correct transformation and preservation of information.
- 3. Updates are implemented without interrupting the operation of websites or applications, so the business continues to function as usual.



Updates make the platform more stable and secure without disrupting the operation of your application.



Question 11. «Right now they have attractive prices, but later they will raise them».

Answer: OneEntry's pricing policy was designed from the start to provide businesses with significant savings in both money and time. This gives us a competitive advantage. **If the platform stopped delivering tangible benefits to users, it would lose its value in the market.**

Yes, over time there may be a need for reasonable price indexation due to inflation or changing market conditions. However, such adjustments will always be proportional and transparent. For businesses, the key point is that using OneEntry helps optimize expenses and preserve freedom in managing their product.

These values were built into the philosophy of the platform from the beginning and remain its foundation.



P.S. It should be understood that most recommendations from developers are primarily aimed at securing their own benefit, whether in time or money.

What initially appears to be a profitable offer often turns into additional expenses for support, modifications, or hidden fees over time. The most dangerous part is that this does not happen immediately, but only when the business has already become fully dependent on a specific contractor.



OneEntry was created to establish honest and transparent relationships, where businesses maintain freedom, independence, and confidence in their infrastructure.