Launch your website, even with shortcomings

Am I drawing a conclusion right away with the above title? Almost, but not quite. A more nuanced story will emerge as you read further and you read between the lines.

Is it really worth to launch an incomplete website? Definitely yes.
A priority must be set on what type of content is necessary at the start and which ones can be added gradually in the future.

What elements were enough to start a running a webshop that presents the client's art? Photos and data's about the artworks, artist's contact information, a section where we upload the questions that are more likely to be asked by those, who wants to buy, rent available art piece(s) or to order a similar (sold) one.

The right backbone of the website

The ideal CMS (Content Management System) behind the e-commerce serves two different kind of end user: a non-technical and an another person, with various technical skills.

The first case is the client who wants to learn the usage of the CMS's administration panel in short-term. It should be mapped quickly and thanks to the transparent and intuitive UI (User Interface) it may become familiar in no time. Besides, we can even help by handing over the system with the appropriate authorization levels.
What I meant by this is, that if it is enough for the customer to change product prices or correct possible spelling mistakes or typos, than it can be quite confusing for them to see menu points as Advanced Parameters, HTML Block or Modules and Services. It is more than enough to have access to parts such as the Catalog and within it the Products options.
After selecting the product to be edited, there should be only one click away from modifying the price or some text parts of the short and/or long product description.

The character of the second case is a person who:

- wants to write and edit HTML,

- override Bootstrap template with custom CSS elements

- and prefers PHP programming language for backend scripting.

There is a secondary role connected with the above-mentioned primary one: operation with the CMS.

Even without system operator experience, or writing a command line script, it would be great to add new users with different authorization levels, granting and denying access to certain settings or functions.

In the graphical user interface [GUI], it is only needed to click on the checkbox and save the current settings.

In a similar way, updating installed modules, adding new ones or deleting existing ones would be just as easy.
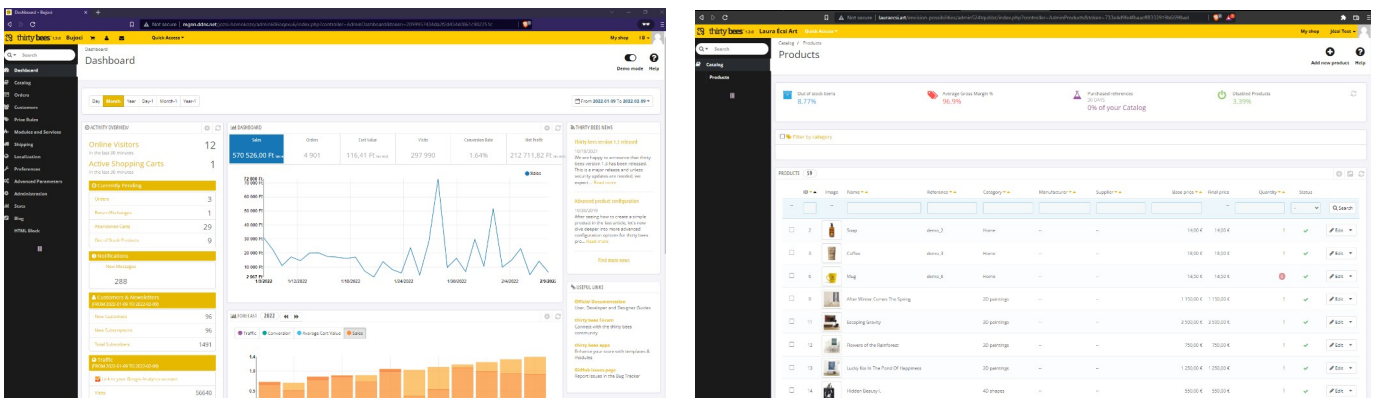
In the menu bar on the left, you can see how much a super admin has access to, and on the right, it is shown that this amount of authorization is enough for a basic user, for a quick job done.

The best solution - not just only from business perspective
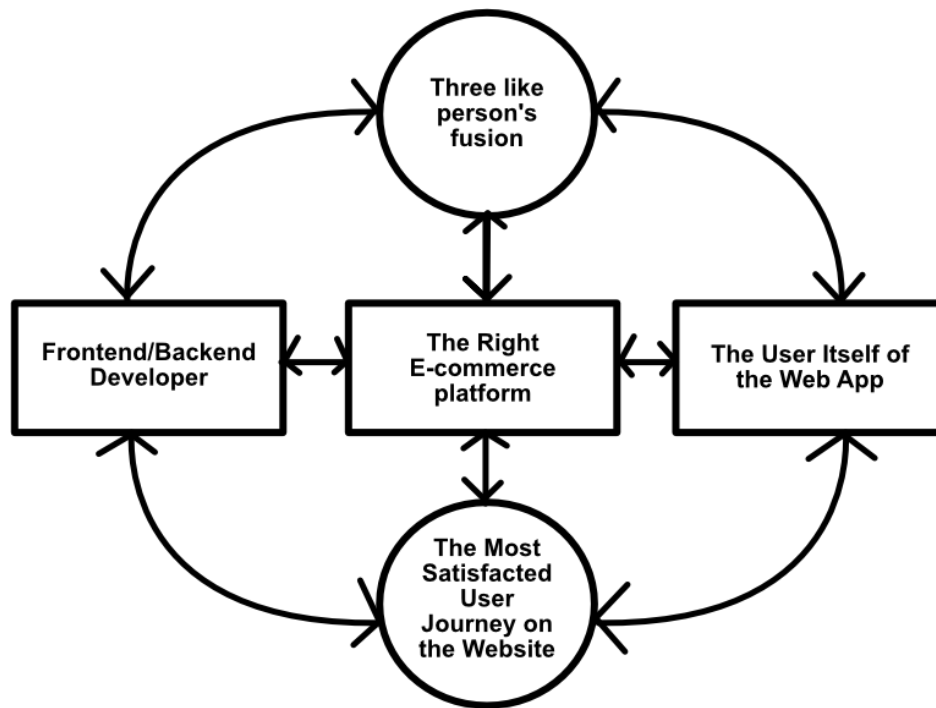
Thirtybees, an open source e-commerce platform, exceeds all of the previously written expectations. Including the appearance, ease of handling which is important for the customer. The template engine for PHP called Smarty, facilitates the developer's job. But how:

The PHP code and the code for the design are in separate files.
I must say that it gives less opportunities to make mistakes as an individual.
Even that this setup is more ideal when a developer and a designer works on a website at the same time. With this solution they cannot interrupt each other's work.

The structure of the components are clear and logical, it's fast to scan it.
The syntax is also easy to understand. Replacing new variables, tags and changing the existing ones is simple.

The community not just developed several types of plugins, but they are constantly updating it to avoid possible vulnerabilities.

Test or not to test

All participants of the project can only enjoy the positive side effects of testing: the customer, the developer, the content writer and especially the future visitors of the website. With the confirmation, that the webshop meets the basic requirements, it can be launched for the general public.

It's not about that I'm not interested in quality, but as a frontend developer I can't objectively look at the e-commerce site I have created. Besides this, errors may occur if a person works under pressure or because of a simple human error, misspellings, typos may be left in source code. These may be discovered more faster by an external person.

So I asked a friend of mine to test in the early stage of the website build. With this decision I have saved not only development time, but also time for fixing, if an error appears after all.

The webpage was examined according to various aspects such as usability, accuracy, efficiency, etc..
Among the manual and automation testing methods, exploratory testing was deliberately chosen to simulate the possible behaviors of the end user in a short time by testing the most important functions on the website.

Imagine that you start by scanning the user interface, then you continue testing the main functions in the following order:

1. Main page (Home page)
2. Navigation menu
3. Sub-pages
4. Procedure of adding item(s) to the basket/cart
5. Checking the content of the basket that does it contain all the appropriate items
6. Looking at the ordering process and filling out the required or non-required information
7. Confirmation of the successful order has been recieved or not

The examination of extreme values and API requests has also taken place.

It was necessary to ensure, that the number of available item is actually included in the cart, and more than one item cannot be purchased for the price of one item.

It turned out, that the quantity of the product can be rewritten via the browser. So the function of decreasing and increasing the quantity of the product had to be removed.

Truth is, there would have been no point keeping it, as each painting is unique, only one piece is made of each. In the product data sheet, under the product name, "I am the one & only, one of a kind item." text also confirms this statement.
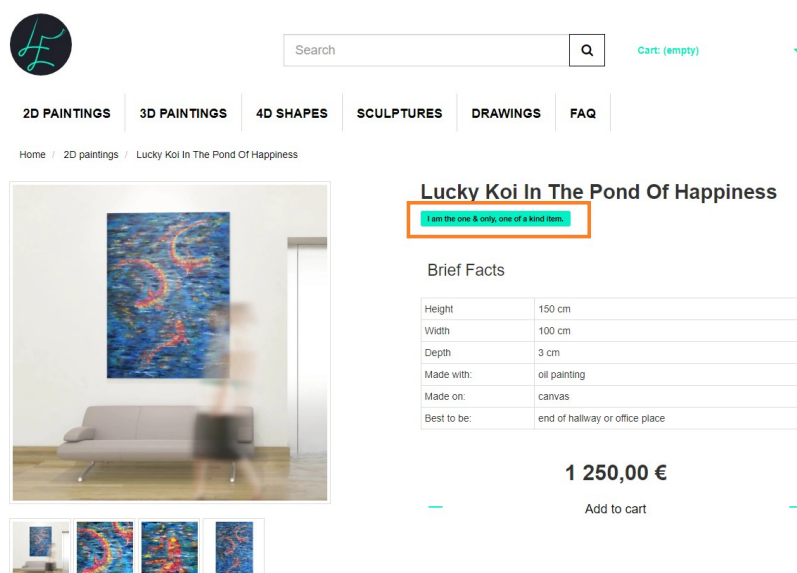
With the text below the product name, we emphasize even more that this is not an image produced on an assembly line.

What's the next step beside constantly uploading new abstract arts?

To correct all the errors that arise in the early, but already launched, phase of the project. We would not classify them as a critical issue, since they do not affect the operation of the site. It is more than likely, they will be unnoticed by website visitors.

Creating an environment as natural as possible. Even if it's virtual.

In our present case, if the interested party likes a painting that has been sold, he/she must write a separate e-mail to the artist. It can be done by clicking on the e-mail address displayed on the page's footer or by typing the mail address manually into the e-mail system used.

In addition, the fact that the price of the already sold work is no longer listed on the product sheet, nor is there a button for what to do there, can also cause a worse user experience.

For this reason, instead of the 'Add to cart' button, a 'I want a similar' button will appear here. This also helps website visitors by not having to think about the purpose of a product page without a function.
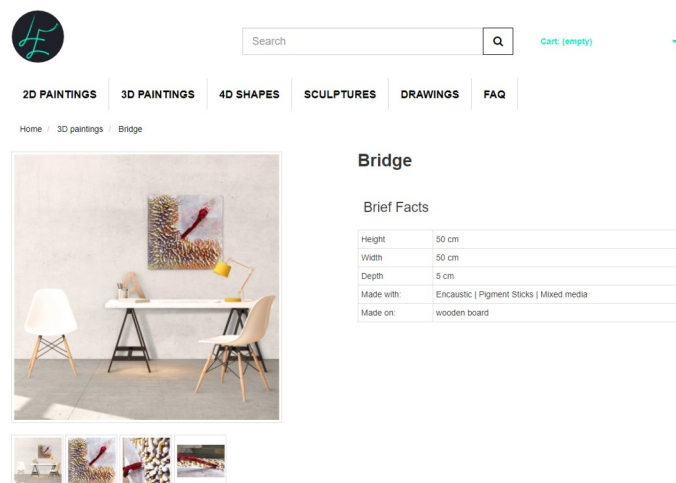
It is not indicated that this product has been sold, and there is no button encouraging to do an action, the visitor cannot do nothing or can be confused with the datas provided here.

After this, the following subpages in chronological order will be created:

A timeline exhibit.

You may seem earlier in museums an installation with vertical solution. But here, the chain of exhibitions will be in a horizontal way, descending by date, from top to bottom. Since it will be more easier to scroll down with the mouse or by tapping on a mobiles devices screen.

The top date always means the artist's participation in the earliest organized exhibition. By clicking on it, the website visitor can view the photos of the particular event.

Blog

The most interesting part of this topic is, that you can tag within few clicks related products to the common article and they will appear on the bottom of the page. This type of cross marketing is less violent than a neverending pop-up action during the check-out which is asking: „Are there any other supplement needed?"

To be continued

Even that we barely scratched the surface, it's a hands on experience what you have read, based on a true story. Which does not end here.

It can carry on presenting even more details how a user can for example change prices or correct text in product descriptions, or how to set the authorization levels, or what the users see with the privileges set up for them. Or should we focus on the testing process itself?

I am also open to other ideas as well. Just send an e-mail to fi.izabella.farago@gmail.com and I will come back with the most requested topic in a form of an article.