

E-commerce platform for creating online shops

1. Summary	1
2. Categories	1
3. Products	2
4. Orders	2
5. Shop	2
6. Technology	3
7. Screenshots	3

1. Summary

The main idea is to have a main application that will generate, manage and publish online stores/shops for merchants. In this document we will refer to merchants as customers to the platform and use the term customers for the actual users that will buy products from the shops, or customers to our customers. Our customers will register to our platform and with or without a sample shop and will be able to create and manage multiple shops, categories, products, view orders from their customers, generate reports and see statistics for number of orders per day, week, month and more in depth reports. We will convert the data and use AI for better user experience and give predictions to our merchants.

Merchants will register to our platform, but their customers will register on their shops. Each shop will be a new site/domain separate from the platform merchants are registered on. Each merchant will be able to see their customers' details and ship their orders. Merchants are associated as user accounts and they can create multiple users under their account.

We would offer price plans to the merchants based on the number of shops and different kinds of functionalities we will offer.

2. Categories

Merchants will create categories which will represent the menu in their shop. The menu builder is dynamic so you can create nested menus and choose to show or hide them from the shop.

3. Products

All products are categorized. When creating a product, the user will choose a category the product belongs to. This filtering will be used to show the product under the exact category/menu on their shop.

Type of products:

- Simple products, users will input only price, quantity, if the item is in store or not.
- Configurable product, which is an extension of the simple product but will add additional attributes like color, size, etc. Attributes are separate modules and the user will configure them separately and then when creating a product will choose the right attribute from a dropdown. These attributes will generate so-called variants and the idea is that some of the variants can be different price, ex.a t-shirt size XL and color Blue might be a bit more expensive than the other variants. In the actual shop this translates to the customer choosing the right combination of attributes and showing the right price.

Products are managed based on the number in stock when order is being created, meaning the number is described based on the quantity in the order and this validation is used if the customer will be able to select the product.

4. Orders

Orders can be created by registered users or anonymous with providing at least email address. The flow of ordering also includes forms for gathering input from the customer about billing and shipping address. Payment through cards will be offered in the future as well as details about shipping and tracking orders from merchants to customers. Merchants will see the orders in the platform and we will implement a flow for approving or rejecting the order. Merchants and customers will be notified via email about the details of the order.

5. Shop

Each merchant will create a shop to display his products. The shop will have a landing page with images that the merchant will upload. There is a search page where the customers can filter the products based on attributes, colors, sizes and other attributes that the merchant will manage and apply to products. There is a product details page for adding the product to cart. The cart flow will lead the customer to making the order. The shop also acts as a CMS so merchants can add pages(full word text) through the platform and be shown on the shop. The design is responsive for both desktop and mobile display.

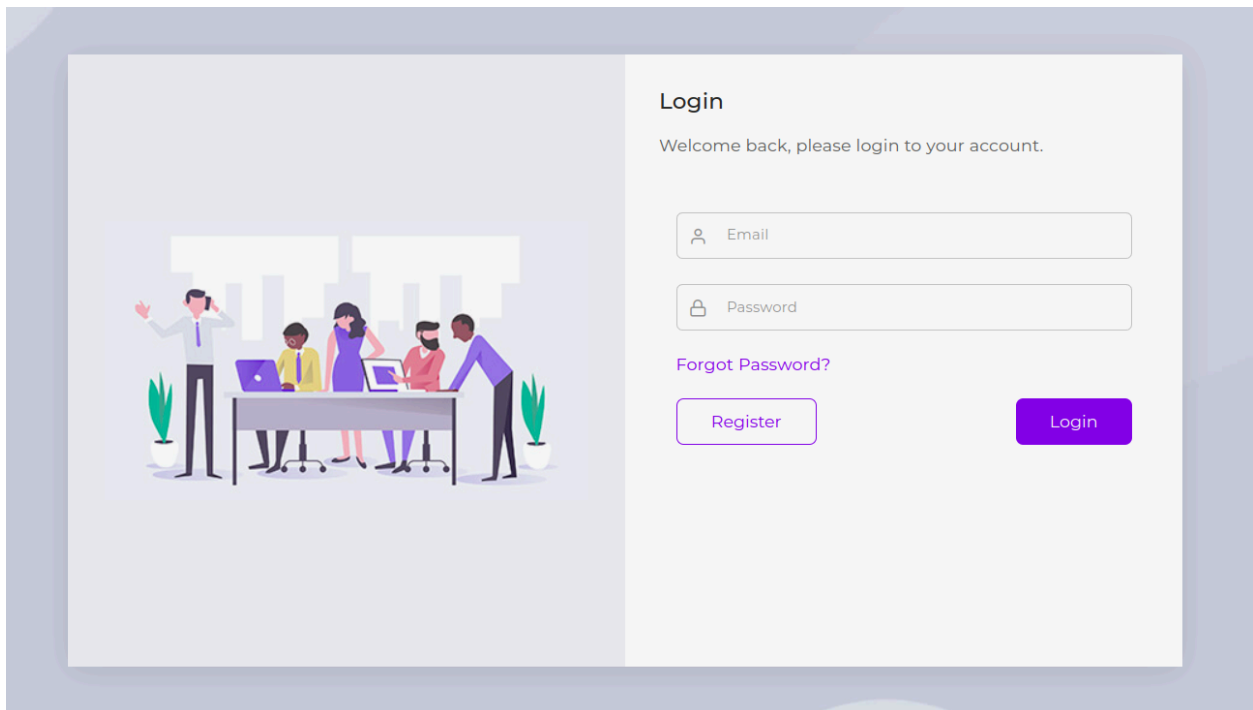
6. Technology

The platform is built using a microservices approach with different technology stacks.

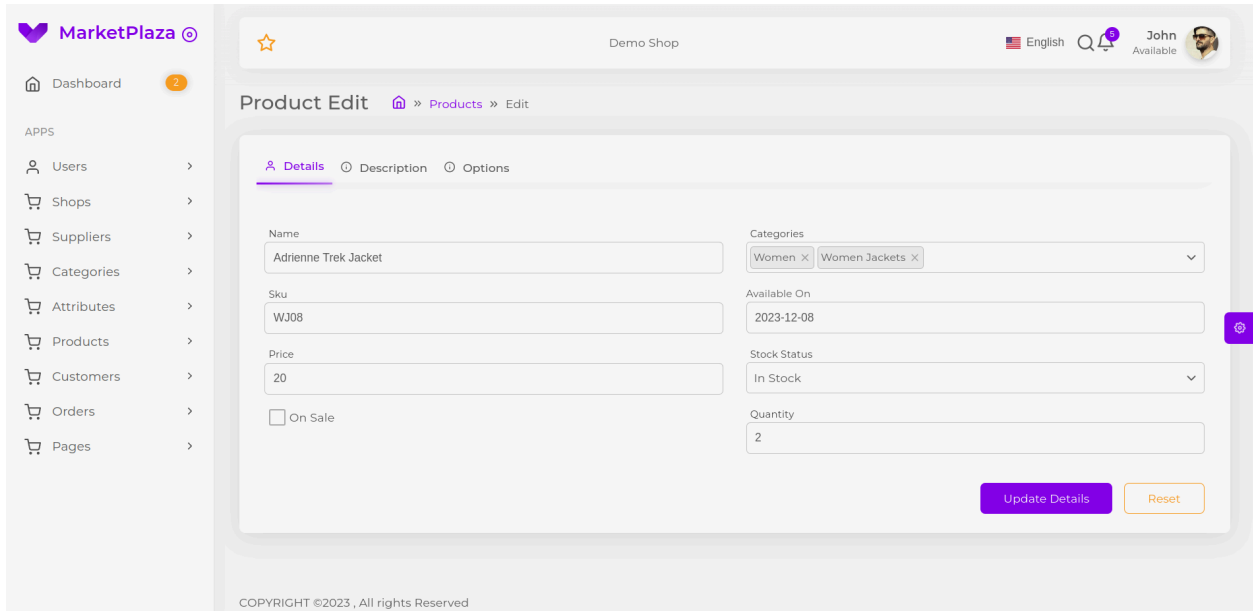
- Backend part of the services we use the Golang language.
- On The Frontend part we use the Nuxt/VueJS framework.
- Microservices are managed using kubernetes, so each of the services will be auto-scalable and configured to use resources. This will include redirecting the requests from shops and the platform to the right backend service, manage load on the consumers, migrations, seeders.
- Managed services: Postgres database, Redis cache and Rabbitmq for message broker.
- RabbitMQ for event driven architecture. Some of the services send jobs to rabbitmq and we process them in the background to handle the load and better user experience. This will allow us to scale to lots of requests per second, rate limit requests to external systems and provision products quickly.
- Deployment will be done on some of the cloud providers (aws, google cloud).

7. Screenshots

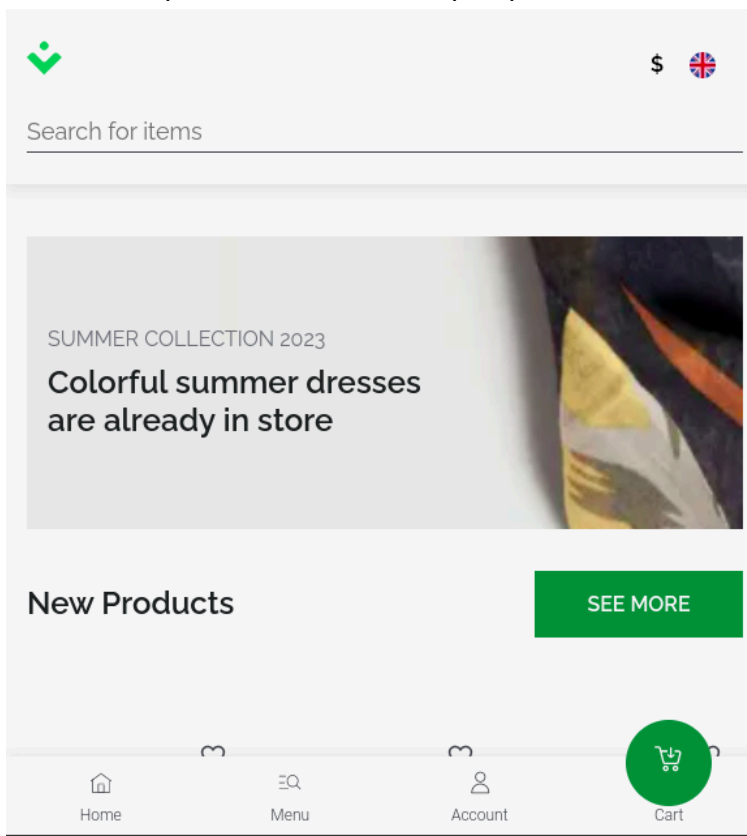
Login page for the platform for the merchants.



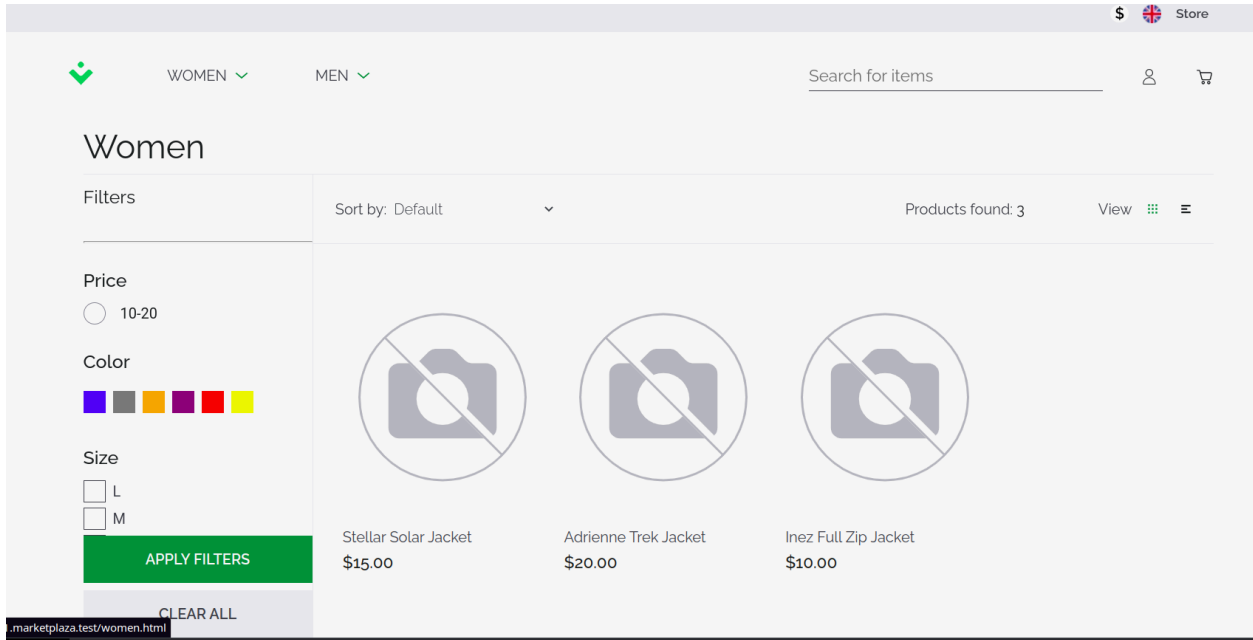
Editing of products on the platform. You can also see other modules we offer at the moment on the sidebar on the left. Theme is customizable and we support dark themes as well.



How the shop looks from a mobile perspective. This is the front page when opening the shop.



This is the shop from a desktop perspective. This is the filtering page based on attributes.



Shop checkout page after a customer adds a product to the cart. After adding the details in each tab the order will be completed and the merchant notified.

