

English - SAE

Introduction (Maxime)

Casporama is a website about sport gear in four specific sports : Football, Volleyball, Badminton and Martials-Arts.

The user can navigate on the website to see different products, if they are interested in something the clients can create an account and then login to buy it. During the register they have to give his information such as his Name, phone number, address, etc. Later the contact details can be modified or even add multiple address up to 6 max. When the order is completed they can track it at any time. Another functionality is the monthly subscription to have some reduction on a lot of products.

The administrator have same specs but they have an administrator panel where they can manage users, products, orders and import and export to the Database.

Before buying a product they pass through a cart which contains all the desired products, they can save it, modify the present cart or the saved carts and delete products from the 2 types of cart. The price of the cart will be display with the VAT, the possible reduction you have (been apply directly to the subtotal of the cart) and the transport fees.

Many tools have been used during this project, this is a list of them :

- PHP with the framework Codeigniter3
- HTML with CSS
- JavaScript
- Kotlin
- MariaDB

Frontend

What we implemented :

- We used a template system to reduce the number of views in our project, because with this system, some views are reusable everywhere such as the header.
- We used resource caching technologies, such as HTTP caching, to reduce the amount of data downloaded by users when they access your site.

What we wanted to implement :

- We wanted to implement a dark mode, which can result in using less energy because the site displays less color, and OLED screens that turn off pixels to display black save even more energy.

Backend

What we implemented :

Optimization du traitement de données

- We used a cache system for the database, that way we don't have to query the database for every request, but we can use the cache to get the data faster, and if the data is not in the cache, we can query the database and then add the data to the cache, so that the next time we need the data, we can get it from the cache.

Framework eco responsable

- We used a framework that is eco-responsible, that way we don't have to worry about the framework using too much resources, and we can focus on the rest of the website.

What we wanted to implement :

Compression des données

Suppression des données non utilisé

Contents (Maxime)

What we implemented :

- The first measure for the content is, when an admin wants to export some data from the Data Base through the admin panel. They download a compressed file who has been created on the server for them and who will be delete from the server when there is at least five files with the same extension. This measure allows us to save space and avoid an overload of the server which gives us an low-consumption.
- The second measure that we opted for is using a correct text formatting tags, it allows the users to have access in an short time and an short amount of clicks the wanted page. Ideally this allows to reduce the traffic in the website which limit the usage of the data base.

What we wanted to implement :

- One measure with may improve the saving space would be to also compress the PDF invoice that we are sending by mail to the customers.
- Another measure would be, using compressed images which doesn't compromise the quality of it and like the first measure it allows us to save space on the server which leads to reducing our footprint carbon.

UX/UI

What we implemented :

- We made sure that the website is easy to navigate and understand, so that users do not have to spend a lot of time searching for the information they are looking for.
- We used visual elements to aid in understanding and navigation on your site, such as images, graphics, and icons.
- We used colors and fonts that are easy to read and not too demanding on the eyes, in order to reduce visual fatigue for users.

What we wanted to implement :

- We wanted our website to be optimized for mobile devices and for users to easily access the information they are looking for on those devices.

Architecture

What we implemented :

- We used a modular approach for the website, that way components of certain pages are really well reused so that it uses less storage space.

What we wanted to implement :

- We could have used a more up to date framework that would use less resources, but at the scale of what we did and what we could have done, it may have not have a big effect.
- We could have use another server for hosting the database, that way for every duplicate database that we hosted, either on our local machines or remote server, we could have use only one server, used it, but it would have used more bandwidth and that could have been an issue as well
- Hypothetically, we could have use a Content Delivery Network, if our website was public and of a massive scale, for less global bandwidth usage

Hosting

What we implemented :

- We have made our website wit the concern of having a low idle usage of the server, that way because its planed that it can be in a virtualised server, the resources not used by our server can be used by others
- We used a fairly old but proven web server : Apache2, which doesn't need much resources to run (but its just the web server so it's not a very consuming part of the website),

What we wanted to implement :

- We wanted to make a CI/CD pipeline for continuous testing and deployment, that would have been used for checking if every updates that we made did not break on the remote server, and also verify if the performance did not regress and not use more resources.

Bibliography :