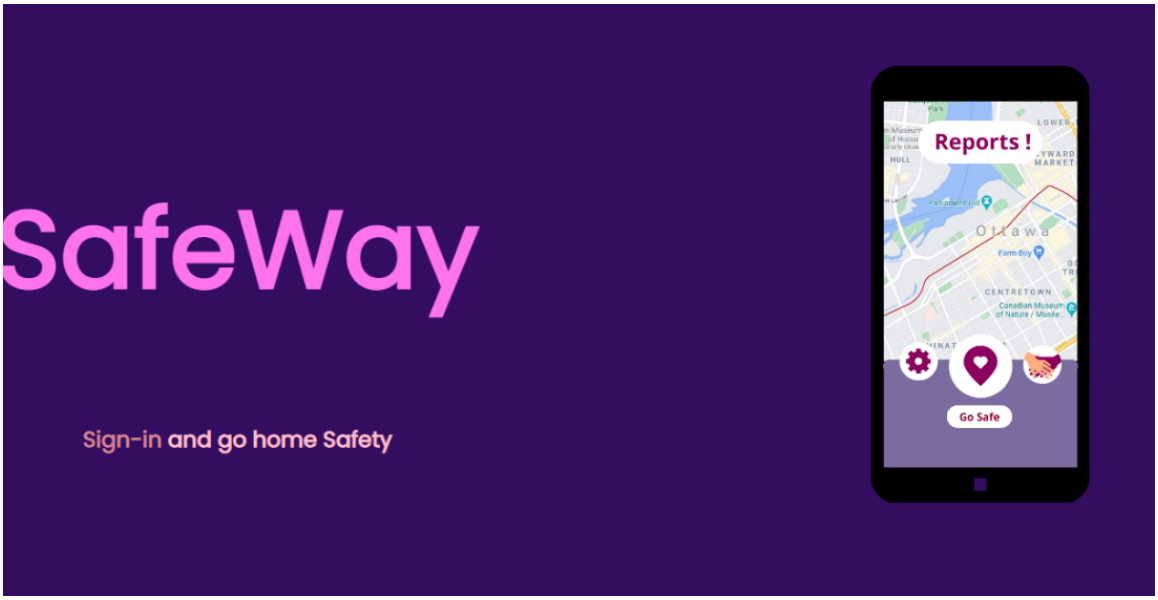
SafeWay

The Best Way for Your Safety



MOHAMED ZAMMIT CHATTI – RAYEN HADDAD – YOUSSEF MAZOUZ
ADVANCED WEB PROGRAMMING
Lecturer : DANIEL MAGO VISTRO

Table of Contents

Introduction

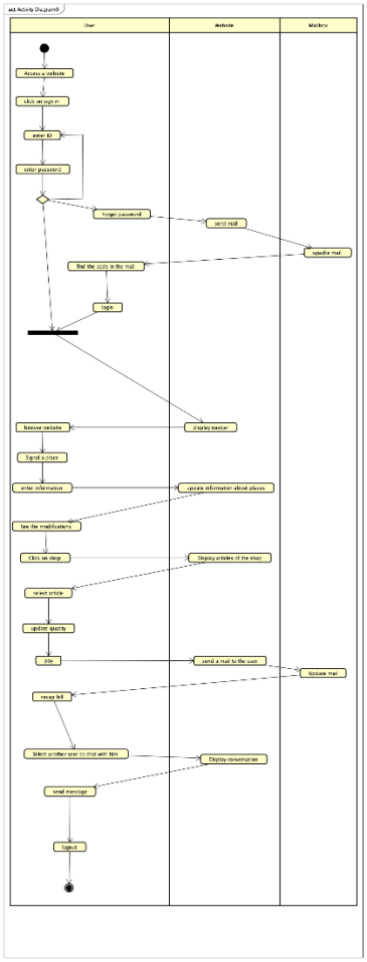
We are three students from APU, and we have the mission to develop a website in PHP, html and CSS. Before starting the development of our site, we needed an idea of subject. Nodaway it is complicated for vulnerable people to walk in the street without danger. The number of aggressions is increasing, and the feeling of insecurity is omnipresent in urban areas. It is with this in mind that we have developed SafeWay. SafeWay is a website that lists all the riskiest places nearby, so that you can avoid them and use a safer route. In this report, we will first introduce you to the features of our website and how we went about making this project a success. Then, we will explain the backend, the engine of our site. Finally, we will show you all the difficulties we encountered and how we managed to overcome them.

SafeWay, the beginnings of an ambitious project

Principle of the site and presentation of the functionalities

SafeWay aims to list all the dangerous places in your city and its surroundings. Users will have to create an account, once the registration is finalized the user will receive a confirmation email. If the user forgets his password, don't worry, he will have the possibility to recover his account with the forgotten password feature, he will then receive a temporary password that he will have to modify right after. Users will be able to report all dangerous places so that people at risk can avoid them. In order for users to communicate with each other, they will have a chat room available. In the chat, customers will be able to give details about the places to report, the chat is live so they will also be able to find each other in order to regroup and help each other in case of danger. There will also be a search bar available so they can quickly find the history of conversations. We have also made an online shop available to purchase self-defense items. Customers will be able to add the products they are interested in the shopping cart. Customers will be able to choose the quantity they want for each product. They will then have to proceed to the payment. Once the order is validated the customer will receive a receipt in his mailbox. Once the product is received the customer will be able to evaluate the quality of the product, he will give a note from one to five stars and will be able to put a comment about the product.

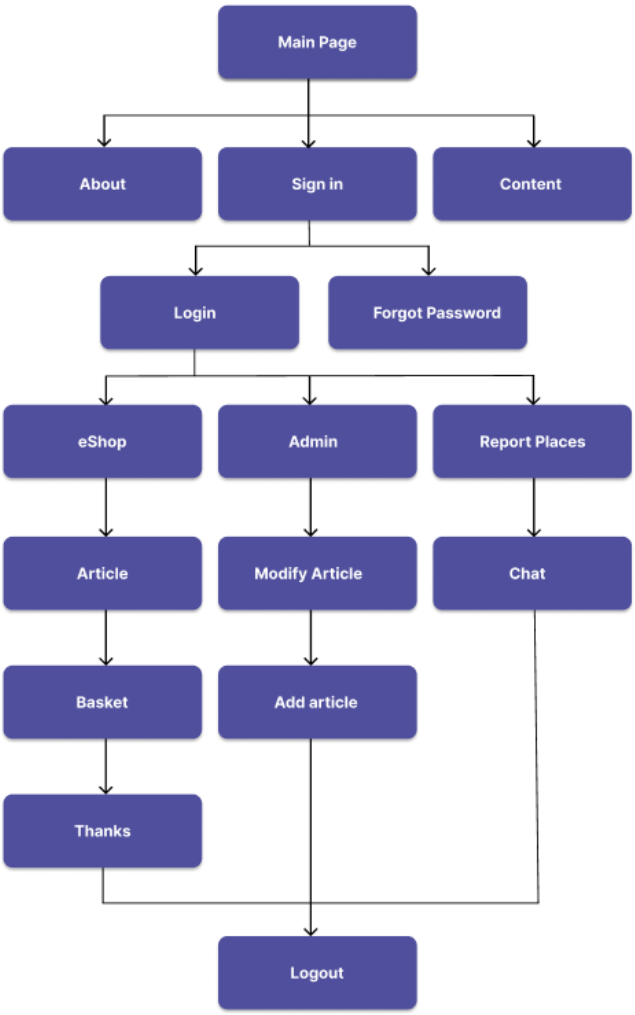
As for the administrator, he will be able to access the reviews in order to moderate the comments. He will also be able to add items and modify them, he will be able to put back the stock when they are out of stock. The administrator will also be able to change the price, and an email will be sent to users when the price is changed.



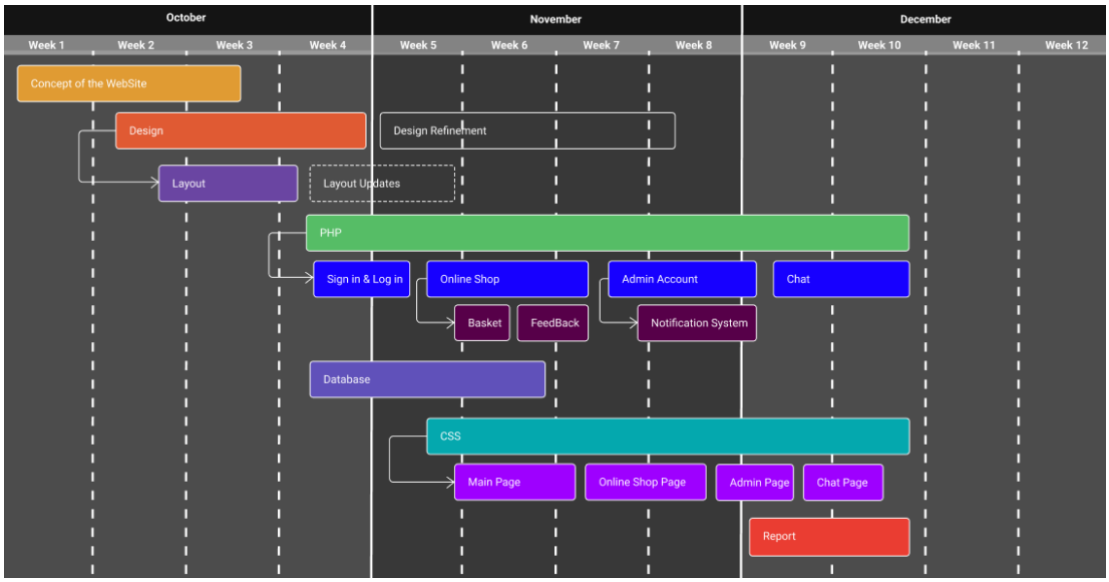
First of all, the user must access to the website, after he should sign in by enter his ID (here his mail) and his password. I the user does a mistake about his email/password (or both), he will be to require later. Also if he forget his password, he will trigger another process, an email will be send to his mailbox with temporary password, and then he will be able to change his password.

After, the user will browse the website and perform several actions. He will signal different place where harassments and assaults happened. Also the user will be able to speak with other people, or buy some product to protect himself.

The design of the Web Site



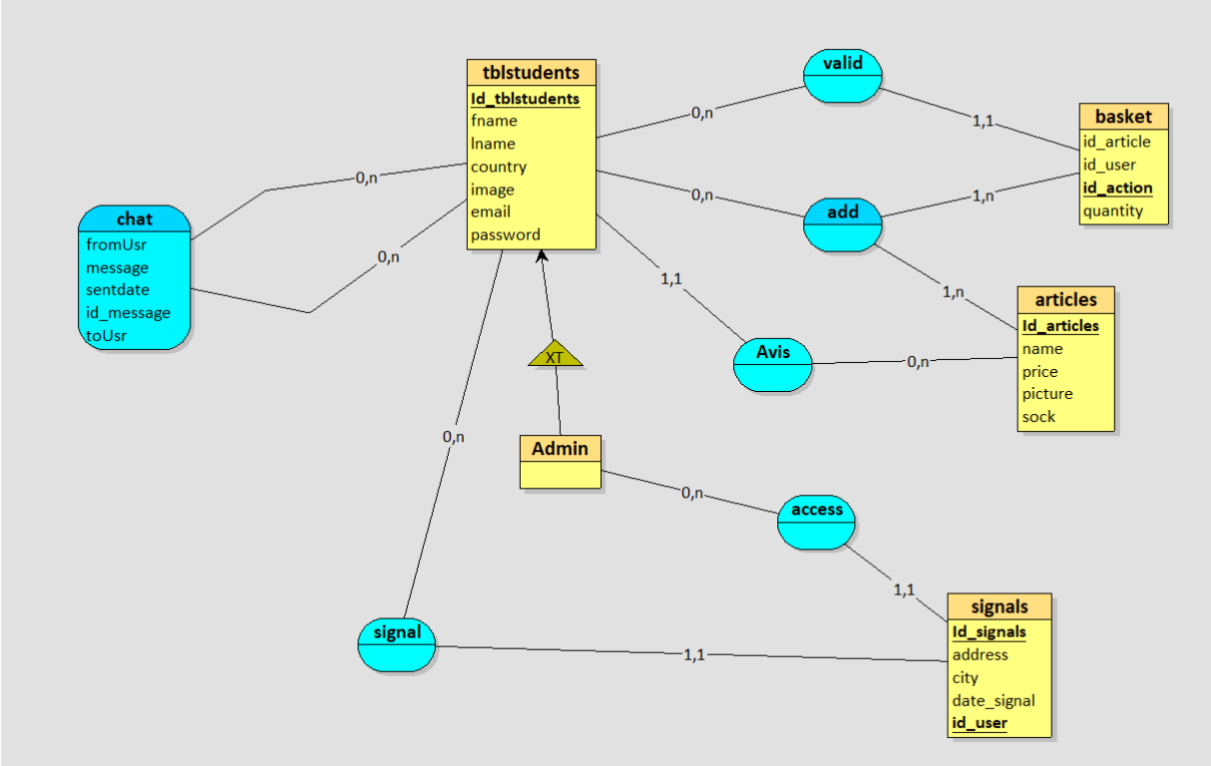
A well-organized project



Look On the Code

The Database

The first step of a creation of a website is the implementation of a complete database. To explain this database, We create a scheme where we can see all tables and links between them.



- Tblstudents: The main table of the website, it represents all users of the site. Every user has a unique ID, a first name, a last name, a country, an email, a password, and a picture that he can change.
- Admin: He's a user with more rights and actions. To recognize the admin:

- his mail is rayen.haddad@efrei.com

-his ID is 1062

Signals:

This table represents all signals of dangerous locations signaled by users. It is composed of a unique ID, an address, a city, a date, and the user who makes the signal. A user can signal locations and the admin has the access of all signals.

Articles:

All article is an instance of the table ‘article’. They all have a unique ID, a picture, a name, a description, a price, and a stock.

Basket:

This table is like a big card common for all users Where every line of the table indicates a transaction between the user, an article, and the basket. It contains a unique id_action, an id_user, an id_article and a quantity. To access the card of the specific user who is connected we must target the ‘id_user’ in SQL requests. So, we have all specific cards with a unique table.

b. Link website and database using PHP

To connect database with our website we must use PHP. Let’s explain the process of linking database and website with a concrete example.

So, we want to display all articles of the website in the shop. And to do this, we must search all articles into the database. From the database to displaying articles in the shop there is 4 steps.

1) Connect the page with the database

There is code below necessary to connect them and we are going to explain it.

```php
<?php

$host = 'localhost';
$user = 'root';
$passwrd = '';
$dbname = 'webtest';

$connect = mysqli_connect($host, $user, $passwrd, $dbname);

if($connect === false){
    die('error in connection' . mysqli_connect_error());
}
```

Firstly we have to define the host called here ’localhost’, the user here Is ’root’, the password which is an empty string and the name of the database which is ’webtest’.

After, to connect everything we simply use the native function ***‘mysqli_connect()’***.

To finish, we display the error if the connection doesn't work correctly.

2) Sql requests into PHP

After connectiong database and the code, the second step is to write our SQL requests. We use 3 types of requests.

1. Insertion

Let's take the example of the creation of a new article.

```
$query="INSERT INTO `article`(`name`, `picture`, `description`, `stock`, `price`)
VALUES ('$name','$imgContent','$desc','$stock', '$price')";
```

We have to create a variable '*$query*' which contain the request. We just must to insert into the table **'article'** variables corresponding of every value of the table.

2. Search

Like the insertion, we put in a php variable our request which here is a research. The request below is the request that we use in the shop to find all articles.

```
$query = "SELECT * from `article`";
```

3. Updating

Let's take the example of modifying the picture of an article.

```
$newpic="UPDATE `article`
    SET `picture` = '$imgContent'
    WHERE `ID` = '$id'";
```

Here, we update the picture in the table article we the ID of the article equals to current ID.

3) Query

The ***'mysqli_query()'*** function performs a query against a database. The function links the connection to the database with the request. Here is the research of all articles in the shop.

```
$result = mysqli_query($connect, $query);
```

4) Fetch

The '***mysqli_fetch_assoc()'*** function fetches a result row as an associative array. To access to every single line of the table we can use a loop where every recurrence of the array corresponds to a row of the table.

```php
while($row = mysqli_fetch_assoc($result)){?>

    <div class="columns portfolio-item">
        <div class="item-wrap" style="border: 1px solid white ;">
            <a href="article.php?id=<?php echo $row['name']?>" target="_blank">
                <img alt="" src="
                <?php echo base64_encode($row['picture']); ?>" style="width: 100%;
                height: 100%;">
                <div class="overlay">
                    <div class="portfolio-item-meta">
                            <h2><?php echo $row['name'] ?></h2>
                        <p><?php echo $row['price']; ?>$</p>
                            </div>
                    </div>
                <div class="link-icon"><i class="icon-plus"></i></div>
            </a>
        </div>
    </div> <!-- item end -->

<?php } ?>
```

Here, We use a loop ***'while()'*** where *'$row' correspond to every row of the table* **'article'** to display all articles in the shop*. For example, if we want to show the name of the article, we use the variable* `$row['name']`

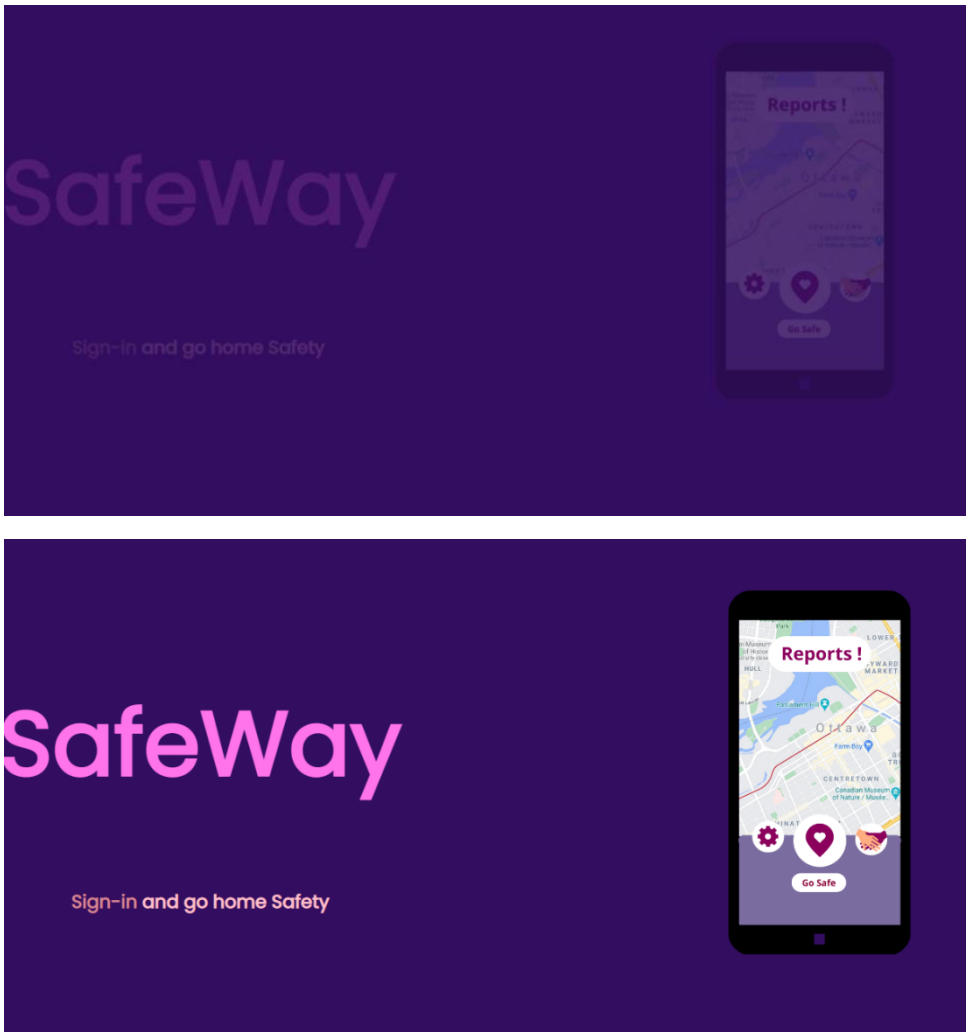
c. Dynamic with JavaScript

the use of php in a website is not enough. Every website must have a JavaScript part to dynamize all the website. JavaScript could have many utilities in a site.

1) Animations

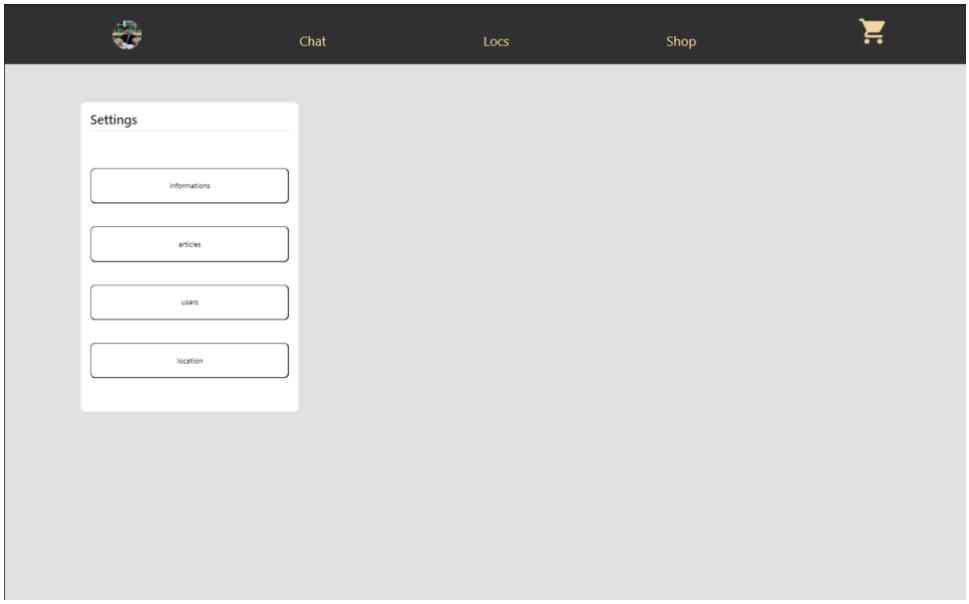
```javascript
/* -------------- Animations ------------ */

const handleIntersect = function(entries, observer) {
    console.log("entries", entries)
    entries.forEach( (entry) => {
        if (entry.intersectionRatio > ratio){
            entry.target.classList.add("visible");
            console.log("entry", entry)
        }
        else {
            entry.target.classList.remove("visible");
        }
    });
};


var observer = new IntersectionObserver(handleIntersect, options);
observer.observe(document.querySelector(".reveal"));

for (i = 0; i < 9; i++){
    if(document.querySelector(".reveal"+i) == null){
        console.log("null")
    }else{
        observer.observe(document.querySelector(".reveal"+i));
    }
}
```

The code above is used for the animation of the loading pages, When a element enter in certain ratio of the screen we use the function ***'observe()'*** to lunch animation which corresponds of adding a class called here 'reveal'. There is here an example of the usage of the script above.

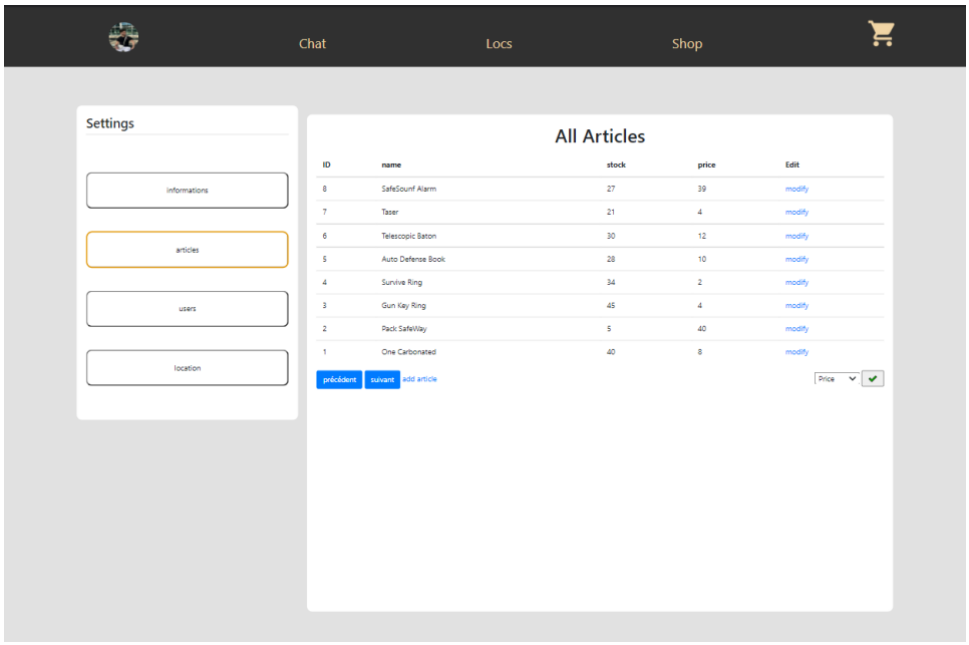


2) Dynamic Displaying

We also JavaScript to create new elements in a dynamic way. If we need for example to create element only under certain conditions, We are going to code in JavaScript like the code below. Let's explain the example of the article rubric in the admin section. When the page is loading, there is no rubric open. We must click on of the buttons to open the corresponding rubric and the page look like that.



Then, When we click on the article button, The Page will display the article section.



This dynamism is executed thank's to this code.

```
articles.addEventListener("click", () => {
    disp_info.classList.add("invisible");
    disp_loc.classList.add("invisible");
    disp_user.classList.add("invisible");

    disp_articles.classList.remove("invisible");
    supprimer()
    getArticles(0);

});
```

When the element '*articles*' corresponding is clicked we start with adding the class invisible to all other sections and remove it from the current section.

3) Creation

After accessing the section article, we have to create the table which contains all articles we firstly have to create the table and the header of the table with the native function ***'document.createElement()'*** and we put this creation into a variable.

```
/* ---------------------- Initialisation table ---------------------- */

    const tab = document.createElement('table');
    tab.classList.add("datatable");
    tab.classList.add("table");
    const titres = document.createElement('thead');
    const l1 = document.createElement('tr');
```

After We create all titles of the header in the same way.

```
/* ----------------------- table articles ---------------------- */

    const artID = document.createElement('th');
    artID.textContent = "ID"
    const name = document.createElement('th');
    name.textContent = "name"
    const stock = document.createElement('th');
    stock.textContent = "stock"
    const price = document.createElement('th');
    price.textContent = "price"
    const artEdit = document.createElement('th');
    artEdit.textContent = "Edit"
```

Then, We use just have to use the function ***'append()'*** to insert the child in the parent element.

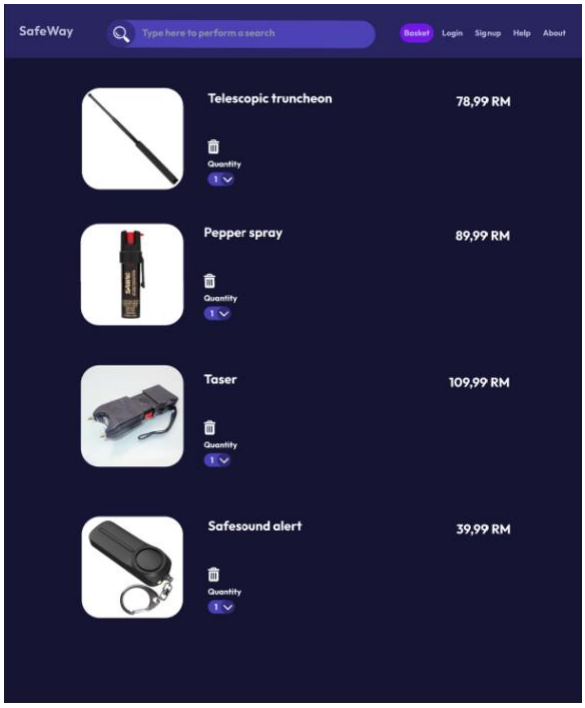
```
newItem.append(articlID, articlename, articlestock, articleprice, artedit);
tbody.append(newItem)
```

d. Style with css

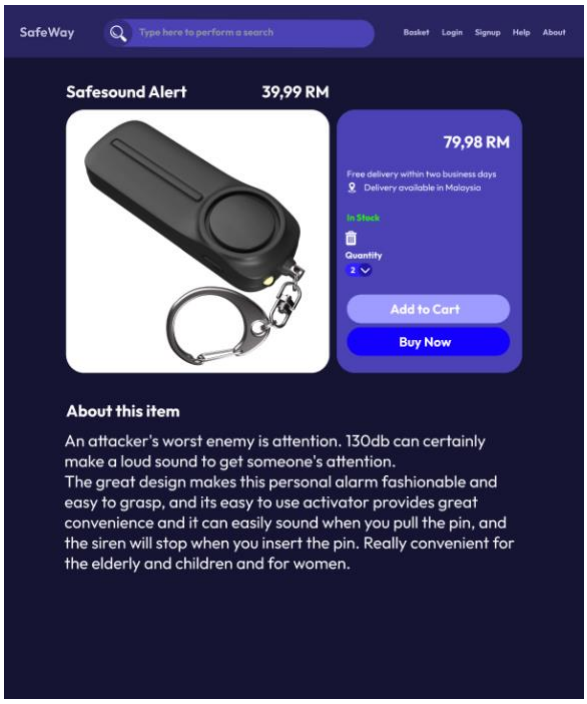
Once the concept of the site, the functionalities, the php and the JavaScript were in place, we thought about the design of SafeWay. For this we used figma.

Here is a first look at our online shop:

We can see a first Vue of the card.



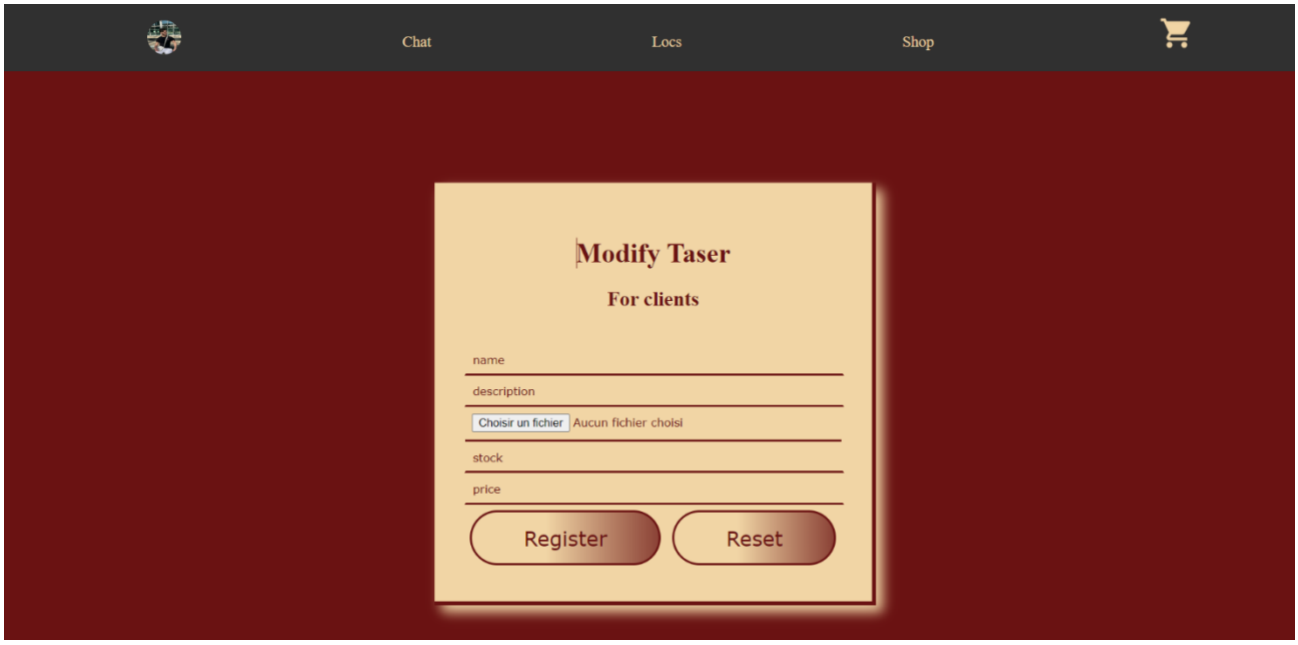
And here a first idea of the page of an article.



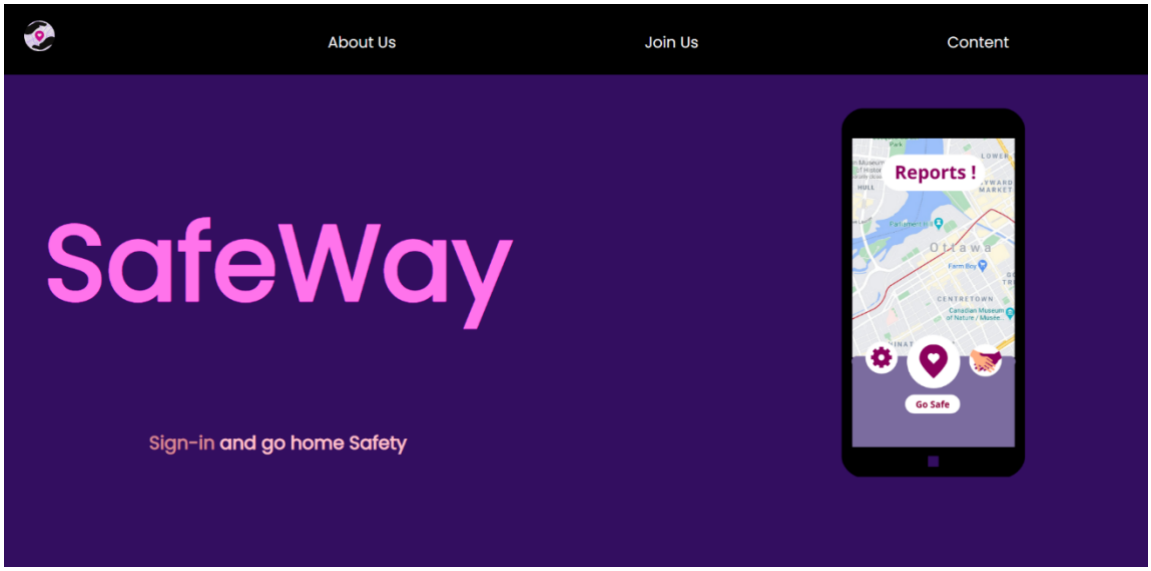
We will split this part in 2. The main design and the responsive design.

Main design: We chose to make a website with a colored theme.

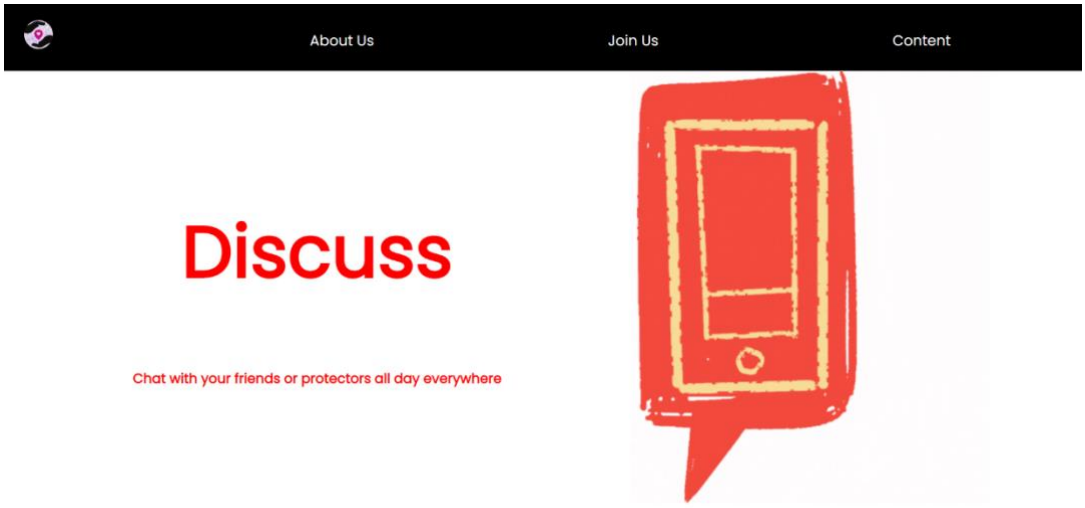
Forms: we chose similar color theme for all forms (registration, modification).

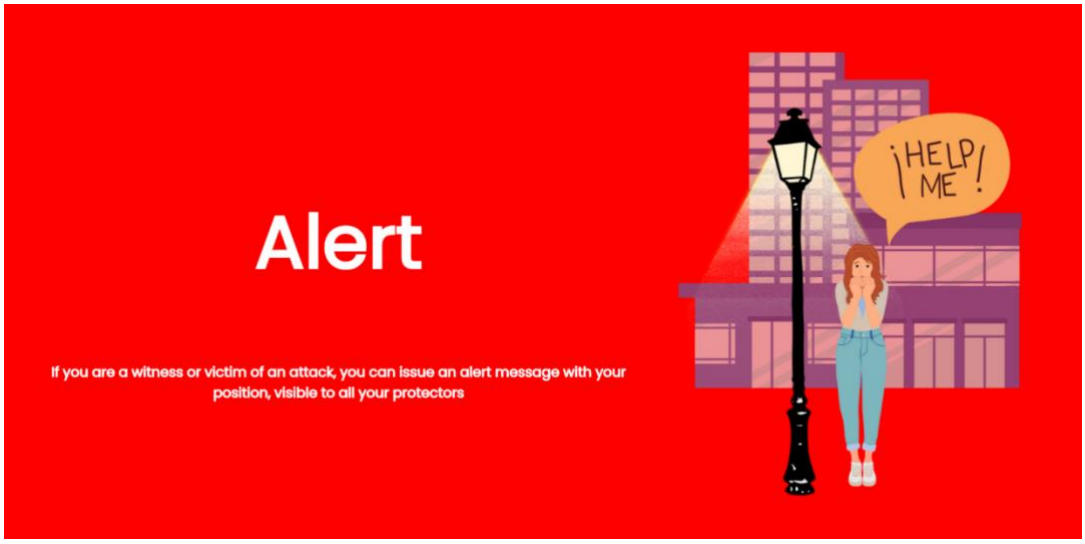


Main Page: For the main Page we choose a mixt of dark and light purple.

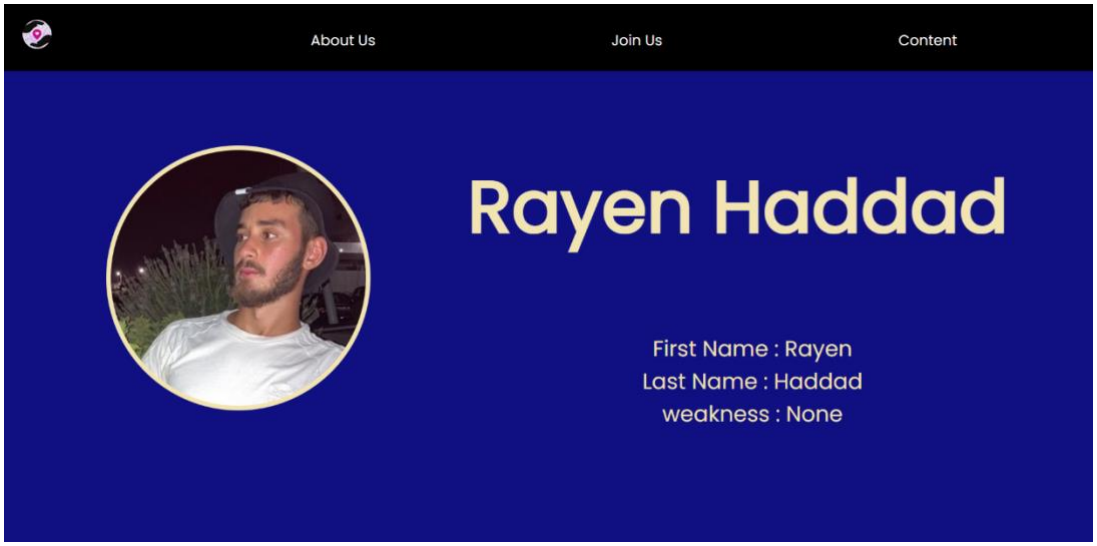


Content : For the content part we chose a mixt of white and red





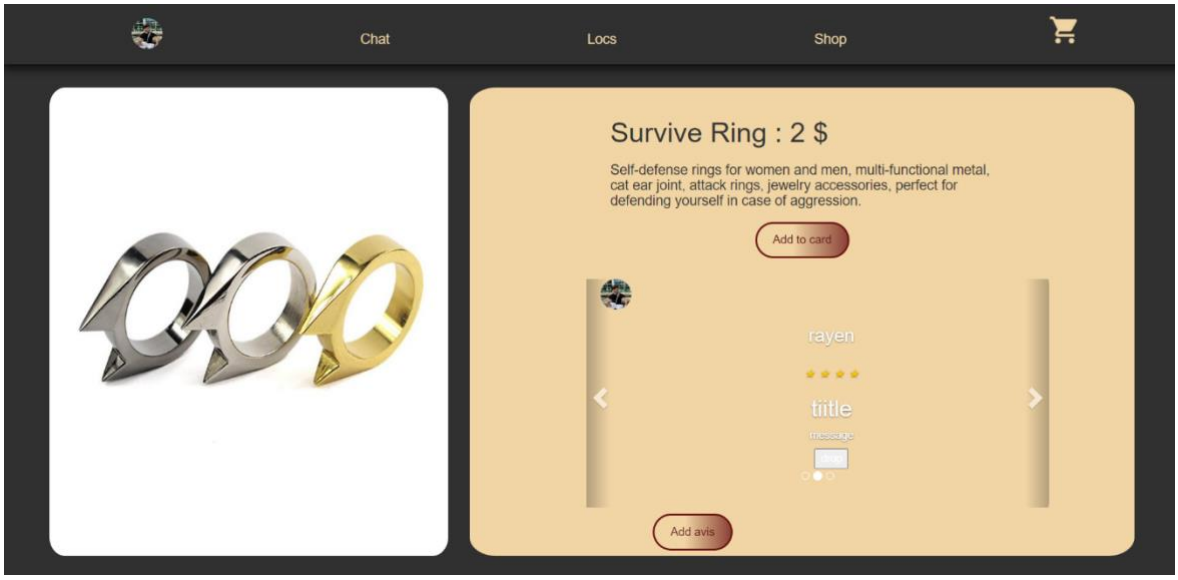
About us : For the About part we chose à mix of blue and yellow.



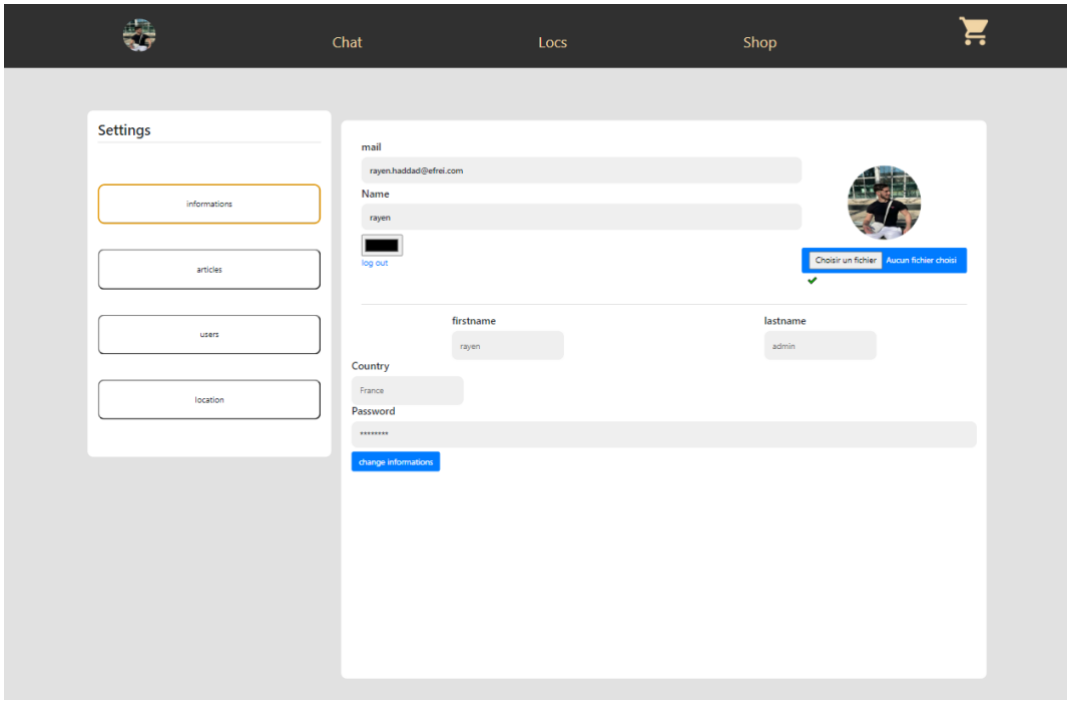
Mohamed Chatti

First Name : Mohamed
Last Name : Chatti
weakness : to strong

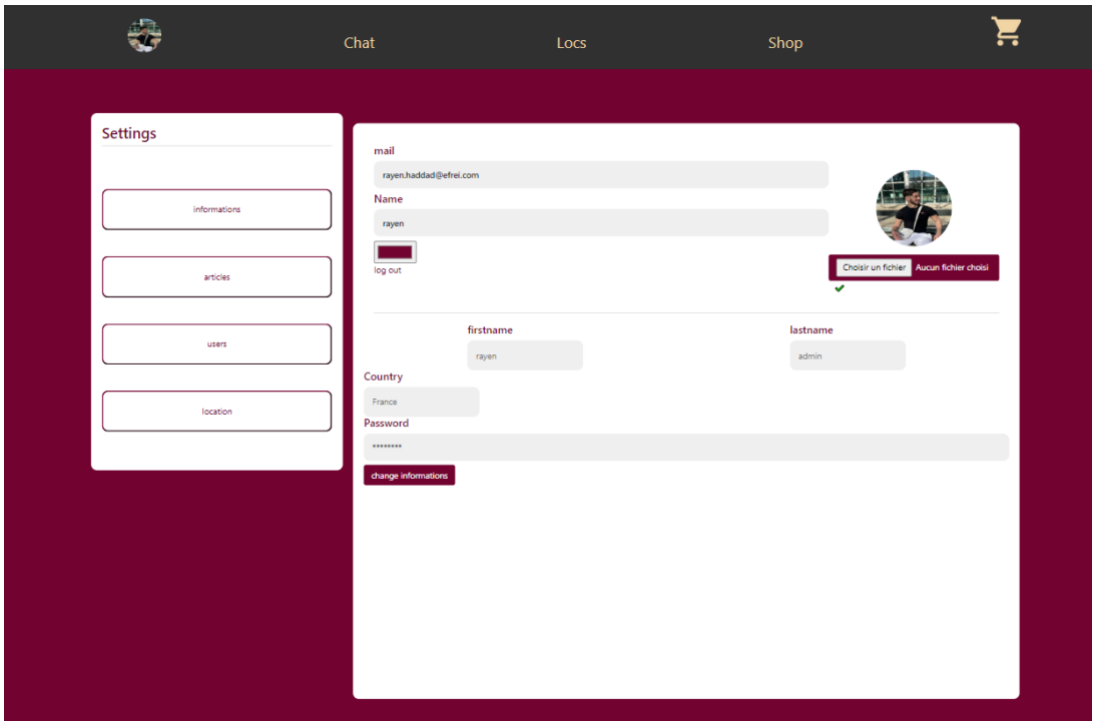
Article : For the article part, we chose a mix of dark gray and yellow



Admin : For the admin part, we chose a main theme in white and gray



But the admin can choose himself the color theme.

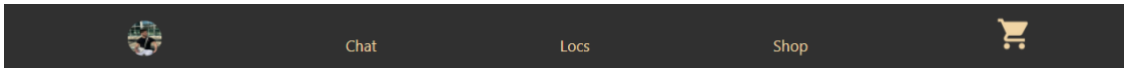


Header : there is 2 distincts headers in the site

Not connected

We can see 4 differents links. The icone of the site bring to the main page, we can also go in the About, Registration, and Content section.

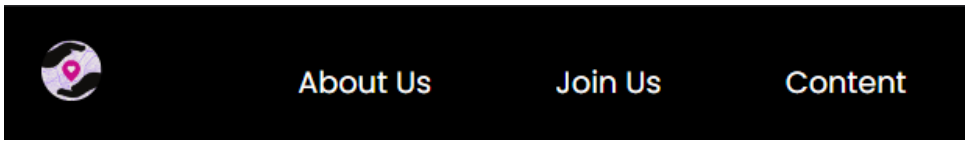
Connected



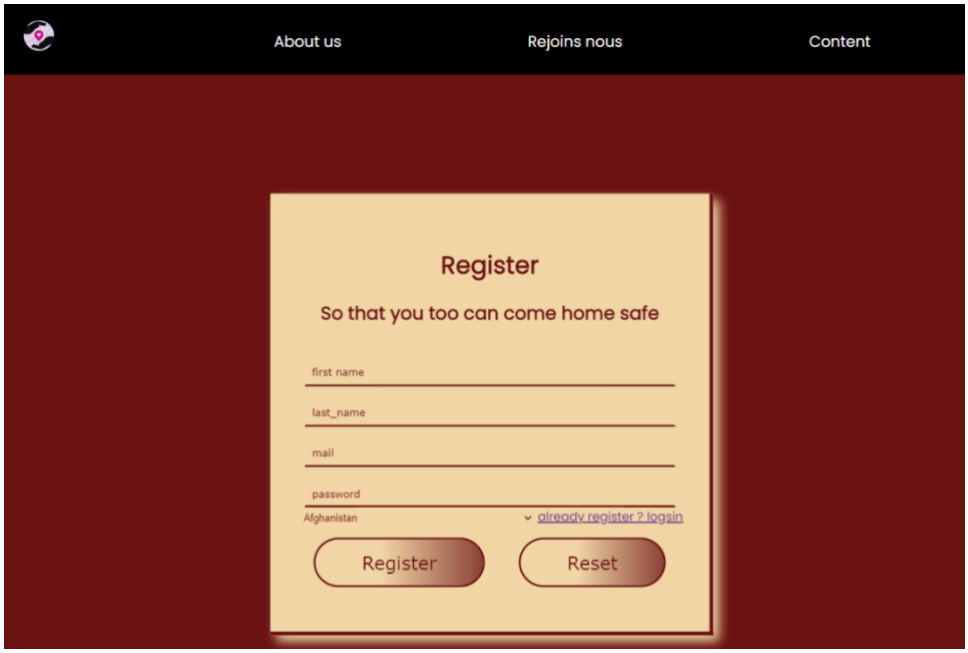
The header changes completly when we register on the site. We can see our profil picture that bring us to our admin/user section. We Can also go to the chat, the shop or signal locations. The caddie bring to our card. The caddie can change depending on the number of articles present into the card. For example, if there is 2 articles, the header will be like that.

Responsive : To make stylish the site for every device we had to use the responsive design, And for that, there is two ways to use it in our situation.

FlexBox : it is only the fact of using the **'display: flex'** in *'div'* elements**,** In this way elements into the divs will automatically reajusted with the size of the screen. Form example for headers, when the size of the screen Shrunk, elements of the header will be closer.



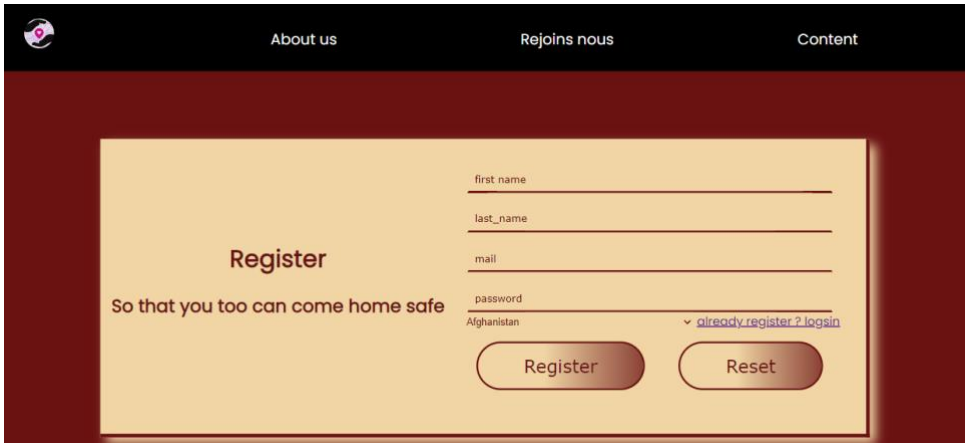
@media-querys: When flexBox is not enough, we must use what we call mediaQuerys. It is the simple fact of personalize the site as we want when the screen came to a particular size.



We can see above the registration form in a big screen and when we turn into a smaller screen and thanks to the code below,

```css
/* -------- Responsive --------- */

@media (max-height: 637px){
    .menu{
        display: flex;
    }
}
```

The registration form will become like that.



Issues and amelioration

In this Part we will talk about issues that we meet during our work, what solutions we bring and other ideas of amelioration.

Problems

We meet during lot of issues that we tried to solve. The first big difficultie that we find is the linking of database and the database. Even If link them is easy, the fact that find good requests in the good place in the code is preatty more difficult. One of the difficulites in Link with is to be careful with the syntax of requests. For example : in every request we must write the name of the table in those hooks (``).

```php
$rech = $user = "SELECT * from `tblstudents` where `email` = '$email' ";
```

An other big diificultie that we find is in link with submissions. Indeed when for example we want to modify the quantity of a product in our card we must refresh the page to see the difference in the page. The solution that we bring to this is to use this code after all submiting form.

```php
echo "<meta http-equiv='refresh' content='0'>";
```

An other big difficultie is with the chat part. Indeed when we how the chat was created, we start with the tbale creation :

	id	fromUsr	toUsr	message	sentdate
Éditer Copier Supprimer	1	1062	999	rue esquirol	2022-11-21
Éditer Copier Supprimer	2	1062	998	rue du géneral Leclerc	2022-11-21
Éditer Copier Supprimer	3	999	1062	rue Edith Piaf	2022-11-21
Éditer Copier Supprimer	4	1062	999	test	2022-11-22
Éditer Copier Supprimer	5	1062	999	test	2022-11-22
Éditer Copier Supprimer	6	1062	998	bonjour	2022-11-22
Éditer Copier Supprimer	7	1062	998	bonjour	2022-11-22
Éditer Copier Supprimer	8	999	1062	3asba	2022-11-22
Éditer Copier Supprimer	9	999	1062	3asba	2022-11-22

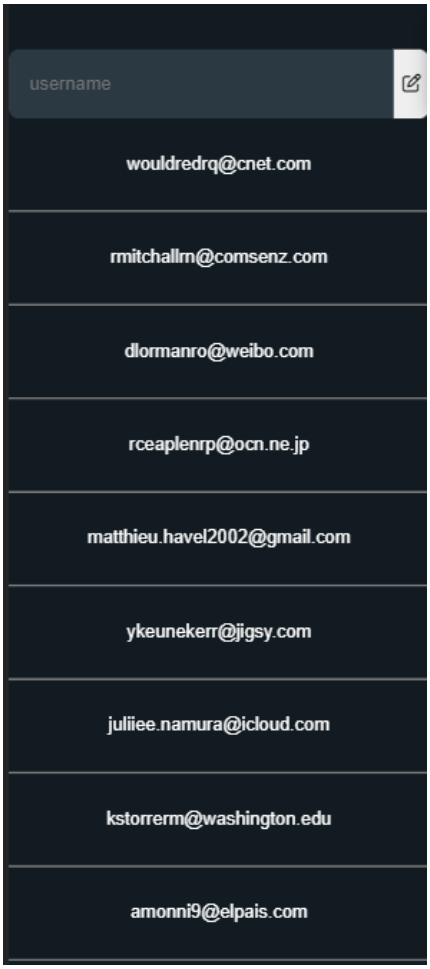
There is 5 columns in the table. The id of the message, to sender, the receiver, the message and the date.

'fromUsr' and *'toUsr'* are 2 foreign keys with the primary key of **'tblstudents'** that correspond of the ID of the user. The problem here is to create all input for every user who is chatting with us.

We have to create only one unique input for every user and the problem that is, as we can see on the table, the ID of a receiver and a sender could appear many times, and we also have to create input only for the current user. for this, with use a particular SQL request that helps us with all we need.

```
$msg = "SELECT DISTINCT `fromUsr`, `toUsr` from `chat` where `fromUsr` = '$from' or `toUsr` = '$from' ORDER BY `id` DESC";
```

the variable *'$from'* here correspond to the ID of the current user. At the end, we have this result like the website of whatsapp to see all user that we chat with.



The last big issue that we find is the sending of mails. One day it works, and the other day nothing is working. We have in our site four differents types of emails that we send.

New passport : When we go on the page **'forget password'** a mail is automatically sent with a temporary password that is generate randomly and its unique, in the mail of the user to modify his password.

Receipt of the command : When a user submit his command, a mail is sent with the details of his command and the total amount.

New price In an article. If a user have an article in his card and the admin just modify the price of the article, a mail is sent to the user to tell him that an article of his card gets discount.

The way of sending those mails is pretty similar, However there is a difference in sending the mail of discount prices because we have to send this mail to all users that have this article in their card. To do this, we must join the '**article'** table, the **'tblstudents'** table and the **'basket'** table.

```
`article` on basket.id_article = article.ID where article.ID = $id ";
```

After that, we fetch all querys of this request and we send the mail to every result of the fetch.

Improvement perspectives

Even if our site is for us quite complete, there is lot of ideas that we want to improve in our sites.

The most important one is the develop a map online where we can see every signals of the users. But for this, we have to buy the API key google Map which count 200$ per month that is to expensive for us.

We also want in the futur implement a system of protector. A user could choose his protectors and when he signal a new locations, a notification directly on the website is sent to all his protectors with the location of the user.

We finally want to improve Ajax development in our site, for example in the chat, when a user send us a message, we have to refresh the page to see the new message.

There is of course many more aspects of our site to improve but those ideas that we put are the most important according to us.

Conclusion

Pour conclure, nous sommes tous les trois satisfait de ce que nous avons accomplis. Nous avons réussi à créer un site Web fonctionnel, avec un design propre et épuré. Nous avons pu implémenter toutes les fonctionnalités que nous voulions. Grâce à ce projet nous avons pu mettre en pratique tout ce que nous avons appris en cours tout en mettant le doigt sur un problème qui persiste depuis des années dans notre société. Nous sommes fiers du résultat final de notre site Web et si nous pouvons améliorer SafeWay, nous pourrons développer l'application SafeWay pour que ce soit plus pratique et maximiser le confort de l'utilisateur.
Nous tenons à remercier Monsieur DANIEL MAGO VISTRO de nous avoir accompagné durant ce projet.