

Laura Scognamiglio

Titre: développeur web et web mobile

26 juin 2022

Dossier De Projet Professionnel

Technology Based Magic



Sommaire

[Page 4 Compétences du référentiel couvertes par le projet](#)

[Page 5 Résumé](#)

[Page 6 Spécificités du projet](#)

- ✿ Description de l'existant et des nouveautés
- ✿ Périmètre du projet
- ✿ Public cible
- ✿ Arborescence

[Page 7 Organisation de travail](#)

[Page 8 Conception](#)

- ✿ Roadmap
- ✿ Maquettes Figma
- ✿ Modèle Merise
- ✿ Pistes envisagées et documentation.
- ✿ Git
- ✿ Charte graphique
- ✿ Typographies
- ✿ Convention de nommage
- ✿ Veille média

[Page 13 Fonctionnalités](#)

- ✿ Responsive
- ✿ Javascript
- ✿ Module de personnalisation
- ✿ Bootstrap
- ✿ Back-office
- ✿ Panel admin
- ✿ Gestion utilisateurs

- ⦿ Gestion articles
- ⦿ Gestion produits
- ⦿ Catégories
- ⦿ Couleurs et matériaux
- ⦿ Robots
- ⦿ Devis
- ⦿ Module de connexion
- ⦿ Espace Utilisateur
- ⦿ Recherche en anglais

[Page 24 Conclusion](#)

Compétences du référentiel couvertes par le projet

Le projet couvre les compétences énoncées ci-dessous.

Pour l'activité 1, “Développer la partie front-end d'une application web et web mobile en intégrant les recommandations de sécurité”:

- Maquetter une application
- Réaliser une interface utilisateur web ou mobile statique et adaptable
- Développer une interface utilisateur web dynamique
- Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce

Pour l'activité 2, “Développer la partie back-end d'une application web ou web mobile en intégrant les recommandations de sécurité.”:

- Créer une base de données
- Développer les composants d'accès aux données
- Développer la partie back-end d'une application web ou web mobile
- Élaborer et mettre en œuvre des composants dans une application de gestion de contenu ou e-commerce.

Résumé

Ce projet est réalisé en groupe. Nous avons rencontré Jean-Marc lors du hackathon réalisé à la plateforme en mars puis Claire par la suite. Technology Based Magic est une entreprise aixoise réalisant des dispositifs personnalisables et interactifs. Les supports sont des animatroniques, écrans tactiles, hologrammes, 3D, vidéos et la production de consoles interactives. Il s'avère que nous avons remporté le prix spécial du jury lors de ce hackathon. C'est donc tout naturellement que nous avons proposé à TBM de continuer le projet en projet professionnel. Etant donné que l'interactivité est une modalité centrale pour l'entreprise nous avons voulu réaliser un site bien plus dynamique que l'existant (qui est un CMS) où l'utilisateur sera directement invité à personnaliser un produit. Pourquoi ce choix?

- ✿ La cible de TBM sont des entrepreneurs, des équipes marketings désireuses de faire du branding ou désirant améliorer leur expérience utilisateur. Il nous a paru indispensable que les futurs utilisateurs puissent directement vivre cette expérience.
- ✿ De ce fait découle l'envie de proposer aux utilisateurs un espace personnel où il pourra voir tous ces produits qu'il a lui même personnalisé.
- ✿ Cet espace permet également de gérer des devis et ainsi de donner une meilleure visibilité à TBM et directement agrandir leur portée d'action.

Les consignes de Jean-Marc et Claire pour la réalisation sont les suivantes:

- ✿ Pouvoir facilement changer les produits et articles en tant qu'admin.
- ✿ Le site doit être « fun », (gaming, interactivité et expérience utilisateur)
- ✿ Design élégant

Pour le reste nous avions carte blanche. Nous avons donc fait une refonte du site où l'espace admin permet de contrôler, les utilisateurs, les produits et les articles. Une page du site est un module de personnalisation des produits et permet la génération d'un devis. Le design et l'un sont aussi deux composantes majeures du site.

Spécifications du projet

✿ Description de l'existant et des nouveautés

Le projet est une refonte du site web existant qui est un CMS, avec un ajout de panels admin, un module de personnalisation de produits, génération de devis et création d'un espace user. La charte graphique est fourni par l'entreprise. L'encodage utilise VS code ainsi que les langages suivant:

PHP / MySQL

Javascript

HTML/CSS

✿ Périmètre du projet

Le site sera réalisé en français et ce dernier devra être accessible sur différents supports, à savoir mobile et ordinateur. L'encodage, le roadmap et les modèles Merise sont en anglais. Claire et Jean-Marc étant pleinement bilingues nous avons pu leur partager ces éléments sans aucun problème.

✿ Public cible

Comme dit dans le résumé, la cible de TBM sont des entrepreneurs, des équipes marketings désireuses de faire du branding

✿ Arborescence du site

page d'accueil, avec une section renvoyant à la galerie et une autre présentant les productions de TBM

page nous (à propos)

page galerie

page module de personnalisation

page d'inscription / connexion

Espace admin

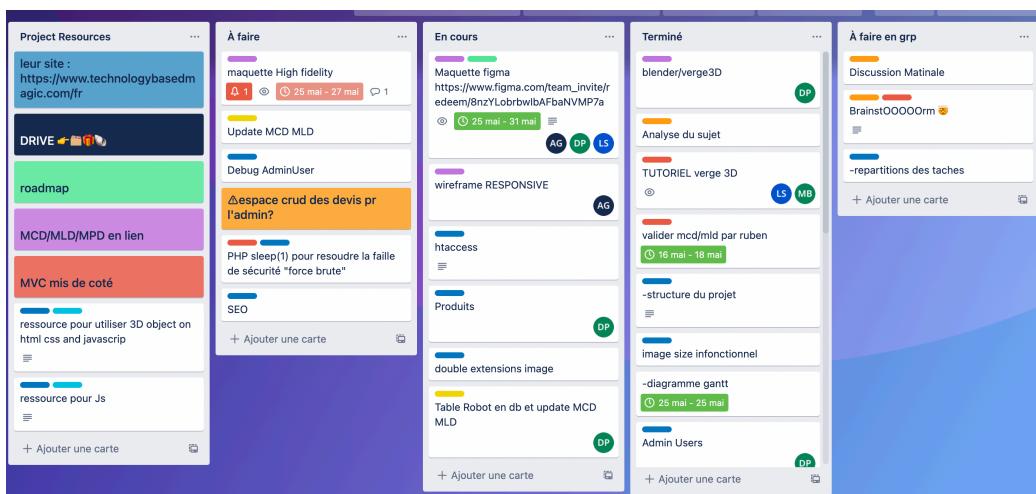
Espace utilisateur

Organisation de travail

Nous avons d'abord eu un rdv en présentiel à l'école avec Claire et Jean-Marc.

Nous avons élaborer un cahier des charges succinct car l'entreprise nous vraiment laissé carte blanche sur la conception et réalisation du projet. Nous avons évidemment tenu compte de leurs souhaits explicités en amont dans le résumé.

En vu de ce rdv un brainstorming de toutes nos idées a été réalisé en groupe, nous les avons soumis lors de cette rencontre. Après leur validation, nous avons créé un Trello pour une meilleure vue d'ensemble et répartition des taches. Nous avons intégré un principe du diagramme de Gantt à savoir le temps accordé pour chaque tache.



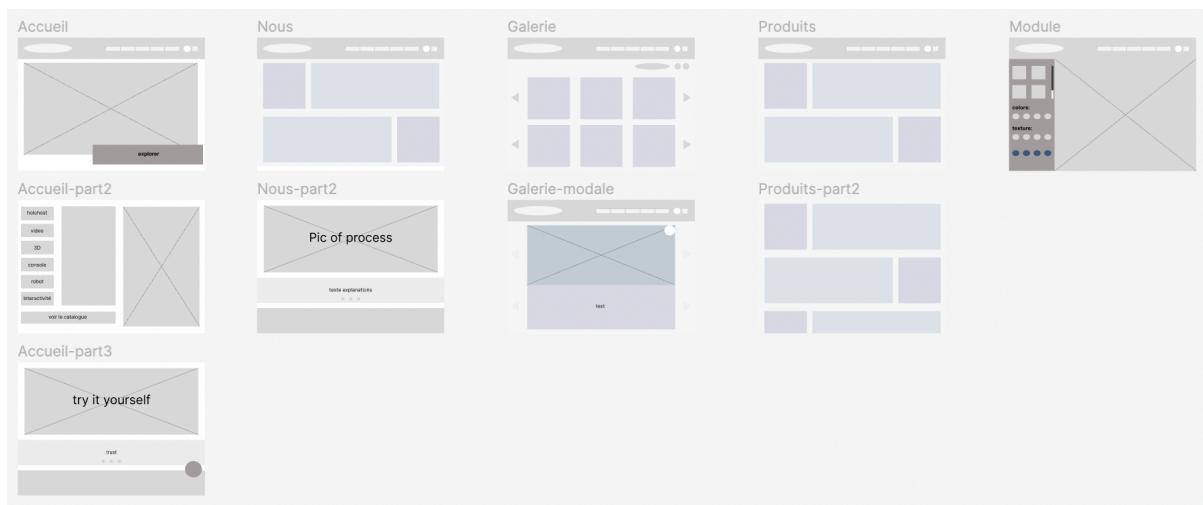
Conception

⚙️ Roadmap

Le roadmap est une représentation simplifiée du projet professionnel. Il nous a permis de mettre en évidence la complexité du projet qui tient à bien organiser les différents panels admin (CRUD articles, CRUD produits, CRUD utilisateur etc). Il a été également très utile pour les maquettes.

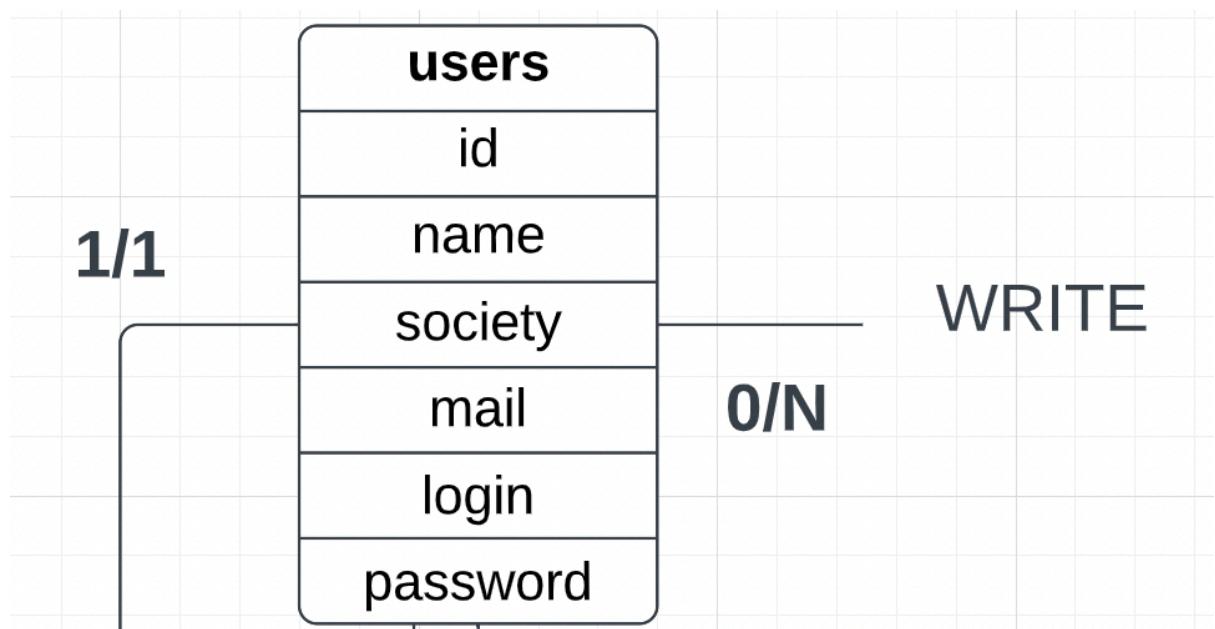
⚙️ Maquettes

Comme dit précédemment il a été nécessaire de maquetter le site dans son entièreté. Mais aussi l'espace admin en lui même qui contient plusieurs panels admin. Nous avons utilisé Figma les premières maquettes ont été des wireframes qui ont permis de situer les différents éléments au sein des pages et aussi de pouvoir vraiment penser l'UX. Ensuite une maquette high Fidelity utilisant la charte graphique fournis par TBM a été réalisée.



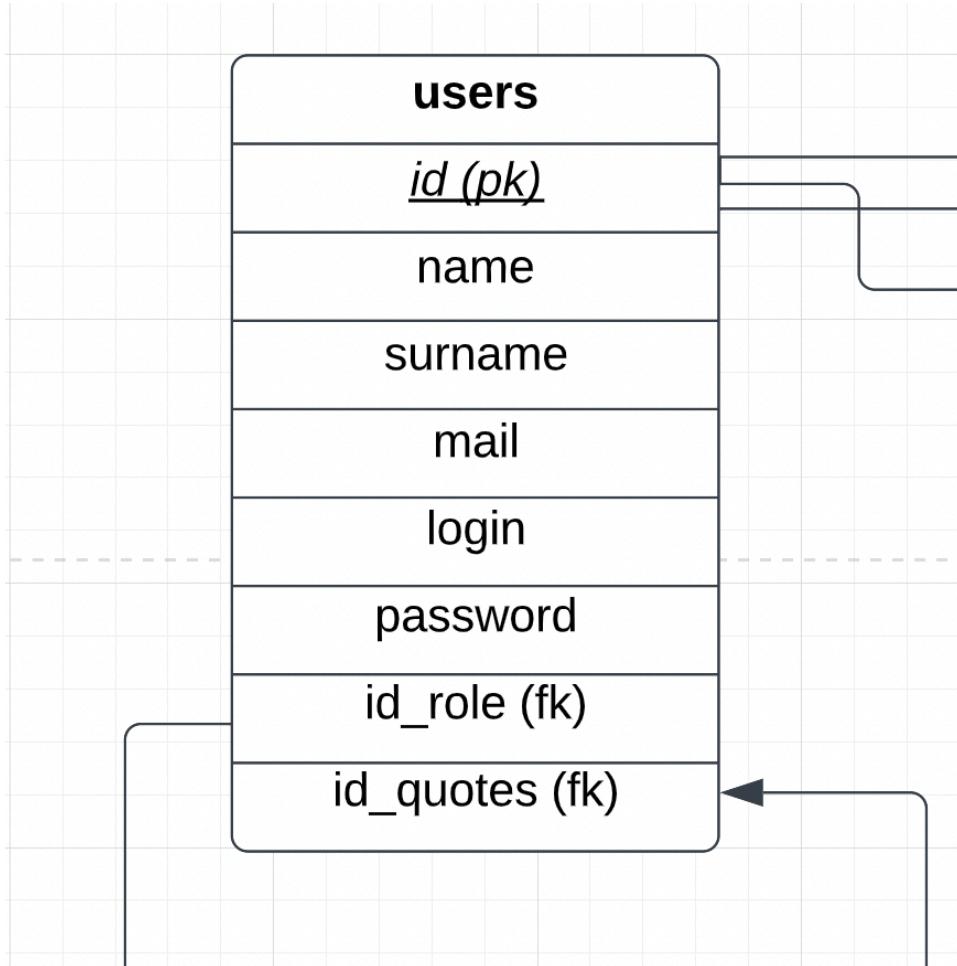
❖ Merise

Un temps fort de la conception est lorsque nous avons crée les modèles Merise. Nous avons d'abord réalisé le MCD qui est le modèle conceptuel de données. Ecrit en anglais il a été simple de le partager et de l'expliquer à TBM. La relation est verbale à l'infinitif et il utilise des cardinalités qui sont les minimums et maximums. Il en existe 4: 0 à un, 0 à plusieurs, 1 à plusieurs, 1 à 1.



MCD

Par la suite nous avons élaboré le MLD modèle logique des données. Il nous a permis de rajouter des tables de liaisons, les clés secondaires, il est voué à changer, c'est une trame, le MLD ne concerne que les clés primaires et secondaires, il n'y a pas de cardinalités mais des relations fléchées.



MLD

Pour finir nous avons réalisé le MPD, car le projet est finalisé. Les cardinalités sont en pattes d'oies et le typage des clés ainsi que leur spécificité (primaire ou secondaire) sont nécessaire.

❖ Pistes envisagées et documentation.

En amont du projet nous avions envisagé d'utiliser un frameworks PHP, Slim et ainsi une structure MVC (Model, View, Controller). Ces pistes ont été mises de côté pour les bonnes performances du groupe étant donné nos disparités, qui sont d'ailleurs plus des atouts que des tares.

Les recherches durant le projet ont été rédigées en anglais, plusieurs de moteurs de recherches ont été utilisés, tel que Google, You.com ou DuckDuckGo. Des plateformes tels que Stackoverflow ont été très utiles pour résoudre les problèmes rencontrés. Le dialogue entre membres de l'équipe et d'autres camarades en plus des conseils de nos formateurs ont été très précieux.

❖ Git

Pour mener à bien le projet, nous avons convenu de réaliser une branche par fonctionnalité. Toute l'équipe utilise le terminal pour le versioning du projet.

❖ Charte graphique



C 100 %	# 111f32
M 83 %	R 17
J 49 %	V 31
N 63 %	B 50

C 76 %	# 149ad7
M 23 %	R 20
J 0 %	V 154
N 0 %	B 215

C 24 %	# bc9b67
M 35 %	R 188
J 62 %	V 155
N 11 %	B 103

⌚ Typographies:

NEUE HAAS UNICA W1G pour les titres.

FIRA pour les sous titres.

AVENIR pour tous les corps de textes.

⌚ Convention de nommage:

PascalCase: pour les classes PHP,

camelCase: pour les méthodes,

kebab-case: pour les attributs de balise HTML.

⌚ Veille médias:

Une vielle média à été opéré sur OWASP et CNIL. Pour les idées Medium. Pour ce qui est des sources d'inspirations: **Dribbble**, **Awwwards** et **Behance**. Pour ce qui est des snippets le moteur de recherche [YOU.com](#) uniquement anglophone. Egalement pour le plaisir **daily.dev** en page d'accueil nous permet d'avoir les dernières actualités, nous pouvons les filtrées en fonction des technologies qui nous intéresse.

Fonctionnalités

⚙️ Responsive

Le projet a été réalisé en desktop first mais le responsive est une partie très importante et non négligeable. Pour les versions mobiles nous avons eu recours au media queries et aussi aux variables CSS

:root peut être utile lorsqu'on déclare des propriétés CSS personnalisées globales dans un document. Nous avons également utilisés d'autre pseudo classe comme **:hover**

⚙️ Javascript

Nous avons intégrer une librairie javascript du nom de Konva qui est récente et maintenue à jour régulièrement. Cela nous as permis d'utiliser le drag & drop ainsi que la redimension d'image. L'installateur de paquet NPM nous permet de l'installer sur le projet mis nous avons préféré l'installer via CDN afin d'optimiser le chargement du site.

⚙️ Module de personnalisation

Animer via JS et sa librairie, le module permet à l'utilisateur de voir les différents modèles de têtes de robot, de corps de robot. L'utilisateur pourra également choisir des couleurs ou matières. Il pourra sauvegardé son modèle personnalisé mais aussi ajouter ses propres images. Et ainsi générer un devis et/ou sauvegardé sa création dans son espace personnel.

⚙ Bootstrap

Nous avons utilisé Bootstrap pour les panels admins. Là aussi nous l'avons l'installer via CDN afin d'optimiser le chargement du site. Le panel admin étant privé

⚙ Back office

Le gérant du site à accès à un espace administrateur, lui permettant de gérer son site et son contenu.

À l'aide de **PHP** et de **requêtes SQL**, je récupère l'ensemble des données que je souhaite retrouver dans l'espace administrateur. Ce qui permettra de les modifier ou de les supprimer.

Nous retrouvons pour ce projet, deux types d'administrateur:

Le super administrateur, celui-ci correspond au gestionnaire de la plateforme, il possède tous les droits.

L'administrateur détient les mêmes droits que le super administrateur cependant, il ne peut pas avoir accès à la gestion des administrateurs et n'a pas l'accès au devis. Seul le super administrateur peut gérer les administrateurs.

⚙️ Gestion des utilisateurs

Pour définir les droits d'un utilisateur, nous retrouvons en base de données une table titles. Nous avons défini dans la structure de notre table, que la **primary key** ne s'**auto incrémente** pas ce qui nous permet de pouvoir attribuer des rôles différents selon l'**ID**.

<input type="checkbox"/>	Éditer	Copier	Supprimer	1	user
<input type="checkbox"/>	Éditer	Copier	Supprimer	10	admin
<input type="checkbox"/>	Éditer	Copier	Supprimer	100	superAdmin

L'administrateur à la possibilité de gérer les utilisateurs c'est-à-dire modifier leurs informations: noms, mail, login... Ou supprimer l'utilisateur.

Il peut également modifier les droits d'un utilisateur et faire de lui un administrateur.

En sélectionnant à l'aide d'une **requête SQL**, tous les utilisateurs qui possèdent un **id_role 1**

```
$sql = "SELECT * FROM users WHERE id_role = 1
```

Avec un **Foreach** j'affiche les informations des utilisateurs récupérées au préalable.

```

<?php foreach ($users as $user) :

?>
<tr>
    <td><?= $user['name']; ?></td>
    <td><?= $user['surname']; ?></td>
    <td><?= $user['mail']; ?></td>
    <td><?= $user['login']; ?></td>
    <td><?= $user['id_role']; ?></td>
    <td><a href="adminUser.php?id=<?= $user['id'] ?>">User management</a></td>
</tr>
<?php
endforeach; ?>

```

À la fin du tableau affichant l'ensemble des informations d'un utilisateur, nous retrouvons un lien contenant l'**ID** de l'utilisateur, pour afficher le formulaire permettant de modifier les informations d'un utilisateur ou de supprimer l'utilisateur.

L'affichage du formulaire est généré à l'aide de la **variable \$_GET** et l'**ID**.

```

<?php
if (isset($_GET['id'])) {

    $singleUser = $userAdmin->getSingleUser($_GET['id']);
    $userAdmin->updateUser($_GET['id']); ?>

    <form action="" method="post">
        <input type="text" name="nameUser" value="<?= $singleUser['name']; ?>" placeholder="">
        <input type="text" name="surnameUser" value="<?= $singleUser['surname']; ?>">
        <input type="email" name="mailUser" value="<?= $singleUser['mail']; ?>">
        <input type="text" name="loginUser" value="<?= $singleUser['login']; ?>">
        <input type="text" name="id_role" value="<?= $singleUser['id_role']; ?>">
        <button type="submit" value="<?= $singleUser['id']; ?>" name="submitUser">Update User</button>
        <a class="a_admin" href="adminUser.php?delete=<?= $singleUser['id']; ?>">Delete User</a>
    </form>
<?php }
?>

```

Dans le cas où l'administrateur souhaiterait modifier un utilisateur et donc afficher le formulaire, j'utilise deux **méthodes**.

La première me permet de récupérer les informations d'un utilisateur unique et de les afficher dans la **value** des **inputs** de mon formulaire. Elle prend en paramètre **\$_GET['id']** qui correspond à l'**ID** de l'utilisateur sélectionné en affichant le formulaire à l'aide du lien.

```

$sql = "SELECT * FROM users WHERE id = ?";
$stmt = $this->pdo->prepare($sql);
$stmt->execute(array(
    $id));

```

La deuxième permet en cas de modification des informations de les **update** en **base de données**.

Après validation du formulaire, je définis des **variables** qui ont pour valeur les champs renseignés dans les **inputs** du formulaire à l'aide de la **superglobale \$_POST**. Il existe plusieurs superglobales et elles rendent accessibles les variables dans tout le script. Ce qui me permet de pouvoir les utiliser pour la **requête SQL** qui va modifier les informations de l'utilisateur.

```

public function updateUser()
{
    if (isset($_POST['submitUser'])) {

        $name = secuData($_POST['nameUser']);
        $surname = secuData($_POST['surnameUser']);
        $login = secuData($_POST['loginUser']);
        $mail = secuData($_POST['mailUser']);
        $id_role = secuData($_POST['id_role']);
        $idUser = secuData($_POST['submitUser']);

        $sql = "UPDATE users SET name = ?, surname = ?, mail = ?, login = ?, id_role = ? WHERE id = ?";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute([
            $name, $surname, $mail, $login, $id_role, $idUser
        ]);
    }
}

```

Le super administrateur possède la même interface utilisateur avec en plus la gestion des administrateurs. Il a la possibilité de modifier les informations d'un administrateur, de rétrograder son statut ou bien de supprimer son compte.

En utilisant la **superglobale** **\$_SESSION**, si l'**id_role** correspond à **100** (**super administrateur**), j'affiche dans un tableau les informations des administrateurs.

```
if ($_SESSION['id_role'] == 100) :  
    foreach ($getAdmin as $admin) :
```

⌚ Articles

L'administrateur peut créer, modifier et supprimer des articles.

Un article possède une image qui sera stockée dans un dossier **asset** et le nom de l'image sera ajouté en **base de données** dans la **table images**.

J'annonce à mon formulaire qu'il attend un fichier a **upload**.

```
<form action="" method="post" enctype="multipart/form-data">
```

À l'aide de la **superglobale** **\$_FILES**, je récupère les informations de l'image qui sera upload.

```
$tmpName = $_FILES['add-pic']['tmp_name'];  
$name = $_FILES['add-pic']['name'];  
$size = $_FILES['add-pic']['size'];  
$type = $_FILES['add-pic']['type'];  
$error = $_FILES['add-pic']['error'];
```

Une fois le nom de l'image récupéré, je souhaite définir les extensions qui sont acceptées pour l'ajout d'images.

Pour cela je vais tout d'abord définir une **variable** qui prend en valeur un ‘.‘ et la **variable** qui contient le nom de notre image récupéré avec la **variable** **\$_FILES** et qui est scinder avec la **fonction explode**. J'associe cette **variable** à une nouvelle

variable qui prend en paramètre la fonction, **strtolower** pour mettre la **string** en **minuscule** et la **fonction end** pour que ce soit en fin de tableau.

Je définis un **array** ou à l'intérieur nous retrouvons les extensions autorisées .

```
$picExtension = explode('. ', @$name);
$extension = strtolower(end($picExtension));
$extensionsAllowed = ['jpg','png','jpeg','gif'];
```

J'attribue à une nouvelle **variable**, le chemin où stocker l'image, ainsi que son nom et son extension. Et dans une autre je déclare la taille maximum pour l'ajout d'images.

```
$way = "/Applications/MAMP/htdocs/tbmpro/assets/img/".$namePic.'.'.$extension;
$maxSize = 400000;
```

Si on retrouve dans un tableau l'**extension** et qu'elle correspond bien à une **extension** autorisée, que la taille de l'image ne dépasse pas la taille maximum autorisée et qu'il n'y pas d'erreur.

```
if(in_array($extension, $extensionsAllowed)&& $size <= $maxSize && $error == 0)
```

Alors, je vérifie si l'image n'est pas déjà en **base de données**. Pour cela à l'aide d'une **requête SQL** je sélectionne toute ma **table** image où le nom correspond à ma **variable** contenant le **nom** et l'**extension** de l'image à **upload**. En utilisant un **RowCount** si le résultat est strictement égal à zéro, alors je peux **insert** en **base de données** l'image et l'article en associant l'**ID** de l'image à l'article.

```
$sql = "SELECT * FROM images WHERE name = ?";
```

❖ Catégories produit

L'administrateur est en mesure de créer, modifier et supprimer des catégories de produit.

❖ Couleur et matériaux

Les couleurs et les matériaux sont des sous-catégories qui vont nous servir pour trier les éléments dont on aura besoin pour la confection de robots. L'administrateur peut ajouter, modifier ou supprimer des couleurs ou matériaux.

❖ Robots

L'administrateur peut créer des robots, le robot est séparé en deux composants qui sont la tête et le corps sous forme d'image. Chaque composant possède une couleur et une matière, je stocke chacun d'entre eux en **base de données**.

Dans une **balise select option**, j'affiche les couleurs et les matériaux à l'aide d'un **foreach**. Ce qui me permet de sélectionner les composants selon les éléments désirés.

Je les affiche dans des **inputs type checkbox** et place dans la **value** l'**ID** des composants, ce qui me permet d'**insert** en **base de données** le robot si le nom n'est pas déjà pris. L'administrateur peut aussi modifier et supprimer les robots.

❖ Devis

Nous avons décidé de mettre en place un espace regroupant les devis qui sera accessible, uniquement par le **super administrateur**.

Chaque devis est affiché avec l'ensemble des informations de l'utilisateur en utilisant un **INNER JOIN** et un **foreach**.

❖ Module de connexion

Inscription

J'ai mis en place une fonction **générique** pour sécuriser les **\$_POST** que je vais récupérer. Elle permet de se protéger contre les **injections SQL** et les **failles XSS**.

```
function secuData($data)
{
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    $data = htmlentities($data);
    return $data;
}
```

```
$login = secuData($_POST['login_singUp']);
```

Si le formulaire est validé, je vérifie si l'utilisateur n'existe pas en **base de données** avec un **RowCount**. Je vérifie si le mail renseigné est bien un mail avec la **fonction PHP filter_var (filter_validate_email)**.

```
(filter_var($mail, FILTER_VALIDATE_EMAIL))
```

Si le mot de passe est le même que la confirmation du mot de passe et qu'il est supérieur ou égal à huit caractères. Je **hash** le password avec **password_bcrypt**.

```
$hashed_password = password_hash($password, PASSWORD_BCRYPT);
```

Et j'**insert** en **base de données**. Si il y'a pas eu d'erreur lors de l'inscription l'utilisateur est redirigé vers la page de connexion à l'aide d'un **header location**.

Une fois les champs remplis dans le formulaire de connexion, par rapport au **login** renseigné j'utilise un **RowCount**, si le résultat est strictement égale à un alors à l'aide de **fetch** je récupère les données. Si les informations en **\$_SESSION** correspondent au résultat de mon **fetch**, alors l'utilisateur est connecté.

```
if (password_verify($password, $info['password']))  
    $_SESSION['id'] = $info['id'];  
    $_SESSION['name'] = $info['name'];  
    $_SESSION['surname'] = $info['surname'];  
    $_SESSION['mail'] = $info['mail'];  
    $_SESSION['login'] = $info['login'];  
    $_SESSION['id_role'] = $info['id_role'];
```

❸ Espace utilisateur

Si l'utilisateur est connecté, il peut avoir accès à son espace utilisateur. Dans celui-ci il peut modifier ses données personnelles, supprimé son compte.

Il retrouve aussi ses devis qu'il peut télécharger.

❸ Recherche anglophone

Pour le module de personnalisation de robot, nous avons décidé d'utiliser Konva et nous devions nous documenter sur le drag and drop, ce pourquoi nous avons fait nos recherches sur Stack overflow.

I want to be able to drag and drop a group of objects (rect, text, circle, etc.) and get their new x,y position in order to keep them in that new position.

I've checked the documentation for `Konva.js` for [dragging a group](#), but with no luck.

Je veux pouvoir faire glisser et déposer un groupe d'objets (rectangle, texte, cercle, etc.) et obtenir leur nouvelle position x,y afin de les maintenir dans cette nouvelle position.

J'ai consulté la documentation de Konva.js pour faire glisser un groupe, mais sans succès.

First thing that stands out is you are using `mouseout` as the dropped according to the documentation there is a `dragend`, I think that is what we should be using on this case: https://konvajs.org/docs/events/Binding_Events.html

La première chose que l'on remarque, c'est que vous utilisez le mouseout pour le drop ; selon la documentation, il existe un dragend, je pense que c'est ce que nous devrions utiliser dans ce cas.

Conclusion

Le projet TBM étant un projet conséquent, il a nécessité une attention toute particulière. D'avoir mené ce projet à la victoire du hackathon avec mes camarades puis ensuite avec ses mêmes camarades de l'avoir reconduit en projet professionnel de fin d'année, nous rends extrêmement heureux et fières de notre parcours. Nous vous remercions pour votre attention.