

Rapport du projet de Développement Web:

Site de vente de T-Shirt avec créateur de motifs personnalisés

BERNARD Paul-Antoine, AIT ADDI Marwan, HUBERT Gustav

Avril 2020

Table des matières

1	Présentation du site	2
1.1	Acheter des t-shirt à motifs précréés	2
1.2	Créer ses propres motifs	2
1.3	Fonctions administratives	2
1.4	Accès et utilisation	2
2	Boutique	3
2.1	Creation de la Base de Données (Paul-Antoine)	3
2.2	Pages PHP (Paul-Antoine)	3
2.2.1	Accès Client (Paul-Antoine/Marwan)	3
2.2.2	Accès Admin (Paul-Antoine)	4
2.3	Recherche par mot-clef (Marwan)	4
2.4	Gestion des stocks de la boutique (Marwan/Gustav)	4
3	Créateur interactif	5
3.1	Sauvergarde et affichage des images utilisateur dans la base de donnée (Marwan)	5
3.2	Drag & Drop (Marwan/Gustav)	5
3.3	Mask (Gustav)	6
3.4	Redimensionnement et déletion (Gustav)	7
3.5	Couleurs (Gustav)	7
3.6	Sauvegarde (Gustav)	8
3.7	Génération (Gustav)	8
4	Schéma de la Base de Donnée	9

1 Présentation du site

1.1 Acheter des t-shirt à motifs précréés

La première partie de notre site consiste en une boutique classique de vente de t-shirt, on peut parcourir le catalogue librement ou bien effectuer une recherche spécifique. Tout le catalogue est accessible, que l'on soit un utilisateur connecté ou non.

1.2 Créer ses propres motifs

Les utilisateurs ayant choisis de s'inscrire sur le site ont accès à un créateur interactif de t-shirt. Ils peuvent changer les couleurs du T-Shirt et importer leurs propres images (nous n'en fournissons pas/ pas de recherche google directement sur le site pour respecter les droits d'auteurs) pour les placer sur le t-shirt.

1.3 Fonctions administratives

Les administrateurs ont accès à des outils de gestion tels que la suppression ou l'ajout de nouveaux t-shirt dans la boutique ainsi que la gestion des stocks de t-shirt. À l'aide du créateur, ils peuvent également ajouter leurs propres motifs personnalisés directement dans la boutique.

1.4 Accès et utilisation

Le site se trouve à l'adresse

`https://mira2.univ-st-etienne.fr/~hg08566t/Site/index.php`

Le compte `admin` a pour mot de passe `passee`.

2 Boutique

2.1 Creation de la Base de Données (Paul-Antoine)

Pour stocker tous les produits ainsi que leurs informations nous avons du passer par une base de données. La première étape a été de créer celle-ci par le biais de Phpmyadmin. Au départ il n'y avait que deux tables : Produits qui contient les champs type, description, etc et Quantité qui contient elle les champs taille, nombre restant. Puis a été rajoutée une table pour stocker les images réalisées dans le créateur interactif. La deuxième grosse modification concerne elle la manière de stocker les images dans la base de données. En effet nous avons commencé par créer un attribut "chemin" qui indiquait où aller chercher les images. Au final nous avons trouvé que stocker directement les images dans la base de données (au moyen d'un champ BLOB) était plus simple.

2.2 Pages PHP (Paul-Antoine)

Maintenant que nous avons notre base de données il fallait créer les pages php qui nous permettraient d'effectuer des requêtes. Nous avons donc notamment utilisé PDO pour cela. Il y a deux parties importantes à mettre en place pour gérer une boutique en ligne : l'accès au site par les clients et l'accès au site par les administrateurs.

2.2.1 Accès Client (Paul-Antoine/Marwan)

Nous avons écrit deux pages Php pour réaliser cette partie. Il faut tout d'abord afficher la boutique. En effet lorsque l'on arrive sur un site de vente, on voit l'ensemble des produits avec quelques informations. Dans notre cas, nous avons décidé d'afficher l'image d'un produit avec le nom et le prix de celui-ci. Le principe étant de réaliser une requête de type `SELECT *` puis avec un fetch de récupérer chaque tuple de la table dans une boucle et enfin d'afficher les attributs qui nous intéressent. Il nous restait ensuite à créer une page php qui afficherait des détails supplémentaires lorsque l'on clique sur une image. Pour cela, lors de l'affichage d'une image on crée aussi un lien vers une autre page php qui affichera plus d'informations. On passe par un `$_GET` pour transmettre l'identifiant du produit dans l'url et pour savoir pour quel produit de la base de données on doit apporter des précisions. Nous avons également ajouté une liste déroulante permettant de sélectionner l'une des tailles disponibles pour ce t-shirt, ces tailles sont stockées dans la table `Disponibilité`

2.2.2 Accès Admin (Paul-Antoine)

Comme pour l'accès client on aura un affichage de la boutique mais cette fois-ci tous les champs seront affichés. Une autre différence est que l'on a deux liens qui permettent pour chaque produit affiché de le supprimer ou de le modifier. On passe aussi par un `$_GET` pour récupérer l'identifiant du produit lors de l'activation du lien. Pour la suppression on crée une action delete qui sera une simple requête de suppression de tous les champs. Pour la modification en revanche on est redirigé vers une nouvelle page. Celle-ci est constituée d'un formulaire, on récupère les valeurs qui doivent être modifiées via un `$_POST`. Puis on effectue ensuite les requêtes de modifications seulement pour les champs que l'administrateur aura renseigné. Enfin il y'a une dernière possibilité qui est d'ajouter un produit dans la base de données. Pour cela on utilise un formulaire et au moyen d'un `$_POST` on récupère les nouvelles valeurs à insérer dans la base de données.

2.3 Recherche par mot-clef (Marwan)

Afin de permettre aux utilisateurs de trouver plus facilement un t-shirt qui leur correspond nous avons ajouté une fonction de recherche dans la boutique, le principe de fonctionnement est simple, on récupère les mots-clefs de la recherche avec une méthode post puis on fait une requête à notre base de données en utilisant `WHERE description LIKE "%mots-clefs%"` ce qui correspond à renvoyer tout les produits contenant les mots-clefs quelque part dans la description.

2.4 Gestion des stocks de la boutique (Marwan/Gustav)

Les stocks de la boutique sont représentés par la table `Disponibilité`. On y associe un id de T-Shirt, une taille et une quantité de produits disponible. La page de gestion des stocks, accessible aux administrateurs seulement, permet de visualiser ces données sous forme de tableau. L'accès se fait depuis la page de gestion des produits. À l'aide de cet outil, les administrateurs peuvent aisément supprimer ou ajouter des tailles et du stock de produits. Les valeurs sont additionnées lors d'un ajout de stock sur une taille existante.

3 Créateur interactif

3.1 Sauvergarde et affichage des images utilisateur dans la base de donnée (Marwan)

Les images que l'utilisateur a souhaité upload pour créer ses propres t-shirts sont stockées directement dans la table image sous forme de BLOB qui est un type de données qui stocke des binary files, afin de récupérer ce genre de données on utilise la fonction php `file_get_content()` qui renvoie une chaîne représentant l'image. On effectue ensuite un INSERT sur la table image avec le pseudo de l'utilisateur à qui correspond l'image et le blob de l'image préparé au préalable.

Ensuite afin d'afficher l'image on effectue une boucle sur les résultats d'une requête qui récupère toutes les images de l'utilisateur, à l'intérieur de cette boucle on souhaite afficher l'image dans une balise `` mais comment faire sans chemin vers l'image ?

Afin d'afficher l'image à partir du blob on écrit ceci dans le `src` :

```
"data:image/jpeg;charset=utf8;base64," . base64_encode($donnee['image'])
```

Expliquons chaque partie de cette chaîne de caractères :

- `"data:image/jpeg"` : Ceci explicite simplement sous quel format afficher l'image.
- `"charset=utf8"` : On spécifie ici l'encodage de la chaîne blob.
- `"base64," . base64_encode($donnee['image'])` : On spécifie que la chaîne est encodée en MIME base64 puis on utilise la commande `base64_encode()` pour encoder correctement les données de l'image. Je vous renvoie vers la page du manuel pour plus d'information sur la commande `base64_encode()` (<https://www.php.net/manual/fr/function.base64-encode.php>).

Ainsi notre image s'affiche comme d'habitude avec une balise `img`.

3.2 Drag & Drop (Marwan/Gustav)

Afin de pouvoir glisser et déposer nos images on va leur donner quelques attributs spéciaux et utiliser des fonctions javascript dont voici l'explication. Premièrement on va attribuer à nos images un attribut `ondragstart="drag(event)"` quand on va commencer à drag notre image on appelle la fonction javascript `drag` qui récupère l'événement et prépare un transfert de données avec `ev.dataTransfer.setData` de type `"text"` et on utilise l'id de l'événement pour savoir quelles données récupérer `ev.target.id`, ensuite on calcule les coordonnées de notre image par rapport à la page selon que l'on est dans le sélecteur d'images ou le créateur.

Ensuite nous ajoutons l'attribut `ondragover="allowDrop(event)"` pour lancer une fonction qui permet de donner le droit de drop une image a chaque fois que l'on drag une image avec `ev.preventDefault()` selon certaines conditions (notamment qu'il y a assez de place pour poser l'image. Enfin on fini par l'attribut `ondrop="drop(event)"` qui permet d'appeler la fonction `drop(event)` qui va ajouter l'image dans dans le créateur avec un `div.appendChild(img)` qui ajoute un nouveau "fils" donc du contenu de type image dans le créateur.

Lors du placement d'une image, au lieu de déplacer l'image existante, on utilise un clone de celle-ci. De cette façon on peut réutiliser la même image plusieurs fois. Il faut évidemment ajuster le système d'*id* pour éviter les confusions.

3.3 Mask (Gustav)

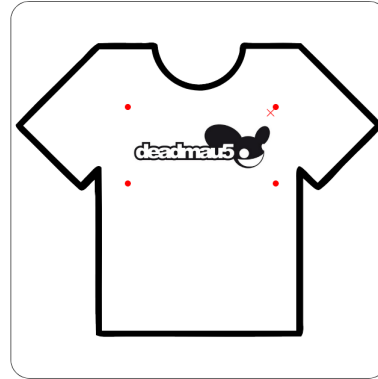
Pour donner une meilleure idée du produit fini, on ajoute au div créateur un modèle simplifié de t-shirt. Cela permet le visionnage de l'emplacement des images par rapport au t-shirt. Il est important que l'image SVG remplit entièrement le div, reste toujours en fond et ne peut pas être sélectionnée par l'utilisateur. En utilisant une image au format SVG (et donc non rastérisé), ce modèle ne perd pas en qualité lorsque la taille de la fenêtre augmente. De plus, étant transparente, on peut simplement changer la couleur du div pour changer la couleur du t-shirt.



Pour améliorer l'expérience on ajoute une deuxième image faisant office de masque pour cacher les bouts d'images dépassant du t-shirt. Il s'agit de l'inverse de la première image placé au même endroit et ayant les mêmes propriétés grâce au mot-clés *inherit*. Similairement, il faut toujours que cette image soit au dessus des autres. On s'en assure grâce à l'indice de profondeur *z-index*. Finalement on utilise *user-select: none* pour éviter que l'utilisateur puisse accidentellement sélectionner le masque.

3.4 Redimensionnement et délétion (Gustav)

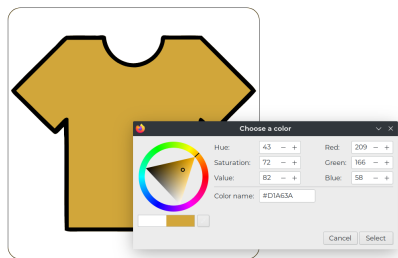
Si l'utilisateur peut changer l'emplacement des images qu'il utilise, il faut également lui permettre de les redimensionner. Ainsi, un clic sur une image fait apparaître des poignées aux coins permettant de l'étirer aux dimensions voulu. Les proportions originales sont respectées en utilisant la touche shift. Les poignées correspondent à quatre éléments *span* créés avec lors du chargement de la page. Au besoin on affiche ces quatre éléments puis on les déplace.



Les poignées étant affichées on passe au redimensionnement. Dès que l'utilisateur appuie sur un des coins, et jusqu'à ce qu'il relache, on utilise la position du curseur pour calculer la taille de l'image sélectionnée. Pour trois des poignées il faut également déplacer l'image. Finalement on vérifie que le curseur ne dépasse pas les limites du créateur ou de l'image elle-même.

À l'aide du même système on peut également faire apparaître un bouton permettant de supprimer l'image sélectionnée.

3.5 Couleurs (Gustav)



Pour changer la couleur du t-shirt on utilise une entrée de type *color*. Initialement la couleur du t-shirt ne changeait que lorsqu'on valide une couleur. Pour directement visualiser la nouvelle couleur sur le t-shirt on utilise un *eventListener*. Celui-ci permet d'appeler une fonction lorsqu'un élément, dans notre cas une entrée, change.

3.6 Sauvegarde (Gustav)

Lors de l'importation d'images il est nécessaire de recharger la page pour ensuite utiliser PHP. Cela marche bien mais a pour effet de supprimer toutes les images déjà placée sur le t-shirt. Pour remédier à cela il faut donc enregistrer tout changements. Malheureusement, accéder à la base de données à l'aide de JavaScript pure n'est pas possible. Reste alors l'utilisation de l'URL. En utilisant les chaînes de requêtes de l'URL on peut sauvegarder toutes les informations nécessaire pour recréer le T-Shirt. Il est important d'utiliser des pourcentages pour représenter les tailles et dimensions des images. Si la fenêtre change de taille on peut garder le même T-Shirt. Lors du chargement de la page il suffit alors de vérifier si les ID images et la couleur se trouvent dans l'URL (à l'aide de PHP) et ainsi recréer l'image.

```
/Site/creator.php?4[]=147+145+188+73&4[]=328+89+112+50&7[]=169+278+152+116&c=%23000000
```

3.7 Génération (Gustav)

Le T-Shirt créé à l'aide du créateur ne peut pas être utilisée tel quel. Lorsqu'il a fini, l'utilisateur avance donc à la page *genererTshirt.php*. Celle-ci récupère alors les informations nécessaires pour recréer l'image sur un *canvas* (de façon similaire à l'éditeur avec l'URL). A cette étape on utilise une vraie image de T-Shirt dont on modifie la couleur. Après avoir placé les images au bon endroit on peut générer une image de T-Shirt près à l'achat. S'il le souhaite, l'utilisateur peut revenir en arrière et modifier son T-Shirt. Au lieu d'acheter le T-Shirt créé, un administrateur à l'option de placer sa création dans la boutique.

4 Schéma de la Base de Donnée

