

TP développement WEB : réalisation d'un site de vente en ligne

JAEV VAVASSEUR – CLÉMENT MESNIL

Compte rendu
TP
Développement WEB

Jaël Vavasseur – Clément Mesnil

03/2020

Régis Jacquelin

Introduction

Le but de ce TP est de créer notre premier site WEB qui sera un site de vente en ligne sur lequel nous pourrions voir des listes d'articles, en ajouter à notre panier, les enlever etc... Nous utiliserons également une base de données Maria DB pour gérer les articles, les familles d'articles ou encore les paniers artificiels des clients. Enfin, le site devra être RWD c'est-à-dire que l'affichage du site WEB devra s'adapter à la taille de l'écran sur lequel il est affiché afin qu'il soit agréable d'utilisation sur ordinateur, tablette et smartphone.

Tout d’abord nous avons créé la page d’accueil. C’est une page qui devait afficher les familles, le panier du client, un endroit pour se connecter ainsi que les informations du vendeur selon le croquis suivant (cf Figure 1):

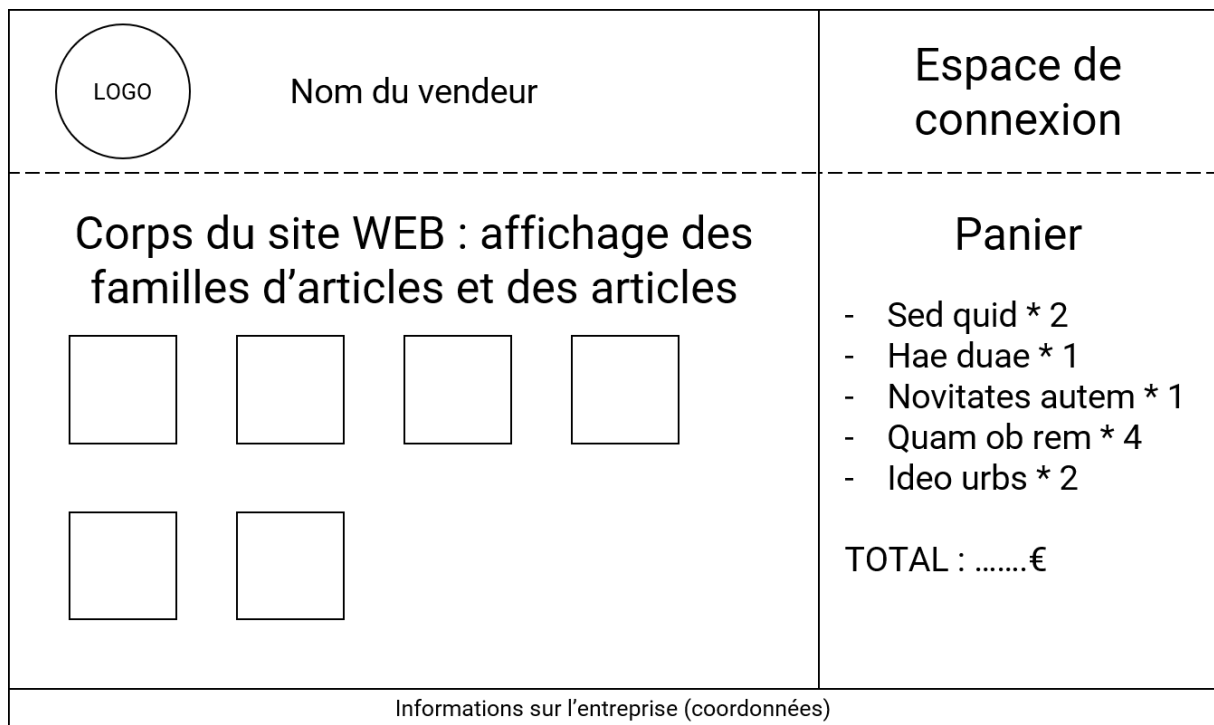


Figure 1 : Maquette de la page d'accueil

Nous avons donc disposé des `<div></div>` afin de créer les différents blocs que nous souhaitions avoir. Nous avons défini leur taille verticale en pixels et horizontale en pourcentage afin que la page s’adapte aux différentes tailles d’écran. De plus, afin que les blocs soient alignés comme sur le croquis (cf Figure 1) nous avons utilisé un fichier `style.css` ainsi que les attributs `float : left` et `display : inline-block`. Par la suite, nous avons défini les attributs plus « classiques » des différents div en leur donnant un id, comme leur `background-color`, la police utilisée, sa taille et sa couleur.

Pour l’affichage des familles d’articles, nous avons utilisé d’autres div. Sur ses images, nous avons utilisé des attributs (cf Figure 2) afin qu’elles soient assez espacées pour le plaisir des yeux.

```
.type_article img
{
    display: inline-block;
    text-align: center;
    margin: 5px;
    padding: 10px;
    width: 80px;
    height: 80px;
}
```

Figure 2 : Attributs utilisés sur les images des familles et articles

Avant de continuer à développer le site WEB, nous avons voulu que l’affichage de la fenêtre s’adapte selon les différentes tailles d’écran., c’est ce qu’on appelle le Responsable Web Design. Le but de responsive design est d’avoir un site, mais avec des éléments différents qui répondent différemment lorsqu’il est affiché sur les appareils de différentes tailles (PC/Mac, Tablette, Smartphone). Prenons un site web traditionnel « fixe ». Lors du visionnage sur un ordinateur de bureau, par exemple, le site Web peut afficher trois colonnes. Mais lorsque vous affichez la même mise en page sur une tablette plus petite, elle pourrait vous forcer à faire défiler horizontalement, quelque chose utilisateurs n’aiment pas ou des éléments pourraient être cachés à la vue ou avoir l’air déformé. De ce fait le RWD permettrait d’afficher les mêmes éléments mais sous différentes formes (cf Figure 3).



Figure 3 : Exemple d'utilisation du RWD

Pour ce faire, nous avons créé des sections pour chaque domaine de largeur d’écran que nous souhaitons personnaliser (exemple cf Figure 4).

```
@media screen and (max-width: 576px) {  
  #titre  
  {  
    width: 100%;  
    height: auto;  
    font-family: "Comic Sans MS";  
  }  
  #authentification  
  {  
    width: 100%;  
    height: 100px;  
    float: top;  
  }  
  #panier  
  {  
    width: 100%;  
    height: 100%;  
    float: bottom;  
  }  
  #contenu  
  {  
    width: 100%;  
  }  
}
```

Figure 4 : Exemple de section pour RWD

Voilà le rendu (une fois le site terminé) de l'affichage en mode grand écran et petit écran (cf Figures 5 et 6) :



Le leader du
modélisme en
ligne

Adresse email

Mot de passe

[Se connecter](#) [Créer un compte](#)



Kits maquettes



Planches en balsa



Planches en bois



Colles



Tubes métalliques



Votre panier

Colle Hart	1 x 9.00 = 9 €
Colle Stypor	2 x 5.00 = 10 €
TOTAL : 19 €	

[Vider panier](#) [Commander](#)

TOPModellisme.com est enregistré au R.C.S. sous le numéro
1234567890
13 avenue du Pré de la Reine

Figure 5 : Site en petit écran (RWD)



Figure 6 : Site en grand écran (RWD)

Par la suite nous avons créé la page d’affichage des articles. Elle devait contenir tous les articles de la famille sur laquelle on cliquait sur la page d’accueil. En soi, cette nouvelle page n’est en rien différente de la page d’accueil. Nous avons donc repris cette page d’accueil pour l’affichage des articles mais avons modifié les div des objets affichés de manière qu’on puisse y insérer l’image, le prix et la description de l’article.

Pour accéder à cette nouvelle page propre à chaque famille, nous avons mis une balise `<a href>` sur le nom de la famille. De ce fait, lorsque nous cliquons sur le nom de la famille, la fenêtre présentant les articles de la famille s’affiche.

Enfin, nous avons mis en page le panier, l’espace de connexion et d’inscription avec un formulaire et des boutons. Il ne manquait alors qu’un lien avec la base de données fournie afin d’afficher de réelles données.

Pour ce faire, nous avons créé un `index.php` qui appelle une fonction `connect_bd($db)` pour se connecter à la base de données. La suite de ce fichier PHP permettait d'afficher ce que nous avons présenté plus tôt dans ce rapport grâce au fichier `body.php`, puis, nous nous déconnectons de la base de données. Nous avons également créé la fonction `Afficher_familles($db)` pour afficher sur la page d'accueil toutes les familles, leur nom et leur image grâce à des requêtes SQL. `Afficher_items($GET['famille'], $db)` a également été utilisée pour afficher les articles, leur nom, leur image et leur prix sur la page d'une famille (cf Figure 7).

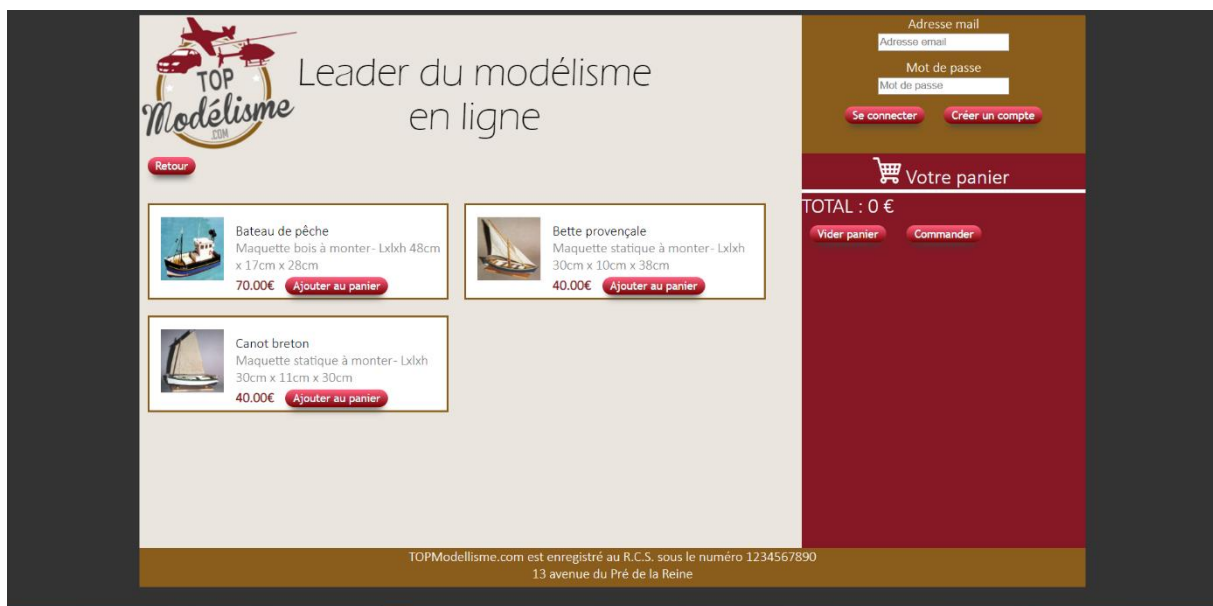


Figure 7 : Affichage des articles d'une famille

Enfin, nous avons eu besoin de 3 fonctions pour gérer le panier :

- `Ajouter_article_panier($article, $db)` qui, lorsque pour un article on clique sur le bouton « Ajouter au panier », ajoute l'article à notre panier. Si l'article est déjà présent dans le panier, cela incrémente seulement la quantité et met à jour le prix total.
- `Afficher_panier($db)` permet d'afficher la liste des articles présents dans le panier, leur quantité, leur prix et le prix total de la commande.
- `Vider_panier($db)` permet de vider la table panier afin de le vider, simplement.

Nous avons fait face à un problème : le panier pouvait être trop grand et le nombre d'articles affichés sur une page de famille également. De ce fait, les <div> sortaient de leur conteneur. Nous avons donc mis en place un **overflow** afin de faire défiler le panier, les familles et les articles.



Figure 8 : Aperçu du site WEB terminé et de l'utilisation du panier

Conclusion

Nous avons géré ce projet en séparant les tâches afin d'être plus efficace dans notre travail et en utilisant GitHub pour le versionning, ce qui nous a grandement simplifié la tâche pour pouvoir travailler depuis n'importe où. La plupart des difficultés venaient du fait que nous ne connaissions pas du tout le PHP avant et l'intégration du SQL directement dans le PHP, mais après quelques recherches sur des forums, tout s'éclaircit finalement et on s'aperçoit que c'était très simple. La plupart de nos difficultés restent le positionnement des div en css, qui est souvent capricieux, mais on arrive toujours à faire ce qu'on veut avec un peu de persévérance. A part ces petites difficultés, nous n'avons pas rencontré de difficultés majeures, souvent une simple lecture de la documentation ou une recherche Google nous a suffi. Le projet nous a apporté des connaissances et une bonne compréhension dans les langages PHP et Html, ainsi qu'en Css, nous connaissions déjà le langage SQL (MariaDB qui est très proche de MySQL que nous avons étudié). L'utilisation de GitHub a aussi permis de comprendre mieux le principe du versionning et sa grande utilité.