**Rapport pour le projet du site web**

Équipe :

Benjamin Rose : 6156635

Othmane Marzougui : 6155414

Fonctionnement

1. Partie exigence générales :

Nous avons réussi à implémenter le modèle MVC et la programmation orienté objet au meilleur de nos connaissances. Dans le modèle MVC nous avons aussi de l’héritage. Les classes PHP CommentManager, ContactManager, LoginManager, ProductManager, RegisterManager, SiteManager héritent toutes de la classe abstraite Manager. Pour le gabarit tous les pages ont le même template et on réutilise le style de la plupart des éléments dans tous les pages. Pour le pied de page le nombre de visiteurs augmente et s’insère dans la BD à chaque refresh de la page et à chaque fois que le fichier est ouvert dans le browser. On insère la date du jour dans la BD avec la fonction NOW (). On obtient la date de la dernière modification en utilisant la function filemtime() à laquelle on lui passe le url de chacune des pages. Pour que ça fonctionne on vérifie si le fichier existe en premier. Si il existe on fait un clearstatcache(), on set la timezone avec date\_default\_timezone\_set() ensuite on s’assure que la date est dans le bon format avant de faire le insert. Nous avons une function qui retourne le sum() de tous les visiteurs de toutes les pages de la base de données et une fonction qui retourne tous les autres infos. La date de la dernière modification est le dernier datetime que le fichier a été ouvert.

Pour la base de données nous avons créés 6 tables : [Structure](http://localhost/phpmyadmin/tbl_structure.php?server=1&db=pepiniere&table=commentaires)commentaires, messages, produits, rating\_info, utilisateurs et visiteurs. Nous avons utilisé PHPstorm ainsi que WAMP.

Nous avons créé nous-même le logo de la pépinière avec Photoshop et utilisé un favicon generator (<https://favicon.io/>) pour créer le favicon de notre site.

2. Page Accueil :

Notre page index est également notre page Accueil. Nous avons réussi à faire un formulaire de login qui vérifie si le nom d’utilisateur OU le mot de passe correspondent avec les informations contenues dans la base de données. Cet if sert surtout à vérifier que le nom d’utilisateur correspond à celui dans la base de données. Nos mots de passe sont hachés donc si la méthode password\_verify() retourne false nous retournons un message d’erreur pour dire que le mot de passe ne correspond pas. À côté du bouton submit il y a un bouton qui permet de créer un compte et qui envoie sur la page Inscription. Sur cette page le formulaire permet de créer un nouveau compte ou d’annuler et revenir sur la page Accueil. Lorsque l’utilisateur clique sur le bouton submit le compte est créé dans la base de données et l’utilisateur est ramené sur la page Accueil pour qu’il puisse se loguer. Si la connexion se fait avec succès l’utilisateur est redirigé vers la page Produits et a accès à toutes les pages à partir du menu de navigation sauf Accueil et Inscription. Lorsqu’il n’est pas connecté il a uniquement accès à la page Accueil et Contact. L’utilisateur peut changer le thème du site en tout temps.

3. Page Produits :

Nous avons fait une page qui affiche tous les produits par tranche de 3 produits en ordre alphabétique. Lorsque l’on clique sur le bouton Voir plus d’un produit il y aura une page-détail qui affichera toutes les informations qui se trouvent dans la base de données sur ce produit avec son id qu’on envoie par la méthode GET. Il affichera la catégorie du produit, l’image du produit, une description, la quantité disponible et le prix. Pour les images nous avons enregistré les paths dans la base de données.

4. Page Contact :

Dans la page contact on affiche un div contenant l’adresse et le téléphone du Collège Rosemont, un formulaire qui enregistre dans la base de données le nom, courriel, l’objet du message, le message et un div avec l’adresse du collège épinglée sur la carte Google Map. On affiche un message de confirmation pour dire que le message a été envoyé avec succès. Pour afficher la carte Google Map sans overlay il faut payer maintenant alors nous avons pris la version gratuite qui affiche un message d’erreur (This page can't load Google Maps correctly).

5. Page Livre d’or :

Dans cette page l’utilisateur connecté peux envoyer un commentaire. Lorsqu’il envoie un commentaire son nom d’utilisateur et la date sera affiché ainsi que son commentaire. Il aura l’option de like ou dislike le commentaire de lui-même ou de quelqu’un d’autre. S’il like un commentaire il pourra recliquer sur l’icône pour annuler son like, même chose s’il dislike un commentaire. S’il like un commentaire et veut le dislike il doit annuler son like en premier et ensuite il pourra cliquer sur dislike. Chaque utilisateur peut like ou dislike un commentaire une seule fois et on affiche le nombre total de like ou dislike de chaque commentaire qui sera évidemment enregistré dans la base de données. Le like/dislike système a été très difficile à implémenter et un de nos plus grand défis. J’ai trouvé personnellement que ton énoncé (En bas de chaque message, on demande si le commentaire est pertinent et on peut cliquer sur + ou et - et le total des points est affiché.) manquait de précision. Alors j’ai décidé de faire un like/dislike système qui a été extrêmement difficile pour moi. J’espère que tu vas le prendre en considération. De plus, l’utilisateur connecté a l’option de modifier ses propres commentaires mais ne peut pas modifier les commentaires des autres. Le lien pour modifier ne s’affichera pas pour le commentaire d’un autre utilisateur. En cliquant sur le lien, l’utilisateur est amené sur la page Modifier commentaire sur laquelle on a mis un formulaire qui affiche son commentaire et permet de le modifier. En cliquant sur le bouton Modifier l’enregistrement est mis à jour dans la base de données et l’utilisateur est redirigé vers la page Livre d’or.

6. HTML :

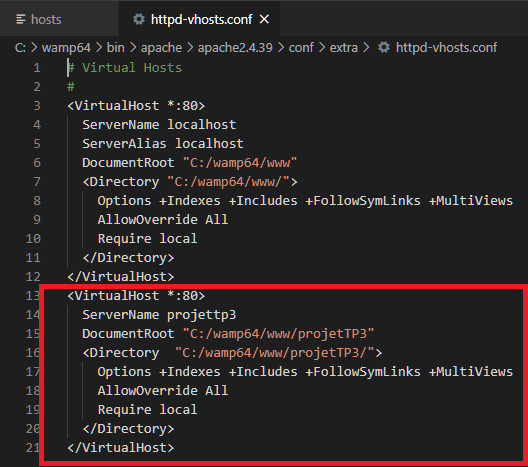
Pour la partie HTML nous avons respecté les normes XHTML. Nous avons utilisé la fonctionnalité Reformat code dans PHPStorm pour s’assurer qu’on a bien indenté notre code. Dans notre HTML nous avons utilisé des div pour bien séparer le code en sections. Toutes nos pages ont des balises header et footer ainsi qu’un div pour le contenu.

7. Page Panier :

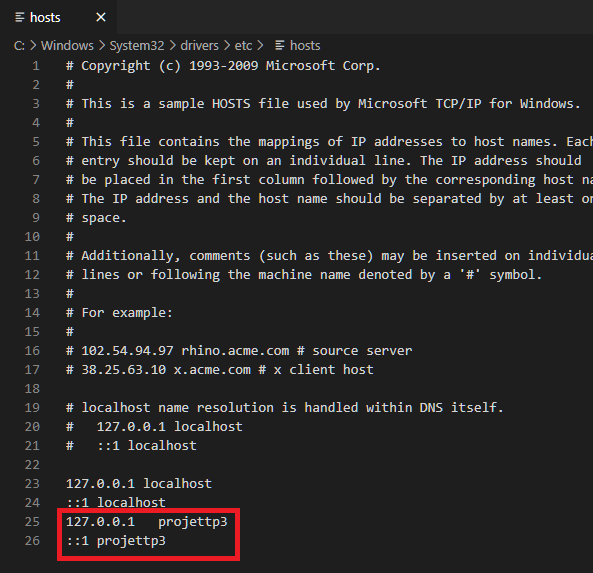
Chaque produit a un bouton Ajouter au panier sur sa page produit mais vu que tu as enlevé la page Panier des exigences on a décidé de le laisser comme ça. Donc cette fonctionnalité n’est pas implémentée.

8. WAMP

Pour répliquer le plus possible le fonctionnement d’un vrai site web j’ai décidé de me créer un virtual host à la place d’utiliser localhost pour mon projet. Donc j’ai modifié C:\wamp64\bin\apache\apache2.4.39\conf\extra\httpd-vhosts.conf



et C:\Windows\System32\drivers\etc\hosts



J’ai suivi les étapes sur <https://stackoverflow.com/questions/23665064/project-links-do-not-work-on-wamp-server/23990618#23990618> à partir de **How to create a Virtual Host in WampServer.**

Description du CSS et du Javascript

1. CSS

Toutes les pages utilisent le fichier style.css. Nous avons deux feuilles de style différentes pour les thèmes. Un pour le thème foncé avec des lettres pales et un fond foncé et un pour le thème pâle avec des lettres plus foncés et un fond pâle. Nous avons un lien dans la barre de navigation qui permet de toggle facilement le thème du site sur toutes les pages. Le contraste entre les couleurs rend nos formulaires plus beaux et plus lisibles. Pour le background du div qui contient le contenu de chaque page nous avons fait une animation de type slider dans le CSS qui alterne entre 3 images de background différentes avec des keyframes. Quand on hover sur les champs d’un formulaire on change la couleur de fond et quand on hover sur les items du menu la même chose mais on change aussi la couleur du texte. Quand on hover sur un lien autre que dans le menu de navigation ou un bouton on affiche un border-bottom de 3 pixels et on change de couleur. J’ai deux styles de boutons de couleurs différentes selon le thème choisi. Un bouton foncé et un bouton un peu transparent.

2. Choix des couleurs

Nous avons opté pour des tons de vert. Nous avons utilisé <https://javisperez.github.io/tailwindcolorshades/#/> pour créer des palettes des couleurs de vert qu’on voulait utiliser pour s’en inspirer.

Pour le thème foncé notre couleur principale ça été #008000 (Forest green). Ça a généré cette palette mais finalement nous n’avons pas utilisé les autres couleurs parce qu’on voulait avoir aussi un vert un peu plus pâle qui flash plus #00a000 (Dark lime green) comme deuxième couleur pour notre custom scroll bar et pour les hover du menu et ce vert pâle #eafbcb (Snow flurry) pour faire un contraste pour le texte. On utilise un border #000000 (Black) pour el div contenu. On a aussi utilisé des couleurs rgba pour faire des overlays sur l’animation des images de background, pour la couleur de background des formulaires et de certains divs. Pour le theme dark on a utilisé rgba(0, 0, 0, 0.2), rgba(0, 0, 0, 0.4), rgba(0, 0, 0, 0.5), rgba(255, 255, 255, 0.2), rgba(255, 255, 255, 0.4) ou rgba(255, 255, 255, 0.5).

Palette dark :





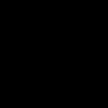
#008000 (Forest green)



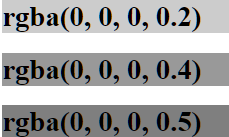
#00a000 (Dark lime green)

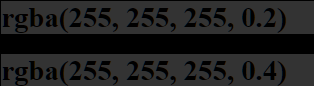


#eafbcb (Snow flurry)



#000000 (Black)





(J’ai affiché ces couleurs-là sur fond noir pour qu’on puisse les voir mais dans notre code ils apparaissent par-dessus l’overlay de l’animation des images de background)

Boutons :



Background: #008000 (Forest green)

Color: #eafbcb (Snow flurry)

Hover: Background: #00a000 (Dark lime green)



Background: rgba(255, 255, 255, 0.2)

Color: #eafbcb (Snow flurry)

Border: #008000 (Forest green)

Hover: Background: rgba(255, 255, 255, 0.4)

Pour le thème pâle notre couleur principale ça été #002500 (Deep fir) qu’on a utilisé pour le menu ainsi que pour faire un contraste avec le texte on hover. Ça a généré cette palette. Nous avons fini par utiliser plusieurs couleurs sur cette palette parce les couleurs pâles qu’on avait choisi au début étaient trop flashy et ça donnais mal aux yeux. Nous avons choisi un vert foncé pâle #4d664d(Nandor) comme deuxième couleur et ce vert pâle #f2f4f2(Honeydew) pour faire un contraste pour le texte et les borders des forms et certains div. On a aussi utilisé des couleurs rgba pour faire des overlays sur l’animation des images de background, pour la couleur de background des formulaires et de certains divs. Pour le thème light on a utilisé rgba(230, 233, 230, 0.2), rgba(230, 233, 230, 0.2), rgba(234, 251, 203, 0.5), rgba(242, 244, 242, 0.5), rgba(0, 37, 0, 0.5), rgba(0, 0, 0, 0.2), rgba(0, 0, 0, 0.5), rgba(255, 255, 255, 0.2) ou rgba(255, 255, 255, 0.4).

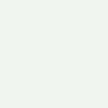




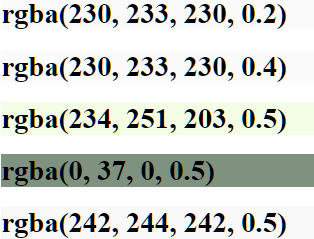
#002500 (Deep fir)



#4d664d(Nandor)



#f2f4f2(Honeydew)





Background: #002500 (Deep fir)

Color: #f2f4f2(Honeydew)

Border: #f2f4f2(Honeydew)

Hover: Background: #4d664d(Nandor), Color: #002500 (Deep fir)



Background: rgba (0, 0, 0, 0.5)

Color: #f2f4f2(Honeydew)

Border: #f2f4f2(Honeydew)

Hover: Background: rgba (234, 251, 203, 0.5), Box-shadow: rgba (255, 255, 255, 0.5), Color: #002500 (Deep fir)

Messages d’erreurs/succès:

Succès:



Color: #008000 (Forest green)

Erreur:



Color: #990000 (Red berry)

3. Javascript

Pour la partie Javascript nous avons utilisé jQuery et nous avons fait :

1. Un curseur avec une image d’arbre prise sur <https://icons8.com/> (Tu nous a donné l’idée)

2. Un effet lorsqu’on focus in sur un input type text, password, email ou textarea qui fait que le contour soit vert et s’enlève lorsqu’on focus out.

3. Une animation sur les formulaires de la page Accueil et la page Inscription qui anime leur position vers le haut

4. Une animation sur la page Contact qui affiche les divs des champs du formulaire, ensuite anime les margin top pour 1500 ms et revient à la normale. Ça donne l’impression que le formulaire s’élargit de bas en haut avant de revenir à sa forme normale.

5. Une autre animation sur la page Contact qui anime la position des divs qui affichent l’adresse et la carte vers la gauche et la droite.

6. Une fonction jQuery pour toggle entre les thèmes du site. Quand l’event click du lien Thème se déclenche on affiche le sous-menu et quand l’event click d’un des thèmes se déclenche on change de feuille de style selon le id du bouton cliqué.

7. Pour les icônes dans la page Livre d’or j’ai utilisé le style de FontAwesome (<https://fontawesome.com/>) et jQuery pour changer de classe quand l’utilisateur connecté clique sur le like/dislike ou annule son like /dislike

8. Pour afficher la carte je suis allé trouver comment faire sur <https://developers.google.com/maps/documentation/javascript/adding-a-google-map> . J’ai copié le code de la function initMap() sur cette page ensuite je suis allé chercher l’adresse du collège, puis j’ai copié le plus code (HC49+X2 Montréal, Québec) et collé dans google maps pour trouver la latitude et la longitude. Pour que la function initMAp() marche je me suis créé une clé pour le Maps Javascript API et je l’ai remplacé dans cette url que j’ai inclus dans le code de la page Contact :

<script defer src="https://maps.googleapis.com/maps/api/js?key=YOUR\_API\_KEY&callback=initMap">

</script>

Vidéo fait avec Loom pour expliquer le fonctionnement :

https://www.loom.com/share/84144abe9c0f4e369674c0f464dd82d2?fbclid=IwAR0jmxM70aoZKGWYMIrDGDc5VCg\_4cTuJDMZL1r0BKSxlHQdHZms\_ketUts

Journal de bord

Lundi 10 août 2020 : Nous avons fait le coding du projet au complet durant les vacances et tout ce qu’il reste à faire c’est le rapport que je suis en train d’écrire. Mon impression sur le projet, personnellement je l’ai trouvé difficile et très long. Je pense qu’à nous 2 ça nous a pris au moins 100 heures de travail. Bien réfléchir à comment faire la base de données c’était long. Il a fallu refaire la base de données plusieurs fois et faire beaucoup d’essai et erreur avec le PHP. J’ai peut-être été trop perfectionniste sur certains points et je me suis sûrement compliqué la vie mais je trouvais que l’énoncé du projet manquais un peu de précision. Parce qu’il n’y avait d’exemples précis j’ai été obligé de faire des recherches sur comment implémenter un système de likes/dislikes un peu comme sur Facebook (c’est comme ça que je le voyais), pour le système de login parce qu’il manquait certains détails sur openclassrooms, sur comment compter le nombre de visiteurs, comment trouver la date de la dernière modification et aussi sur les animations jQuery. Pour le nombre de visiteurs je me servais de l’adresse IP de l’utilisateur connecté pour le compter au début mais ça ne peut pas marcher pour un site qui existe seulement en locale. Alors je compte plutôt le nombre de fois que le visiteur a ouvert le fichier dans le browser et le nombre de refresh. Probablement tu ne nous aurais pas demander d’aller aussi loin, mais je suis quand même fier du résultat final.

Je pense que finir le projet sans les vacances seraient pratiquement impossible, en plus de l’autre gros projet à remettre ainsi que la préparation aux examens finals. Donc j’ai trouvé que ce projet était trop dur pour des débutants et je pense que je ne suis pas le seul à penser ça.