

Développement web :
Projet EyeCust

FRANTZ Charles-Philippe
frantzchar@eisti.eu

LAMARE Michèle
lamaremich@eisti.eu

NJOUKAM Claudelle
njoukamcla@eisti.eu

Vo Christelle
vohoangmai@eisti.eu

16 Juin 2019

Table des matières

I	Introduction	2
II	Mode d'emploi	3
II.1	EyeCust sur PC sous Linux	3
II.2	Eyecust sur PC sous Windows	4
III	Organisation du travail	7
III.1	Organisation générale	7
III.2	Répartition des tâches	7
IV	Cahier des charges	8
V	Difficultés rencontrées	9
V.1	Organisation	9
V.1.1	Controllers	9
V.1.2	Html2Pdf	9
V.1.3	Database	9
V.1.4	JS	9
V.1.5	Index.php	9
V.1.6	Views	10
V.2	Problèmes rencontrés	12
VI	Axes d'améliorations	13
VII	Conclusion	14

I Introduction

Afin de conclure notre semestre de développement web, nous devons réaliser un projet par groupe de quatre à cinq personnes. Dans ce groupe, nous sommes quatre élèves de la CPI2 C2 : Charles-Philippe FRANTZ, Michèle LAMARE, Claudelle NJOUKAM, Christelle VO.

Ce projet a pour objectif de mettre en pratique les techniques, fonctions, et langages étudiés. A l'aide des ressources sur AREL et de nos prises de notes, nous avons rédigé ce rapport pour présenter notre démarche.

Parmi les sujets proposés, nous avons choisi la réalisation du nouveau site de EyeCust. EyeCust est une marque qui vend des lunettes mais qui, surtout, permet de les personnaliser via la gravure de caractères sur les verres ; elle permet la customisation de montures.

Nous avons choisi ce sujet car non seulement nous portons tous des lunettes donc le sujet nous parle, mais les instructions données nous ont également semblé claires et réalisables par rapport à notre niveau. Les fondateurs de la marque ont été très précis sur ce qu'ils recherchaient, nous avons donc un cahier des charges quasiment fourni.

II Mode d'emploi

Afin de visualiser le fruit de notre travail, nous avons écrit deux modes d'emploi pour faire fonctionner notre programme, et donc accéder à notre site web.

II.1 EyeCust sur PC sous Linux

```
/*define("USERNAME", "njoukamcla");  
define("PASSWORD", "lgfiwuOyOh");  
define("DSN", "mysql:host=mysql.etude.cergy.eisti.fr;dbname=njoukamcla;charset=utf8");*/  
  
define("USERNAME", "root");  
define("PASSWORD", "");  
define("DSN", "mysql:host=localhost;dbname=eyecust;charset=utf8");
```

FIGURE 1 – Portion de code à adapter dans les fichiers *index.php* et *receipt.php*

Pour faire fonctionner notre programme sous le système d'exploitation Linux, il suffit de mettre en commentaire les trois dernières lignes de la figure ci-dessus, et de retirer les délimiteurs de commentaires sur les trois premières lignes. Ces six lignes de code à modifier se situent au début des fichiers *index.php* et *receipt.php*, sachant que *receipt.php* se trouve avec le chemin *Views/Receipt/receipt.php*.

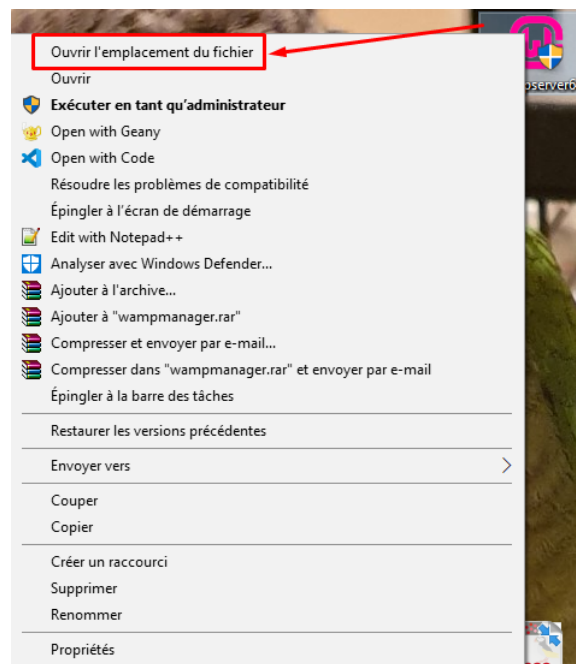
Pour ce qui est de la base de données sous Linux, il suffit d'entrer le nom d'utilisateur et le mot de passe du compte *phpMyAdmin* de Claudelle NJOUKAM afin d'accéder à notre base de données, mais seules les tables dont le nom commence par une minuscule font partie du projet EyeCust : admins, blog, cart, clients, contact, glasses, newsletter, police, et transactions.

Attention à ne pas oublier d'ouvrir un serveur via l'invité de commande Linux à l'intérieur du dossier Eyecust.

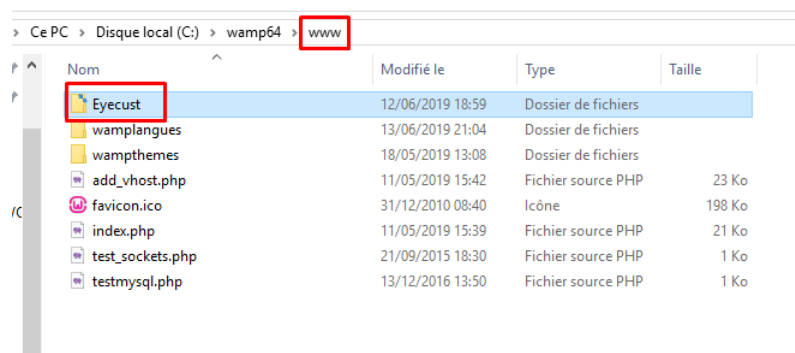
II.2 Eyecust sur PC sous Windows

Pour faire fonctionner notre programme sous le système d'exploitation Windows, il faut tout d'abord mettre en commentaire les trois premières lignes, et retirer les délimiteurs de commentaires sur les trois dernières lignes, comme sur la figure placée à la page précédente. Ces six lignes de code à modifier se situent au début des fichiers *index.php* et *receipt.php*, sachant que *receipt.php* se trouve avec le chemin *Views/Receipt/receipt.php*.

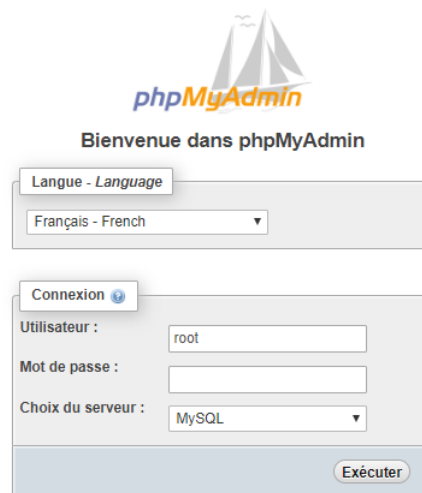
Pour ce qui est de la base de données sous Windows, il faut installer et lancer WAMP.



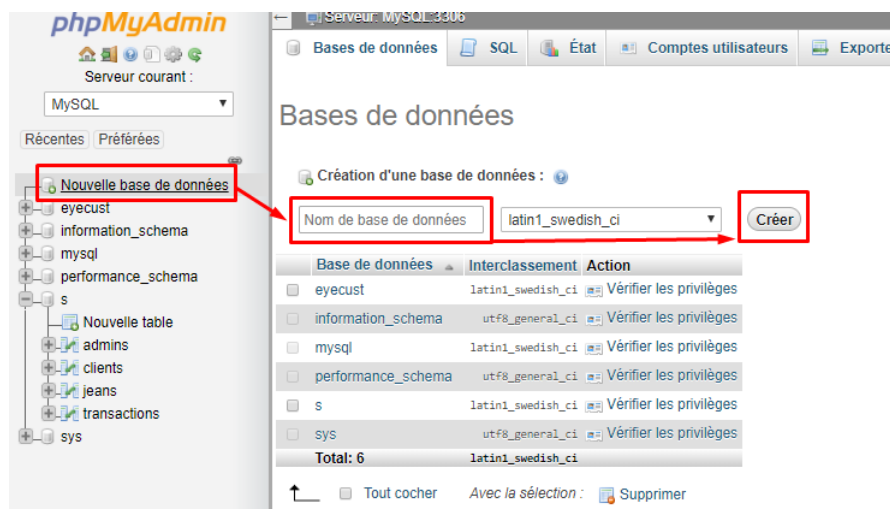
Étape 1 : Il faut installer, puis aller dans le répertoire où est installé WAMP. Le moyen le plus rapide est d'aller sur le raccourci se trouvant sur le bureau, puis de faire un click droit, et de cliquer sur « Ouvrir l'emplacement du fichier », comme vous pouvez le voir sur la figure ci-dessus.



Étape 2 : Une fois arrivé dans le dossier de Wamp, il faut aller dans le dossier *www*, et y placer notre dossier du projet, comme vous pouvez le voir sur la figure ci-dessus.



Etape 3 : Il faut aller sur *phpMyAdmin* en exécutant l'adresse *localhost/phpmyadmin* afin d'arriver à l'affichage ci-dessus. Le nom d'utilisateur que nous utilisons est « root », et le mot de passe est vide. Ensuite, il faut appuyer sur le bouton « Exécuter » afin d'accéder à la base de données.



Etape 4 : Une fois connecté dans *phpMyAdmin*, il va falloir créer une nouvelle base de données. Pour ce faire, il faut appuyer sur « Nouvelle base de données », puis lui donner un nom (ce nom devra être placé dans les lignes de code donc il est important), et appuyer sur le bouton « Créer ».

Structure SQL Rechercher Requête Exporter Importer

Importation dans la base de données « test »

Fichier à importer :

Le fichier peut être compressé (gzip, bzip2, zip) ou non.
Le nom du fichier compressé doit se terminer par `.[format].[compression]`. Exemple : `.sql.zip`

Parcourir les fichiers : Aucun fichier choisi (Taille maximale : 128Mio)

Il est également possible de glisser-déposer un fichier sur n'importe quelle page.

Jeu de caractères du fichier :

Importation partielle :

☒ Permettre l'interruption de l'importation si la limite de temps configurée dans PHP est sur le point

Ignorer ce nombre de requêtes (pour SQL), à partir du début :

Autres options :

☒ Activer la vérification des clés étrangères

Format :

Options spécifiques au format :

Mode de compatibilité SQL :

☒ Ne pas utiliser AUTO_INCREMENT pour la valeur zéro

Étape 5 : Il va falloir importer notre fichier *njoukamcla.sql* situé dans le dossier *DataBase*. Pour importer une base de données à partir d'un fichier externe, il faut se placer dans la nouvelle base de données créée précédemment. Ensuite, il faut sélectionner la rubrique « Importer », puis appuyer sur le bouton « Choisir un fichier », et sélectionner notre fichier *njoukamcla.sql*. Après avoir appuyé sur le bouton « Exécuter », notre base de données est prête à l'utilisation.

localhost/eyecust/

Étape 6 : Pour lancer notre programme, il suffira d'exécuter l'adresse *localhost/eyecust*. La page *index.php* s'affiche par défaut.

Attention à ne pas oublier de lancer Wamp avant d'effectuer ces six étapes.

III Organisation du travail

III.1 Organisation générale

Après avoir chacun lu le sujet de son côté, nous nous sommes réunis pour faire un résumé de ce que nous avons lu et de ce qu'il fallait en retenir, notamment du côté du cahier des charges.

Le principal point était que les fondateurs de EyeCust souhaitaient un site à la fois épuré et différent de leur site actuel réalisé à l'aide de WORDPRESS.

Nous avons donc consulté leur site, puis sélectionné ce que nous souhaitions garder ou non, ainsi que comment nous voulions réorganiser les pages et les rubriques de leur site. Pour finir, nous avons fait un schéma de la mise en page idéale que nous avions en tête, sans réellement savoir si nous pourrions l'atteindre.

III.2 Répartition des tâches

Afin que cela soit plus limpide, voici un tableau récapitulatif de la répartition du travail.

Elève / Tâche	HTML / PHP	CSS	JavaScript	BDD / MySQL	Rapport
Charles-Philippe	✓	✓	X	✓	✓
Michèle	✓	✓	X	X	✓
Claudelle	✓	X	X	✓	✓
Christelle	✓	X	✓	✓	✓

IV Cahier des charges

Mise en page du site : Dans le but de garder un site épuré et agréable à parcourir, nous avons privilégié des tons et des sous-tons clairs avec les couleurs d'EyeCust : gris, blanc, noir, et orange. Un header avec les différentes rubriques du site ; un footer avec les réseaux sociaux, ainsi que les liens de contact et de conditions générales de ventes ; une page d'accueil avec les trois articles les plus récents.

Compte particulier : Compte pour un client particulier ; avec identifiants et mot de passe ; vérification de l'unicité du nom d'utilisateur et de l'adresse e-mail ; obligation d'un mot de passe à au moins huit caractères ; si tout est conforme, ajout des informations du nouveau client dans la table clients. Le client particulier peut commander des montures, mais également ajouter une personnalisation avec le texte, la police d'écriture, ou/et le motif de son choix, puis laisser un avis sur des montures. Possibilité de consulter l'historique de ses commandes.

Compte professionnel : Compte pour un client professionnel, une entreprise, ou un grossiste ; avec identifiants et mot de passe ; emplacement de texte pour sa description ; vérification de l'unicité du nom d'utilisateur et de l'adresse e-mail ; obligation d'un mot de passe à au moins huit caractères ; si tout est conforme, ajout des informations du nouveau client dans la table clients. En plus de ce que le client particulier peut faire, le professionnel peut commander des montures en masse, et accéder au stock des montures restantes. Possibilité de consulter l'historique de ses commandes.

Compte administrateur : Compte de l'administrateur ; qui permet d'ajouter et de retirer des modèles, articles, polices. En accédant à la page de son compte, l'administrateur peut accéder à l'historique de toutes les commandes effectuées et aux reçus pdf des clients, mais également modifier leur prix.

Boutique : Boutique avec l'ensemble des lunettes proposées. Mise en place d'un filtre Unisexe/Homme/Femme ; et d'un classement ordre alphabétique, ordre alphabétique inversé, prix croissants, prix décroissants ; possibilité de filtrer et de classer en même temps.

Panier : Possibilité d'ajouter ou de supprimer des éléments en prenant en compte la personnalisation des lunettes et le client associé ; mise en place d'une facture.

Avis : Formulaire qui permet aux clients (professionnels et particuliers) de donner leurs avis sur des montures afin d'aiguiller les autres clients dans leurs choix.

Actualité : Page référençant tous les articles liés à EyeCust.

Newsletter : Mise en place d'une newsletter par laquelle les fondateurs d'EyeCust pourraient envoyer des articles en exclusivité sur leur histoire ou des montures en avant-première.

Contact : Formulaire qui permet aux clients (professionnels et particuliers) d'envoyer un message à l'équipe EyeCust, ensuite récupéré dans la table contact.

V Difficultés rencontrées

A présent, nous allons expliquer les problèmes rencontrés tout au long du projet, et comment nous les avons résolus.

V.1 Organisation

La première difficulté a été de choisir notre structure. En effet, allons-nous utiliser du Bootstrap, du flexbox grid, ou bien du CSS et HTML simple? Nos pages seraient-elles en PHP ou en HTML? Comment allaient-elles interagir, et comment organiser nos fichiers? Les réponses à toutes ces questions allaient être décisives dans la suite du projet.

Finalement, nous avons créé un fichier *index.php* avec l'en-tête, le pied de page, et la mise en page, tels que nous le souhaitions. Ainsi, ils n'avaient pas besoin d'être figurés dans chaque fichier php dédié à une page. Ensuite, tous les fichiers php sont appelés à l'aide de controllers, et le contenu des fichiers php dédiés à des pages est affiché entre le header et le footer du fichier *index.php*. Leur appel s'affiche notamment dans l'URL. Voici en détail ce que contient chaque dossier et sous-dossiers :

V.1.1 Controllers

HomeController.php : Le fichier *HomeController.php* se charge des redirections vers les fichiers php du dossier *Views*. La classe **HomeController** contient les méthodes à appeler en fonction de l'action.

V.1.2 Html2Pdf

Ce dossier contient tous les fichiers nécessaires à la génération d'un reçu pdf d'une commande. Le contenu de ce reçu est donné par le fichier *receipt.php*.

V.1.3 Database

Le fichier *njoukamcla.sql* est la base de données exportée de notre projet.

V.1.4 JS

Le fichier *jquery-3.2.1.js* simplifie le code en comportant l'essentiel des fonctions permettant d'agir au sein d'un fichier php selon les actions de l'utilisateur.

V.1.5 Index.php

Contient la barre de navigation et le pied de page, ainsi que l'appel des controllers. Les redirections vers les autres pages php sont obtenues via l'utilisation de controllers. C'est de ce fichier que tout notre site « découle ». C'est donc celui-ci qu'il faut ouvrir pour pouvoir utiliser toutes les fonctionnalités du site.

V.1.6 Views

Registration : Le fichier *registration.php* contient le formulaire d'inscription pour devenir client. Il y a deux types de clients possibles : professionnel et particulier. Cette information est récupérée à l'aide du bouton radio « ch ». La partie php du fichier vérifie l'unicité du nom d'utilisateur et de l'adresse e-mail, impose un mot de passe à au moins huit caractères et, si tout est conforme, ajoute les informations du nouveau client dans la table clients, puis redirige vers la page de connexion du *login.php* pour que le nouveau client se connecte.

Login et Logout : Le fichier *login.php* contient le formulaire de connexion (professionnel, particulier, ou administrateur). La partie php de ce fichier vérifie si tous les champs obligatoires sont bien remplis, si le mot de passe est conforme, si le nom d'utilisateur et le mot de passe saisis correspondent bien à un compte dans la BDD (admins ou clients). Le fichier *logout.php* ferme la session, et redirige vers la page d'accueil du *home.php*.

Account : Le fichier *account.php* est nécessaire pour visualiser les commandes et les reçus pdf. L'administrateur peut visualiser les commandes de tous les comptes.

Glasses : Le fichier *glasses.php* a pour objectif l'affichage des montures de la boutique présentes dans la table glasses. Cela permet d'afficher les montures, leur description, leur prix. L'administrateur peut ajouter un modèle, et supprimer ceux déjà présents.

Buy : Le fichier *buy.php* récupère le nom d'utilisateur, le panier, la date, le prix total de la commande, le type du compte (administrateur, professionnel, ou particulier), puis les ajoute dans la table transactions.

Cart : Le fichier *cart.php* récupère les lunettes ajoutées au panier, et les affiche avec la quantité et le prix associés.

Blog : Le fichier *blog.php* affiche l'article choisi.

Avertissement : Le fichier *avertissement.php* permet de demander au client s'il souhaite réellement supprimer son panier. S'il confirme, alors son panier est supprimé.

CGV : Le fichier *cgv.php* comporte les conditions générales de vente de la marque EyeCust.

Edit : Le fichier *edit.php* permet à l'administrateur de modifier le prix des commandes effectuées de tout compte.

Editboutique : Le fichier *editboutique.php* permet à l'administrateur de supprimer un modèle de la boutique EyeCust.

Editblog : Le fichier *editblog.php* permet à l'administrateur de supprimer un article de l'actualité.

Editpolice : Le fichier *editpolice.php* permet à l'administrateur de supprimer une police d'écriture parmi le choix des polices proposées pour la personnalisation.

Home : Le fichier *home.php* permet d'afficher un message de bienvenue à l'utilisateur.

Addboutique : Le fichier *addboutique.php* permet à l'administrateur d'ajouter un modèle à la boutique EyeCust.

Addblog : Le fichier *addblog.php* permet à l'administrateur d'ajouter un article à l'actualité.

Addreview : Le fichier *addreview.php* permet à tout compte de publier un avis sur la monture choisie.

Actualite : Le fichier *actualité.php* affiche tous les articles de la marque EyeCust.

Contact : Le fichier *contact.php* propose un formulaire de contact au client (professionnel ou particulier) afin d'envoyer un message aux membres de l'équipe EyeCust et, pour l'administrateur, affiche les messages du plus récent au plus ancien.

V.2 Problèmes rencontrés

Voici les problèmes rencontrés que nous n'avons pas pu résoudre :

Panier : Nous pouvons supprimer tout le panier, mais pas un seul modèle.

Personnalisation : Par manque de temps pour changer la gestion du panier, si le client désire réaliser une personnalisation, il est redirigé vers le formulaire de contact dans lequel il peut expliquer ce qu'il souhaite.

Filtre : Nous avons vu un filtre (dans la visée de celui de Snapchat) qui permet d'essayer des modèles mais, les montures n'étant pas celles d'EyeCust, nous n'avons pas vu l'intérêt de l'intégrer à notre site.

Fusion : Nous travaillions avec Google Drive, et mettions à jour les fichiers qui y étaient. Toutefois, nous n'avons pas fait les fusions assez régulièrement, donc nous avons eu des problèmes de casses entre le CSS et le PHP.

VI Axes d'améliorations

Tout d'abord, nous aurions voulu améliorer le responsive du site, notamment la barre de menu et celui de l'affichage de la boutique. De ce côté, nous aurions également souhaité recréer l'affichage de leur site actuel de la boutique, il nous paraissait à la fois légèrement interactif et design. La mise en place d'une FAQ (foire aux questions) en réponse au formulaire de contact pour les questions les plus communes aurait été intéressante, et pourrait éventuellement se faire dans le futur puisque toutes les questions sont récupérées dans une table.

Nous avons décidé de fusionner certaines pages comme contact et contact professionnel. L'espace administrateur est, quant à lui, intégré à la page de login. Nous avons également enlever la page protection : elle est dans le dossier *Views*, dans le dossier *CSS*, et en commentaire dans le fichier *index.php*, mais n'a pas de sens à elle toute seule. Pour nous, elle devrait aller dans la description de chaque lunette, avec une partie qui indiquerait comment en prendre soin et leurs caractéristiques.

Par ailleurs, sur le plan des avis clients, il faudrait vérifier que le client ne puisse commenter que les montures qu'il a achetées. De même, il serait avantageux pour EyeCust de créer un compte Trustpilot et de le relier à leur site, c'est l'un des indicateurs de confiance de nombreux sites.

Sur la mise en page, le site n'est pas trop interactif mais non plus comme une vitrine, il est assez simple et agréable à regarder bien qu'un peu triste. Toutefois, nous aurions privilégié la mise en place d'un menu sticky (qui se déplace le long de la page) avec les liens vers les réseaux sociaux, plutôt que de les mettre dans le pied de page, cela favoriserait l'utilisations de ces liens.

Sur le plan de sécurité, trois éléments nous inquiètent :

- le compte professionnel : il faudrait pouvoir vérifier un numéro comme le siret par exemple.
- le paiement : il s'agit d'un site commercial, nous ne pouvons pas nous permettre qu'il ne soit pas sécurisé.
- les liens : une personne qui connaît le lien fourni lorsque nous nous connectons avec les identifiants d'administrateur peut copier ce lien et décider d'ajouter ou de supprimer des modèles, voire même de modifier les prix des commandes.

Pour terminer, nous aurions bien voulu mettre en place l'essayage virtuel des lunettes, mais nous n'avions aucun modèle pour les essayer et, sinon, au moins avoir un rendu avec les polices dessus, sinon c'est très difficile pour un client de se faire un idée.

VII Conclusion

Pour conclure, nous pouvons dire que bien que différent sur le plan de la mise en page que nous avons en tête, notre site est tout à fait fonctionnel et répond en grande partie aux critères indiqués par les fondateurs d'EyeCust et au cahier des charges que nous avons établi. L'esthétique n'est pas exactement comme nous l'aurions souhaité car, bien qu'avertis au début du projet, nous avons sous-estimé à quel point le CSS, le PHP, et le Javascript étaient liés.

Nous sommes néanmoins content du résultat et pensons que, avec plus de temps, nous pourrions le terminer et l'améliorer. Ce projet a été pour nous un bon aperçu de ce que serait un travail de groupe en entreprise, et a permis de nous améliorer rapidement, ce qui donne l'impression qu'il reste encore beaucoup à apprendre en développement web.