

PHP Model

Table des matières

1	Analyse préliminaire	3
1.1	Introduction	3
1.2	Objectifs.....	3
	Planification initiale	5
2	Analyse / Conception / Concept	6
2.1.1	Web-site Template.....	6
2.1.2	Base de données.....	7
	7
	Rouge : Utilisateur [Register, Login, newsletter] aucune dépendance	7
2.1.3	PowerShell	7
2.2	Stratégie de test.....	8
2.3	Risques techniques	9
2.4	Planification	10
2.5	Dossier de conception	10
2.5.1	Dell OptiPlex 7050	10
2.5.2	Outils et autres.....	11
2.5.3	Site web.....	11
2.5.4	Concept de code.....	15
3	Réalisation.....	18
3.1	Dossier de réalisation	18
3.2	Description des tests effectués	20
3.3	Erreurs restantes	21
3.4	Liste des documents fournis	21
4	Conclusions	21
5	Annexes.....	22
5.1	Résumé du rapport du TPI / version succincte de la documentation	22
5.2	Sources – Bibliographie.....	22
5.3	Journal de travail	22
5.4	Manuel d'Installation	22
5.5	Manuel d'Utilisation.....	22
5.6	Archives du projet.....	22

NOTE L'INTENTION DES UTILISATEURS DE CE CANEVAS:

Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.

De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.

1 Analyse préliminaire

1.1 Introduction

Ce projet consiste à créer un modèle PHP clair qui peut générer un nouveau site à chaque fois en utilisant PowerShell. Cela me permettra d'améliorer mes compétences en PHP et d'apprendre à écrire du code qui peut être facilement expliqué et modifié avec les commentaires qui l'accompagnent.

1.2 Objectifs

Selon les instructions données, le candidat doit d'abord analyser et concevoir les différents modules de l'application. Il s'agit de modéliser la base de données pour chaque module et de créer un prototype des pages modèles et de l'application de configuration du site. Une fois cette étape terminée, le candidat passera au développement, effectuera des tests unitaires, puis des tests d'intégration.

Les pages proposées doivent être réactives et s'adapter aux différents appareils tels que les tablettes, les téléphones mobiles et les ordinateurs, en s'adaptant à chaque fois à la taille de l'écran.

La sécurité de la base de données est cruciale et les mots de passe ne doivent pas être stockés en texte clair. Le candidat doit choisir un mot de passe qui répond aux critères de sécurité contemporains.

L'objectif du projet est :

- Le script va copier le site modèle et va demander à l'utilisateur les fonctionnalités à implémenter. Si l'utilisateur demande à supprimer une fonctionnalité, celle-ci sera retirée des pages. En finalité, l'utilisateur aura un nouveau site avec les configurations souhaitées
- A la fin de la génération, un fichier SQL sera généré permettant d'importer la base de données dans le système souhaité. Le fichier sera également adapté par rapport aux choix de l'utilisateur
- La page d'inscription doit demander aux utilisateurs de saisir leurs informations personnelles, telles que leur nom, leur adresse, leur adresse électronique, leur numéro de téléphone, leur nom d'utilisateur et leur mot de passe. Il est important de noter que les champs relatifs au nom, à l'adresse, à l'identifiant, au mot de passe et à l'adresse électronique sont obligatoires.
- Pour garantir la sécurité des données des utilisateurs, je choisirai une méthode sécurisée pour le stockage des mots de passe et j'expliquerai ma décision. Les utilisateurs doivent pouvoir se connecter avec leur nom d'utilisateur et leur mot de passe, et une option de réinitialisation du mot de passe doit être disponible en cas d'oubli.

- Newsletter signup : L'inscription doit permettre d'ajouter une adresse e-mail à une base de données. L'adresse e-mail doit être dans un format correct et interdire les caractères non autorisés. La page doit également envoyer un e-mail pour valider l'adresse.
- Contact/feedback form : La page de contact doit permettre à l'utilisateur d'entrer son nom, prénom, son adresse e-mail, son numéro de téléphone et un message. Tous les champs sont obligatoires. L'utilisateur peut choisir via un checkbox par quel moyen il souhaite être contacté (e-mail ou téléphone) Lors de la validation du formulaire, un message doit être envoyé à l'administrateur du formulaire et à l'utilisateur. L'adresse e-mail doit être valide.
- Social media integration : Pour les médias sociaux, l'utilisateur doit pouvoir afficher le fil de sa page Facebook ou Instagram avec les publications de celle-ci. Il doit pouvoir choisir s'il souhaite les deux réseaux ou uniquement l'un ou l'autre.

Un plugin peut être utilisé pour cette partie. Le choix devra être justifié par le candidat.

- User profile management : La gestion du profil doit reprendre les champs identiques à la page de login ci-dessus. L'utilisateur doit pouvoir modifier toutes ses données et les contrôles doivent être appliqués (adresse e-mail, champs obligatoires).

L'utilisateur doit pouvoir, en plus, supprimer son profil et ses informations.

- Image galleries : La page de la galerie doit permettre à un utilisateur de choisir un dossier de photos après importation de ce dernier sur le serveur web. La page doit demander à l'utilisateur le nombre de photos à afficher sur une ligne en miniature. Le nombre de lignes dépendra du nombre de photos choisies.

La suppression et l'ajout d'images peut se faire via une interface dédiée. Lors de la suppression, l'image devra être supprimée du serveur.

L'affichage des images lors d'un clic sur celles-ci peut se faire via un plugin. Le choix est libre pour le candidat qui devra le justifier dans la documentation.

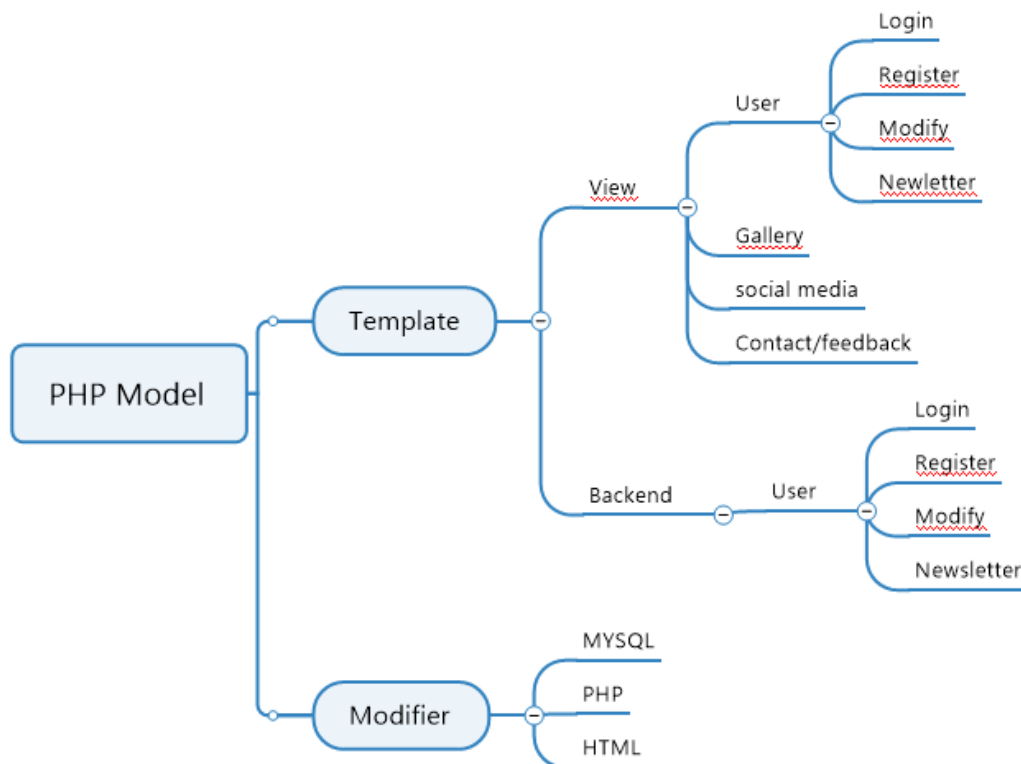
Un plugin peut être utilisé pour cette partie. Le choix devra être justifié par le candidat.

Planification initiale

les chiffres correspondent aux heures des 90 heures dont je dispose

ACTIVITÉ	DÉBUT DU PLAN	DURÉE DU PLAN	DÉBUT RÉEL	DURÉE RÉELLE	POURCENTAGE ACCOMPLI
Meeting initial	1	1			0%
Analyse du début	2	5			0%
Planification	2	5			0%
Récherches	7	2			0%
Frontend User Manager	9	2			0%
Backend User Manager	11	2			0%
Newsletter	13	2			0%
Contact / Feedback	15	2			0%
Facebook integration	17	3			0%
Instagram integration	20	3			0%
Galery Frontend	23	5			0%
Galery Backend	28	5			0%
MCD	33	3			0%
MLD template creation	36	3			0%
SQL script template creation	39	2			0%
Publishing Frontend/PHP and SQL	41	3			0%
Template test	44	3			0%
Analyse du milieu	47	3			0%
Récherches	50	3			0%
SQL generator	53	7			0%
PHP generator	60	7			0%
HTML generator	67	7			0%
Generator test	74	2			0%
PowerShell test	76	1			0%
Visite de Monsieur Charmier EX2	77	1			0%
Test general	78	3			0%
documentation	81	7			0%
Rendu	88	2			0%

2 Analyse / Conception / Concept

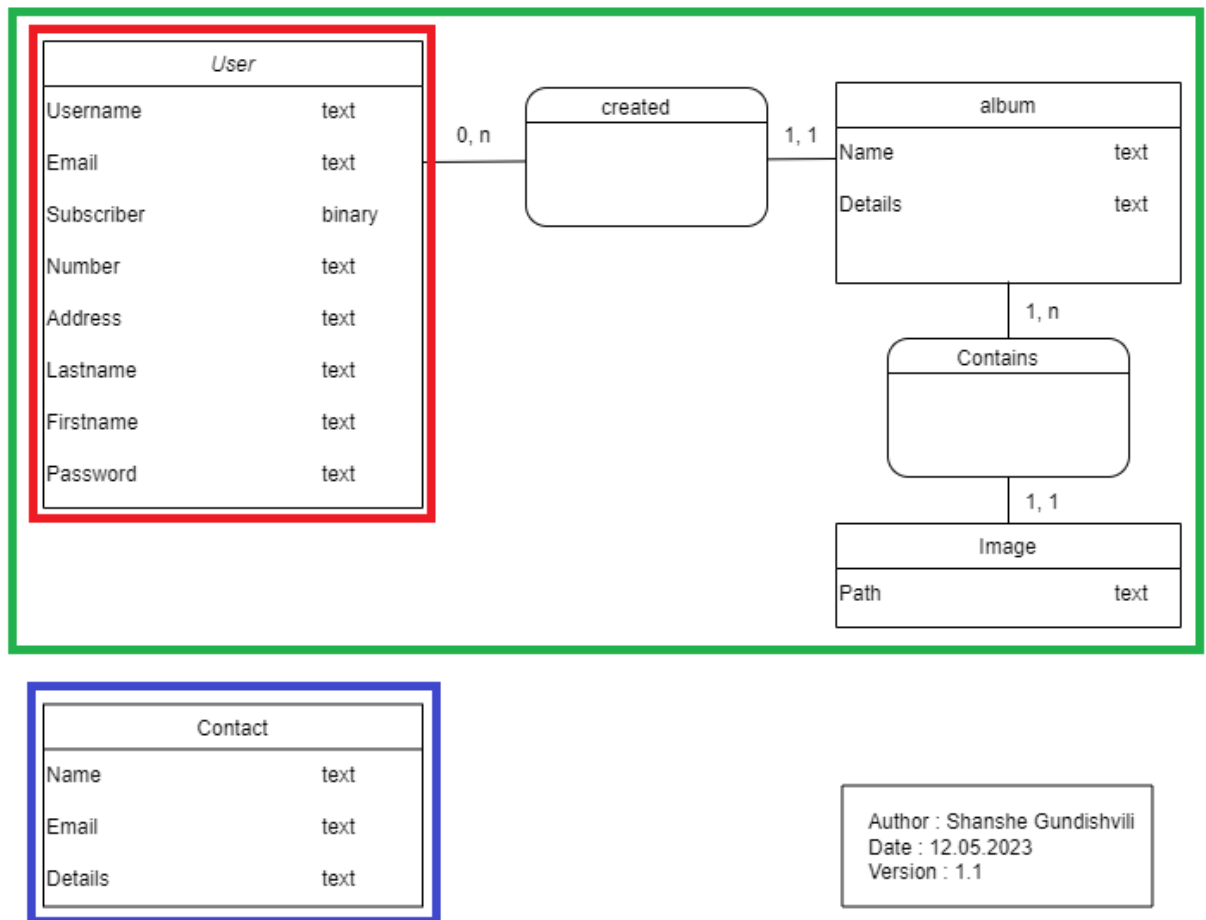


2.1.1 Web-site Template

Je vais créer un site web qui servira de modèle pour mon projet principal, qui est essentiellement un modificateur de texte, dans ce cas ce sera un site web, mais il peut être utilisé sur beaucoup d'autres langages et frameworks.

Ce site n'utilisera pas d'autres frameworks que le pack CSS et Javascript fourni avec le modèle, je travaillerai avec du PHP, du JavaScript, du HTML et du CSS purs.

2.1.2 Base de données



Rouge : Utilisateur [Register, Login, newsletter] aucune dépendance
Vert : Album/Galerie [créer/modifier un album] dépendant de l'utilisateur
Bleu : Contact/feedback [Contact] pas de dépendances

J'ai conçu mon MCD de cette manière parce que c'est la version la plus facile à modifier sans rien casser à l'avenir. L'utilisateur et le contact ne dépendant de rien d'autre, cela me permet de mettre des balises sans trop de soucis.

2.1.3 PowerShell

J'utiliserai des balises dans le code pour marquer les fonctions, les balises ressembleront à `//<tag>` pour ouvrir et `//</tag>` pour fermer dans tous les langages (PHP, HTML, MySQL, etc) puis j'utiliserai PowerShell pour supprimer entre ces balises tout ce que j'ai besoin de supprimer.

Version brute :

```

1  # Define tags
2  $startTag = "//<tag>"
3  $endTag = "//</tag>"
4
5  # Find start and end positions
6  $startIndex = $content.IndexOf($startTag)
7  $endIndex = $content.IndexOf($endTag)
8
9  # Check if tags were actually found
10 if ($startIndex -ne -1 -and $endIndex -ne -1) {
11     # Remove the section between the start and end tags
12     $content = $content.Remove($startIndex, $endIndex - $startIndex + $endTag.Length)
13 }
14
15
16 # Write modified content back to PHP files
17 Set-Content -Path C:\PHPModel\TMP\index.php -Value $content

```

2.2 Stratégie de test

- Analyse et conception des modules : Dans un premier temps, j'effectuerai une analyse approfondie de chaque module et je créerai un modèle de base de données pour chacun d'entre eux. Je créerai ensuite un prototype des pages du modèle et de l'application de configuration du site. Pour m'assurer que ces modules fonctionnent correctement, j'effectuerai une série de tests unitaires sur chacun d'entre eux.

- Conception réactive : Pour m'assurer que le site web est réactif et qu'il s'adapte bien à différentes tailles d'écran, je le testerai sur divers appareils tels que des tablettes, des téléphones mobiles et des ordinateurs. Je vérifierai les fonctionnalités, la mise en page et la conception du site web sur différentes tailles d'écran.

- Sécurité des mots de passe : Les mots de passe étant un élément essentiel de la sécurité du site web, je veillerai à ce qu'ils soient stockés en toute sécurité et qu'ils ne soient pas visibles en texte clair. Pour ce faire, je sélectionnerai une méthode sécurisée de stockage des mots de passe et effectuerai des tests approfondis pour vérifier qu'elle fonctionne comme prévu.

- Génération de sites : Pour m'assurer que la fonction de génération de site fonctionne comme prévu, je créerai des cas de test couvrant tous les scénarios possibles auxquels un utilisateur peut être confronté. Je testerai la fonction de génération de site pour m'assurer que le site généré correspond aux spécifications de l'utilisateur et que le fichier SQL est généré correctement.

- Page d'inscription : Pour m'assurer que la page d'inscription fonctionne correctement, je créerai des cas de test pour vérifier que tous les champs obligatoires sont correctement validés et que les informations de l'utilisateur sont correctement sauvegardées. Je m'assurerai également que la méthode de stockage des mots de passe fonctionne correctement et que les utilisateurs peuvent réinitialiser leur mot de passe si nécessaire.

- Inscription à la lettre d'information : Pour m'assurer que la fonction d'inscription à la lettre d'information fonctionne comme prévu, je vais créer des cas de test qui couvrent tous les scénarios possibles qu'un utilisateur peut rencontrer. Je testerai la fonction d'inscription à la lettre d'information pour m'assurer que la validation de l'adresse électronique fonctionne correctement et que l'adresse électronique est correctement ajoutée à la base de données.
- Formulaire de contact/réponse : Pour m'assurer que le formulaire de contact/réponse fonctionne comme prévu, je créerai des cas de test couvrant tous les scénarios possibles auxquels un utilisateur peut être confronté. Je testerai le formulaire de contact/réaction pour m'assurer que tous les champs obligatoires sont correctement validés et que le message électronique est correctement envoyé à l'administrateur et à l'utilisateur.
- Intégration des médias sociaux : Pour m'assurer que l'intégration des médias sociaux fonctionne comme prévu, je testerai le plugin que j'ai sélectionné pour m'assurer qu'il affiche correctement le flux Facebook ou Instagram de l'utilisateur. Je testerai également le plugin pour m'assurer que l'utilisateur peut choisir les plateformes de médias sociaux à afficher.
- Gestion des profils d'utilisateurs : Pour m'assurer que la fonction de gestion des profils d'utilisateurs fonctionne comme prévu, je créerai des cas de test couvrant tous les scénarios possibles auxquels un utilisateur peut être confronté. Je testerai la fonction de gestion du profil de l'utilisateur pour m'assurer que toutes les données de l'utilisateur peuvent être correctement modifiées et que tous les champs obligatoires sont correctement validés.
- Galeries d'images : Pour m'assurer que la fonction de galeries d'images fonctionne comme prévu, je créerai des cas de test couvrant tous les scénarios possibles auxquels un utilisateur peut être confronté. Je testerai la fonctionnalité des galeries d'images pour m'assurer que les images peuvent être correctement importées, affichées et supprimées.

2.3 Risques techniques

L'un des défis potentiels que je pourrais rencontrer est l'utilisation de l'API d'Instagram. Cela peut être compliqué car il faut avoir à la fois un compte professionnel sur Instagram et un compte développeur sur Meta.

Compte professionnel Instagram : Pour accéder à l'API Instagram, je dois disposer d'un compte professionnel sur Instagram. Cela implique de créer un compte Instagram et de passer à un profil professionnel en fournissant des informations supplémentaires sur mon entreprise ou mon organisation.

Compte de développeur Meta : En plus d'un compte professionnel Instagram, je dois également m'inscrire en tant que développeur sur la plateforme Meta, qui est la société mère de Facebook (propriétaire d'Instagram). Cela implique de

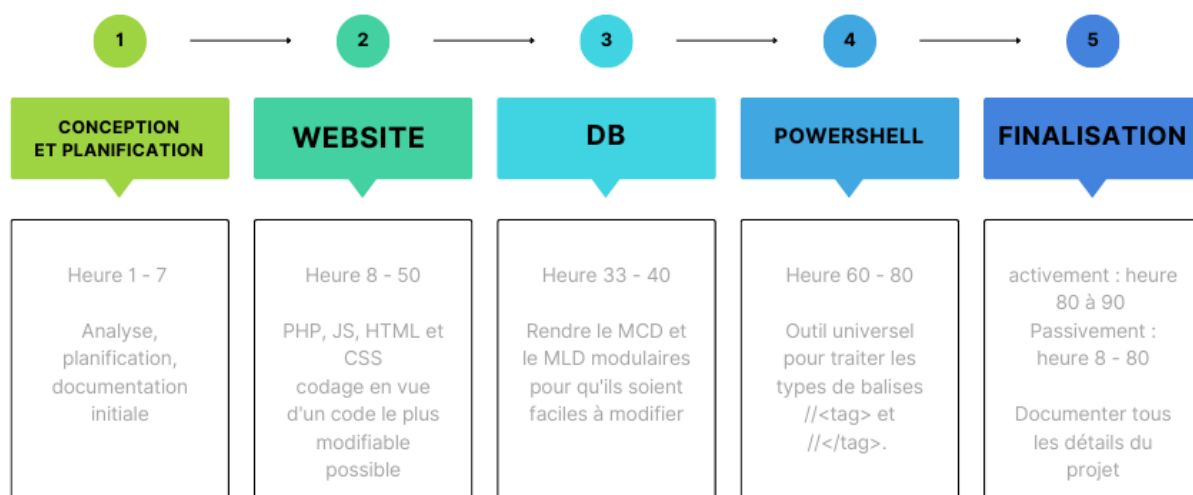
s'inscrire en tant que développeur et de créer une application pour obtenir les identifiants et les clés API nécessaires à l'utilisation de l'API Instagram.

2.4 Planification

Du mardi 2 mai 2023 à 8h50 au vendredi 2 juin 2023 à 14h05

Lundi	08h00-12h15	13h30-16h55
Mardi	08h50-12h15	13h30-16h05
Jeudi	08h00-10h35	13h30-16h55
Vendredi	08h00-12h15	13h30-15h05

PHP Model



L'idée du projet étant qu'il soit modifiable et pratique pour ajouter de nouvelles fonctionnalités, le projet devrait être aussi modulaire que possible. L'idée est d'avoir une évolutivité durable avec un script PowerShell qui n'a pas besoin de changer le cœur du code, si des fonctions supplémentaires sont ajoutées, il suffit d'ajouter les nouveaux paramètres à la liste des balises avec les fonctions associées. Pour cette raison et mon peu d'expérience, j'ai choisi de ne pas utiliser le PHP orienté objet mais le PHP traditionnel avec la structure MVC.

2.5 Dossier de conception

2.5.1 Dell OptiPlex 7050

Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
Intel(R) HD Graphics 530
Ram: 16Go 2133 Hz

Windows 10 Education edition

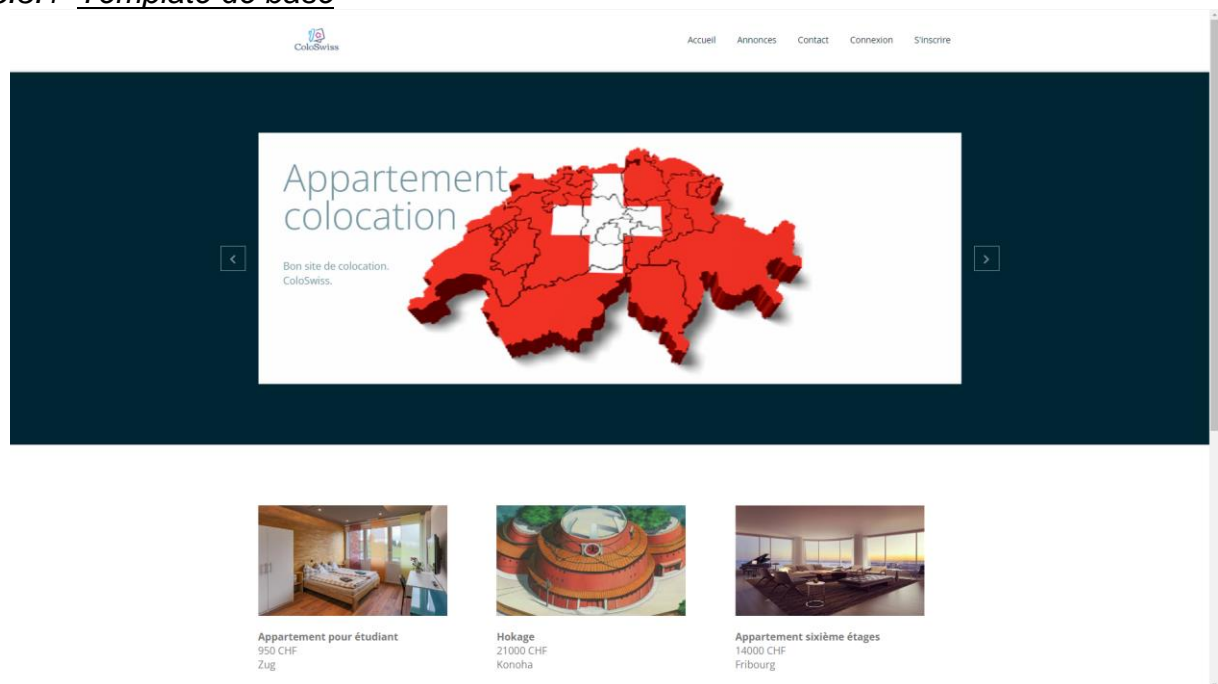
2.5.2 Outils et autres


Draw.io	MCD
MySQL Workbench	MLD
PHPStorm	API PHP/HTML/CSS/JS
Windows PowerShell ISE	PowerShell
PHP 8.2	PHP
XDebug	Debugger PHP
Office 365	Documentation
Sociablekit.com	Plugin Instagram
Compte développeur facebook	Pour Plugin Instagram
Instagram Business account	Webhook Instagram
Mind Manager	Mind manager

2.5.3 Site web

J'utiliserai un modèle auquel je suis déjà habitué, c'est pourquoi je n'utiliserai pas d'outils tiers pour simuler les pages.


2.5.3.1 Template de base





[Accueil](#)
[Annonces](#)
[Contact](#)
[Connexion](#)
[S'inscrire](#)

Enregistrement


COMPANY INFORMATION



Company Address
Avenue De La Gare 26,
Sainte-Croix, Vaud,
Switzerland



E-mail
coloswiss@gmail.com



Phone Number
079 895 91 73

[Upload Photo now](#)

ENREGISTREMENT





Email address *


We'll never share your email with anyone else.

Password *


Password confirm *

[S'enregistrer](#)








[Accueil](#)
[Annonces](#)
[Contact](#)
[Connexion](#)
[S'inscrire](#)


Annonces




Appartement pour étudiant
950 CHF
Zug




Hokage
21000 CHF
Konooha




Appartement sixième étages
14000 CHF
Fribourg




Maison avec ferme
7894 CHF
Yverdon




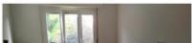
Appartement Familial
7877 CHF
Yers



mansion
69420 CHF
orbe







cpnv
Centre professionnel
du Nord vaudois

Accueil Annonces Contact **Connexion** S'inscrire

Se connecter

COMPANY INFORMATION

Company Address
Avenue De La Gare 26,
Sainte-Croix, Vaud,
Switzerland

E-mail
coloswiss@gmail.com

Phone Number
079 895 91 73

SE CONNECTER

Email *
Enter user name

Password *
Password

Connexion

f t in y

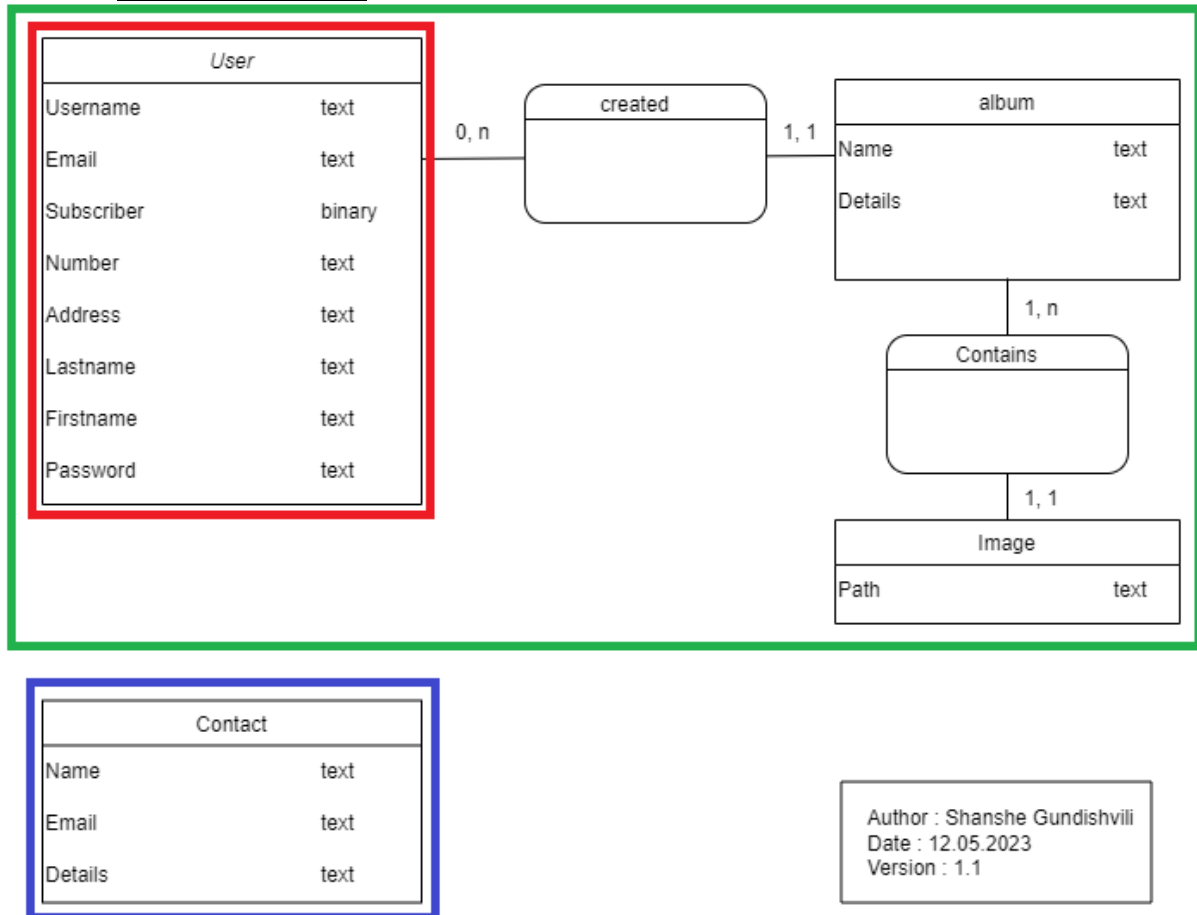
OUR PHILOSOPHY CONTACT US

2.5.3.2 Hébergement

J'ai reçu un espace sur SwissCenter, mais je n'y mettrai pas de site jusqu'à ce que je mette en place le lien GitHub et le lien de téléchargement pour le public. C'est parce que le site que je suis en train de faire n'est pas mon projet actuel mais une partie du projet entier.

modelephptpisha.mycpnv.ch (ne fonctionne pas pour l'instant)

2.5.3.3 Base de données



2.5.3.3.1 Tables

Table **users**:

Champs: id (INT), username (VARCHAR), email (VARCHAR), subscriber (INT), number (VARCHAR), address (VARCHAR), lastname (VARCHAR), firstname (VARCHAR), password (VARCHAR).

La colonne id est une clé primaire.

Table **albums**:

Champs: id (INT), name (VARCHAR), details (VARCHAR), users_id (INT).

La colonne id est une clé primaire.

Table **images**:

Champs: id (INT), path (VARCHAR), albums_id (INT).

La colonne id est une clé primaire

Table **contacts** :

Champs : id (INT), name (VARCHAR), email (VARCHAR), details (VARCHAR).

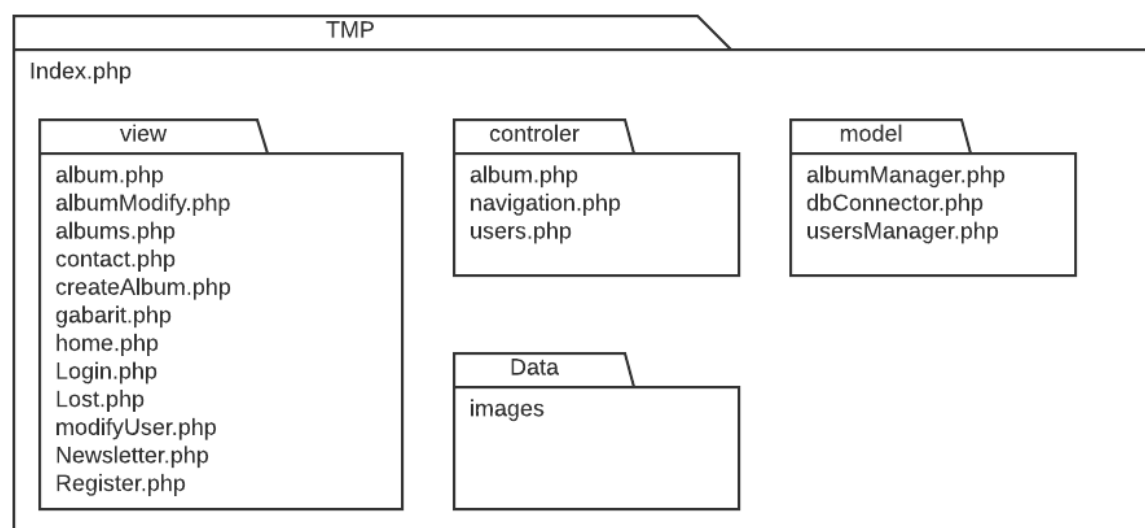
La colonne id est une clé primaire.

2.5.3.3.2 Relations

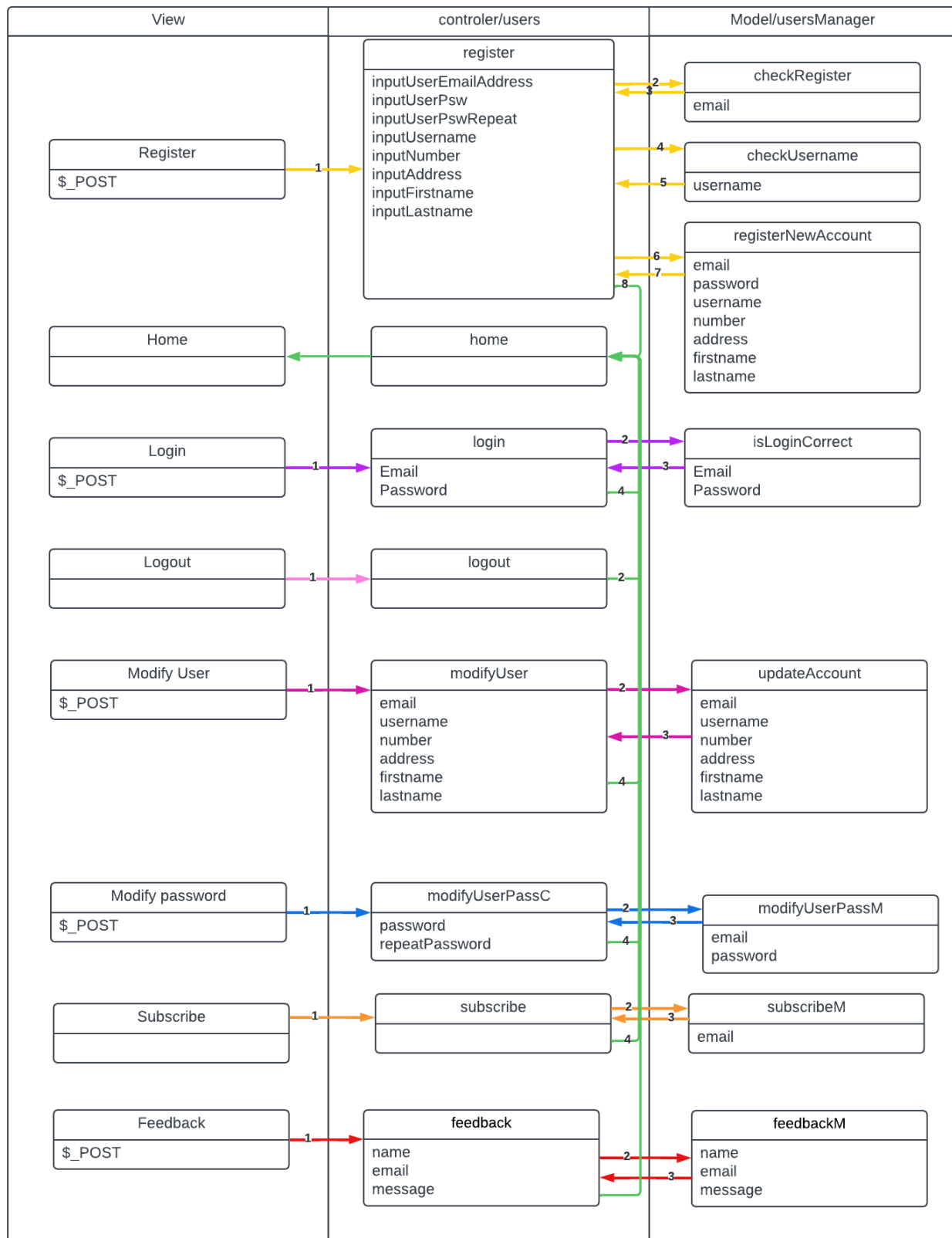
- La table **users** représente les utilisateurs et possède une clé primaire id. Les autres tables peuvent avoir une relation avec la table **users** en utilisant cette clé primaire.
 - La table **albums** représente les albums et possède une clé étrangère **users_id** qui fait référence à la colonne id de la table **users**. Cela indique qu'un album appartient à un utilisateur spécifique.
 - La table **images** représente les images et possède une clé étrangère **albums_id** qui fait référence à la colonne **id** de la table **albums**. Cela établit une relation entre une image et son album correspondant.
 - La table **contacts** représente les feedbacks et stocke des messages donnés par les utilisateurs. Elle n'a pas de relations directes avec les autres tables dans le script fourni.
- Chaque utilisateur dans la table **users** peut avoir plusieurs albums dans la table **albums**.
 - Chaque album dans la table **albums** peut avoir plusieurs images dans la table **images**.
 - La table **users** agit comme table parente pour la table **albums** grâce à la clé étrangère **users_id**.
 - La table **albums** agit comme table parente pour la table **images** grâce à la clé étrangère **albums_id**.

2.5.4 Concept de code

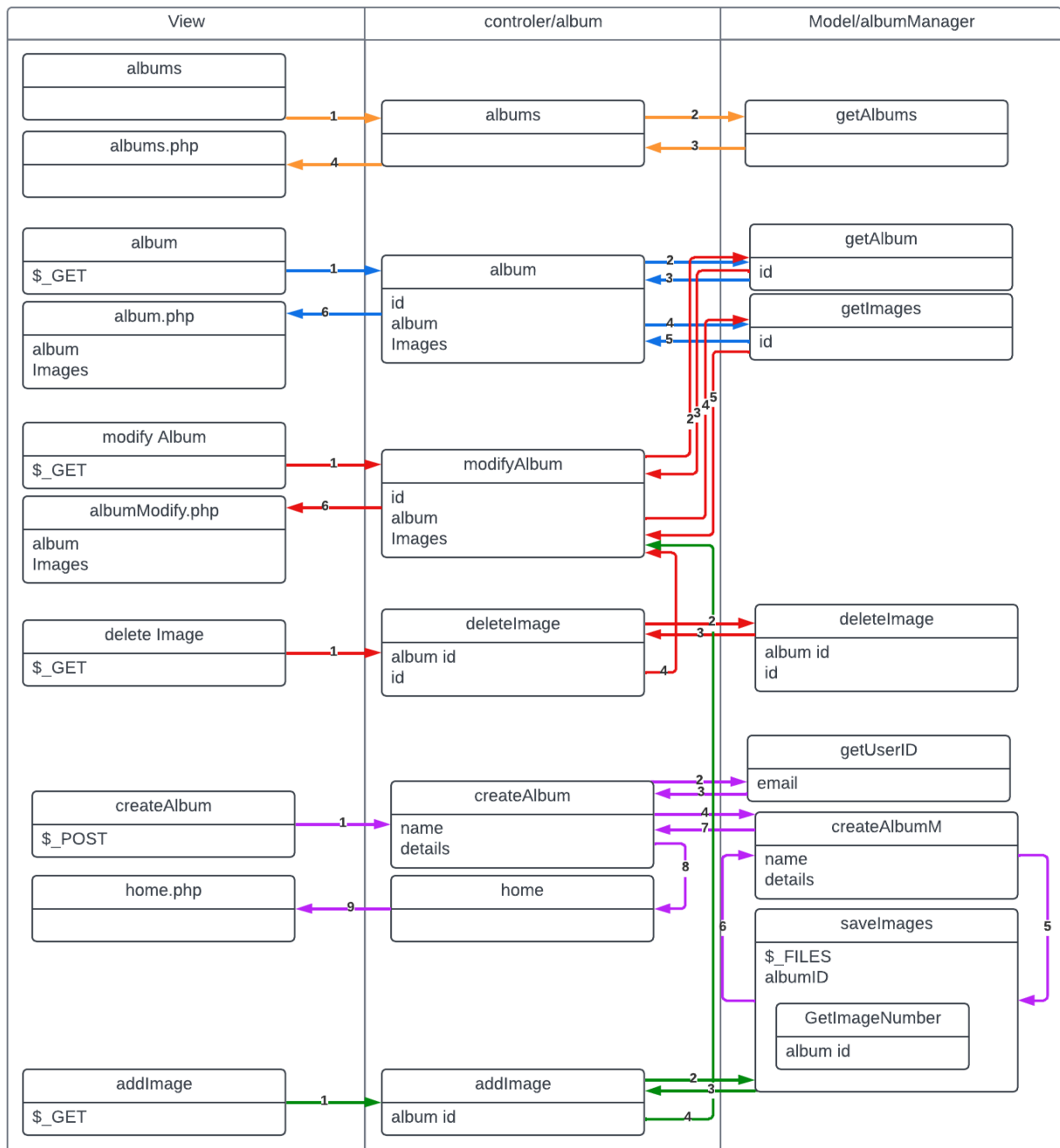
2.5.4.1 Structure de fichiers



2.5.4.2 Structure d'user et contact



2.5.4.3 Structure d'album

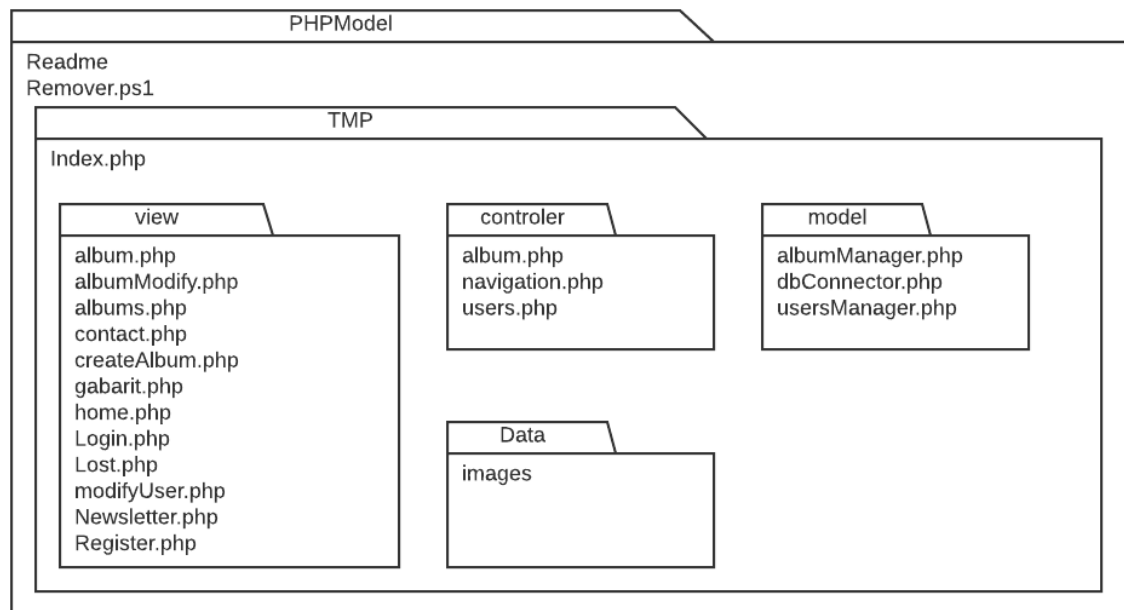


Chaque couleur représente une action de l'utilisateur.
Chaque "fonction" dans les vues est en fait un cas de commutation dans index.php qui reçoit des informations soit avec les variables super/globales \$_GET ou \$_POST, où il est décidé quelle fonction du contrôleur doit être appelée, donc le modèle est toujours [vue]>[index]>[contrôleur]>[modèle]>[contrôleur]>[vue].

Parfois, le contrôleur n'a pas besoin d'appeler une fonction du modèle et parfois il doit l'appeler plusieurs fois.

3 Réalisation

3.1 Dossier de réalisation



3.1.1 **Controler de users**

3.1.1.1 register

Une fonction pour vérifier l'intégrité de la demande d'enregistrement transmise par \$_POST par l'utilisateur

3.1.1.2 Login

Cette fonction vérifie si l'utilisateur a fourni une adresse électronique et un mot de passe pour se connecter. Si c'est le cas, la fonction vérifie les informations d'identification et connecte l'utilisateur ou affiche un message d'erreur. Si les informations de connexion sont manquantes, il affiche la page de connexion pour que l'utilisateur puisse les compléter. Cela signifie qu'il faut définir \$_SESSION

3.1.1.3 Logout

Supprime \$_SESSION

3.1.1.4 ModifyUserPassC

Cette fonction vérifie si l'utilisateur a fourni son adresse électronique, un nouveau mot de passe et un mot de passe répété pour modifier son mot de passe. Il s'assure que les mots de passe correspondent et tente ensuite de modifier le mot de passe de l'utilisateur. Si la modification est réussie, il redirige l'utilisateur vers la page d'accueil. En cas d'erreur, il affiche un message d'erreur. Si l'information requise est manquante, il affiche un formulaire permettant à l'utilisateur de saisir les détails nécessaires.

3.1.1.5 ModifyUser

Cette fonction vérifie si l'utilisateur a fourni une adresse électronique pour modifier son compte. Il vérifie si l'adresse électronique et le nom d'utilisateur sont uniques dans la base de données du système. Si c'est le cas, il met à jour les informations du compte de l'utilisateur et le redirige vers la page d'accueil. En cas d'erreur au cours du processus, un message d'erreur approprié s'affiche.

3.1.1.6 Newsletter

Cette fonction effectue une opération d'abonnement pour l'utilisateur associé à l'adresse électronique stockée dans la session, puis affiche la page d'accueil.

3.1.1.7 Feedback

Traite et enregistre les commentaires de l'utilisateur en utilisant le nom, l'adresse électronique et le message fournis, puis affiche la page d'accueil.

3.1.2 **Controller de albums**

3.1.2.1 addImage

Cette fonction récupère l'identifiant associé à l'adresse électronique de l'utilisateur actuel, enregistre les images téléchargées dans l'album spécifié et effectue toutes les modifications ou mises à jour nécessaires concernant l'album.

3.1.2.2 deletelImage

Cette fonction supprime l'image avec l'ID spécifié, puis effectue toutes les modifications ou mises à jour nécessaires relatives à l'album à partir duquel l'image a été supprimée.

3.1.2.3 modifyAlbum

Cette fonction récupère les détails et les images de l'album, puis affiche une vue appelée "albumModify.php" qui permet de modifier l'album.

3.1.2.4 album

Cette fonction récupère les détails et les images de l'album, puis affiche une vue appelée "album.php" qui permet de visualiser l'album.

3.1.2.5 albums

Cette fonction récupère des informations sur les albums, puis affiche une vue appelée "albums.php" qui montre les informations sur les albums, comme une liste d'albums.

3.1.2.6 createAlbum

Cette fonction tente de créer un nouvel album en utilisant les données fournies, si elles existent. Si les données ne sont pas disponibles, il affiche une vue permettant de créer un nouvel album. Si une exception se produit, elle ne sera pas détectée et traitée dans cet extrait de code. Enfin, il redirige l'utilisateur vers la page d'accueil.

3.1.3 Model

3.1.3.1 Fonction typique du modèle

```
function getUserID($userEmailAddress)
{
    $sep = '\\';
    $userTypeQuery = 'SELECT id FROM users WHERE email=' . $sep . $userEmailAddress . $sep . "\"";

    require_once "model/dbConnector.php";
    $queryResult = executeQuerySelect($userTypeQuery);

    return $queryResult[0];
}
```

La fonction type du modèle reçoit des informations, puis, à l'aide de ces informations, effectue un SELECT/DELETE/UPDATE/INSERT à partir de la base de données et renvoie les résultats.

3.1.3.2 registerNewAccount

Cette fonction prend les informations du compte de l'utilisateur, hache le mot de passe pour des raisons de sécurité, prépare une requête SQL pour insérer le mot de passe haché et d'autres informations dans la base de données, exécute la requête et renvoie un résultat indiquant le succès ou l'échec de la procédure d'enregistrement.

3.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- *les conditions exactes de chaque test*
- *les preuves de test (papier ou fichier)*
- *tests sans preuve: fournir au moins une description*

3.3 Erreurs restantes

S'il reste encore des erreurs:

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

3.4 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

4 Conclusions

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

5.2 Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Archives du projet

Media, ... dans une fourre en plastique