**Présentation**

**Pour répondre aux exigences de l'examen, je présente 2 projets.**

**Projet pour la partie front-end:**

Développement d’un site vitrine simple mais avec un design adaptatif qui doit reprendre les cordonnées de chaque société et les afficher sur la page.Ce site permet aux utilisateurs de pouvoir contacter rapidement l’entreprise ciblée.

**Projet pour la partie back-end:**

Développement d’un site qui simplifie le processus du règlement des factures par chèque.

Le site permet d’enregistrer les factures , imprimer le(s) chèque(s) , avoir le statut , la date de la signature et de l’expédition ainsi que la possibilité d’une description si celui-ci est finalement annulé.

**Compétences acquises dans ces projets**

* Maquettage de l’application
* Réalisation d’une interface utilisateur web statique et adaptable,
* Développer une interface utilisateur web dynamique,
* Création de la base de données
* Développer les composants d’accès au données
* Développer la partie backens d’une application web et web mobile.

**Les technologies**

**Les outils utilisés**



**Balsamiq**, est un logiciel de conception de Wireframes qui permet aux équipes de

créer des maquettes et des prototype interactifs, mais aussi de réaliser des tests

utilisateurs.

 **Visual Studio Code,** également communément appelé **VS Code,** est un éditeur de

code source créé par Microsoft avec le framework Electron, pour Windows, Linux et

macOS



**phpMyAdmin**, est un application web de gestion pour les systèmes de gestion de

base de données MySql et MariaDB .Elle est réaliséenprincipalement en PHP.



**WampServer**, est une plateforme de dévoloppement Web sur Windows qui permet

de créer des applications Web dynamique avec apache2, PHP, MySQL et MariaDB.

**Les langages utilisés**

****

**HTML**, HyperText Markup Language, est un langage qui permet de mettre en forme

du contenu. Les balises permettent de mettre en form le text et de placer des

élémments interactifs, tel des liens, des images ou bien encore des animations.

 **CSS**, Cascading Style Sheets, est un langage qui permet de gérer la présentation

d’une page web. Les styles permettent de définir des règles appliquées à un ou

plusieurs documents HTML.Ces règles portent sur le positionnement des éléments,

l’alignement, les polices de caractères, les couleurs, etc.

 **Javascrip**t, est un langage de scripts principalement employé dans les pages web

interactives et à ce titre est une partie essentielle des applications web . Il est la

3ème couche technologis standards du web, les deux premières étant les langages

Html et CSS.



**PHP**,Hypertext Preprocessor, est un langage de scripts généraliste et open source.

Il est spécialement conçu pour le développement d'applications web. Il peut être

facilement intégré au HTML. C'est un langage orienté objet .

**Les freamworks utilisés**

****

**Bootstrap**, est un framework open source de développement web orienté interface

graphique. Utilisant les langages HTML, CSS et JS, il est pensé pour développer des sites avec un design responsible.



**Jquery**, est un bibliothèque Javascript, open source, qui permet aux développeurs

d’ajouter des fonctionnalités supplémentaires aux sites web.

**Projet pour la partie front-end**

**Cahier des charges :**

La société Etoile Secours, basée à Saint-Palais-Sur-Mer,spécialisées dans la gestion de 80 entreprises d’ambulances ,les pompes funèbres et immobilier , souhaite a réaliser d’un site vitrine simple mais avec un design adaptatif qui doit reprendre les cordonnées de chaque société et les afficher sur la page.Ce site permet aux utilisateurs de pouvoir contacter rapidement l’entreprise ciblée.

* les fonctionnalités au minimum
* simplifier le contenu au maximum
* pas de création de compte
* personnaliser l’interface
* optimiser pour le référencement (ASO)

***La structure du site doit répondre au format suivant:***

La racine du répertoire doit reprendre le nom de l’entreprise. Ce nom doit être analysé pour être retrouvé dans la base de donnée et retrouver l’adresse le nom, les coordonnées GPS de celle ci si l’entreprise à été retrouvé dans la base de donnée.

La version mobile simple sociales accès rapide au téléphone avec une zone importante concernant l’accessibilité et facilement cliquable par le pouce droit, sur 2 éléments important, une zone pour déclencher un appel à l’aide d’un lien et une deuxième zone pour déclencher une navigation par GPS à l’aide d’un deuxième lien .

***Les pages du site:***

* Accueil
* Emplois
* Véhicules
* R.D.V
* Connexion

Etoile Secours m’a demandé de faire uniquement la page Accueil qui est plus important car les 83% ds visiteurs le site appellent ,les 15% utilisent le GPS et le reste 2% ballade sur le site et la page de Connexion.

**Maquettage:**

Pour la partie du maquettage, les couleurs et les polices ansi que le logo étaient fourni par la société.

Les couleurs utilisées sont:



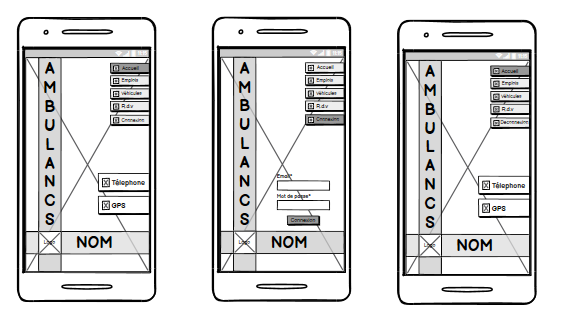
#005b8c #92a6b1 #fff

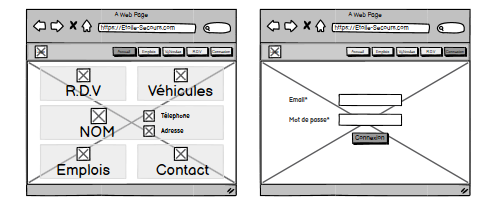
Les polices utilisées sont:

H1 à H4, nom de société, **Okay BQ**

H5, H6, P et autres, **Helvetica**

J’ai donc créée une maquette sous Balsamiq présentant une page d’accueil et une page connexion au format mobile et desktop.





Après avoir analysé les maquettes et après d’une discussion intern, ils m’ont accepté la version mobile mais pour la version desktop m’ont fourni une version design graphique.

C’est une site monopage , avec quatre sous-pages,

* ACCUEIL
* SERVICES
* L’AGENCE
* CONTACT

avec une aside gauche qui est fixe pour la page, ou on retrouve le nom du société , les liens vers sous-pages et un footer.

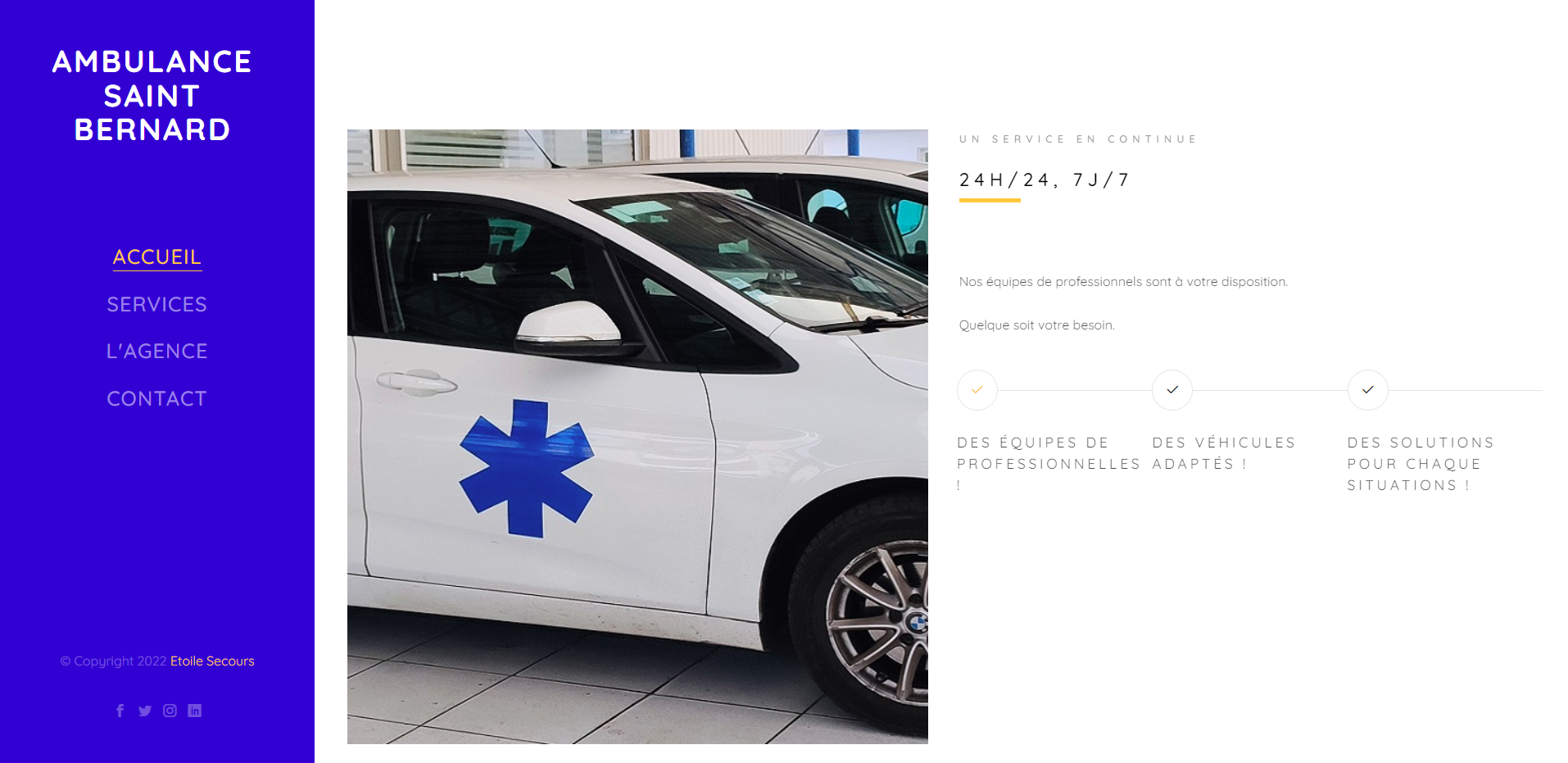
*Contenue d’accueil*



On voit bien que la sous-page l’accueil contient un carrousel avec un pop-up.

Les nom des images et le texts des pop-up on recouper de la base de données avec un hauteur 100vh.

Contenue services

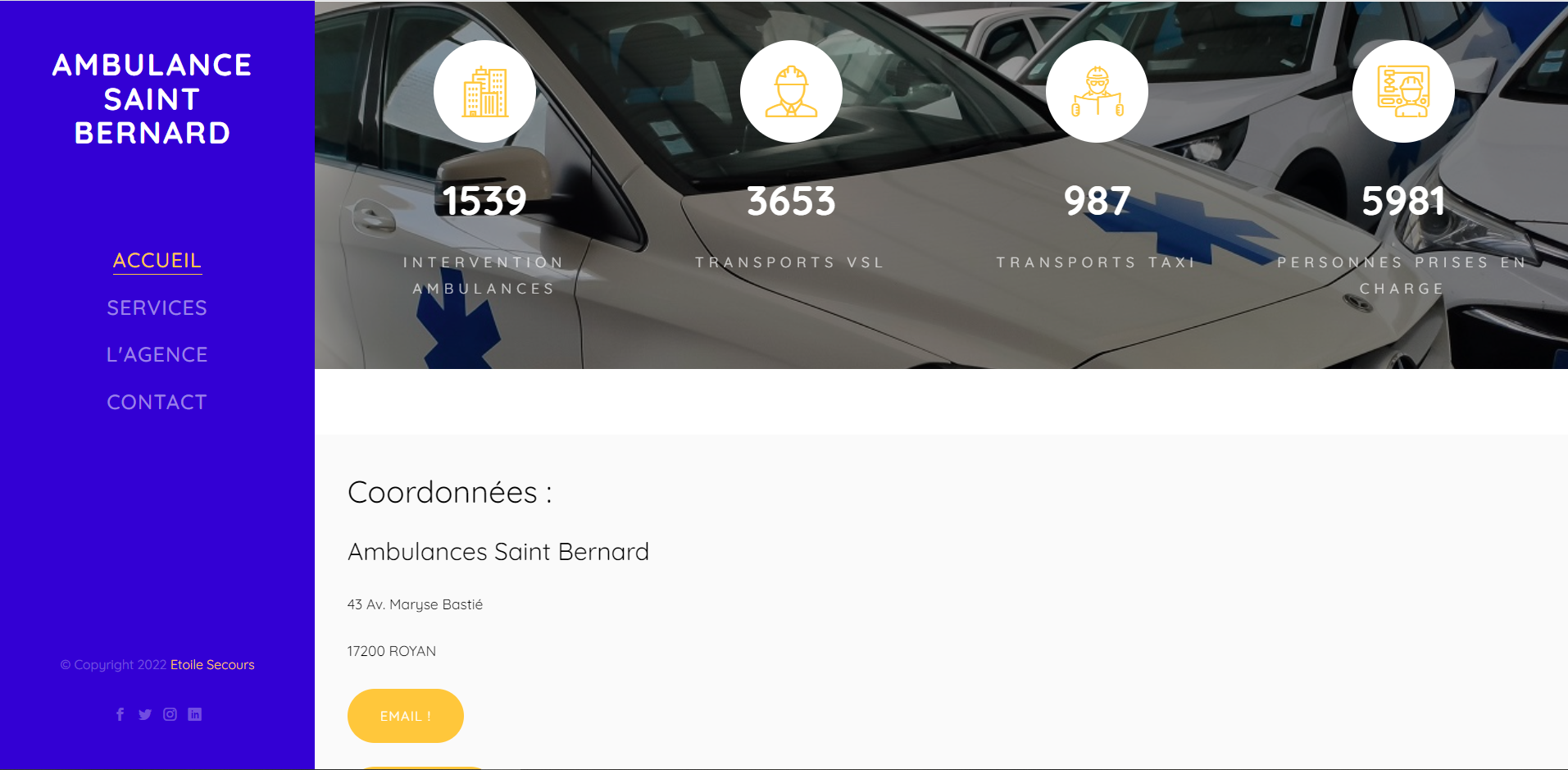


On voit bien que la sous-page l’accueil contient un div avec une image ainsi que des texts qu’on recouper de la base de données avec un hauteur 100vh.

Contenue pour l’agence et contact:

La contenue pour la sous-pages l’agence et contact couvert une hauteur de l’ecran.

La sous-page l’agence prend la partie haut avec les chiffres correspondent les ambulances, les transports VSL, les transports taxi et les personnes prises en charge qui s’affichent avec un animation de zéro justqu’a le nombre. Egalement ya une \*\*\*\*\*\*\*\*\*\*\*.

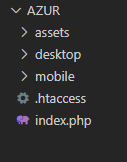


Dans la contenue contact on vois le nom de la société , l’adresse, le code postale et la ville.

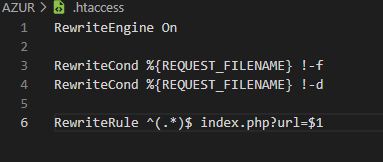
C’est demandé aussi ajouter une map avec les cordonées de la societes.

***Tout les informations on recouper de la base de données de la société.***

**La structure:**

* .htacces-permet de définir des régles
* index.php-switch la version mobile ou desktop
* desktop-tout les repertoirs et les fichies de la version desktop
* mobile-out les repertoirs et les fichies de la version mobile
* assets-pour les fremzworks et les images de la societe

Dans le ficier .htacces

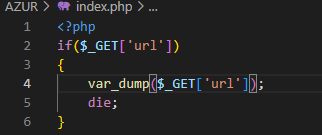


J’ai reecrit une roule en présisant que le URL de site remplace par index.php et le reste passe a index.php comme URL.



Ici on voit que le URL est accueil, donc avec .access ca devient index.php?accueil

Maintient on fait un petit test pour voir la fonctionnalité de la roule que j’ai intègre

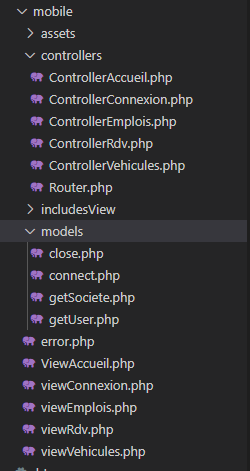
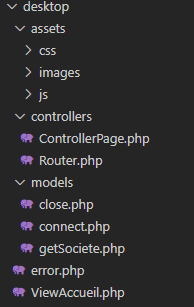


Après avoir afficher le résultat, on voit bien que ça nous donne le résultat attendu.

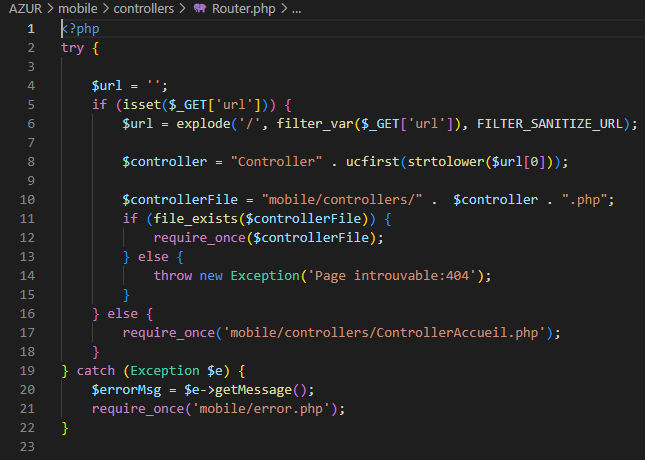


Dans la fichie index.php j’ai utilisé $\_SERVER[‘HTTP\_USER\_AGENT’] pour pouvoir détecter le divace on includ Router.php correspondent.

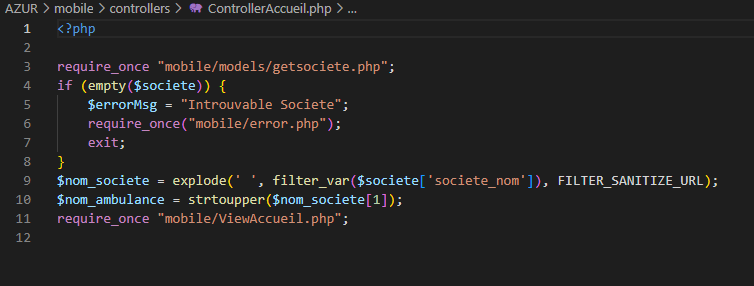
L’architecture pour la version mobile et desktop a ete fait avec la pattern disine MVC en PHP procedurale.L’arcitecture du site pour la version mobile et desktop reste le méme.



Dans la fichier Router.php on verify si Get[‘url’] exicte, si oui on explode le url, filtrer et puis en voir dans la dossie controller existe une fiche pour cette url et integrer sinon throw une nouvelle exception comme quoi page est introuvable, et si y’a pas de url on integre le fichier controller pour le page d’accueil(ControllerAccueil.php)

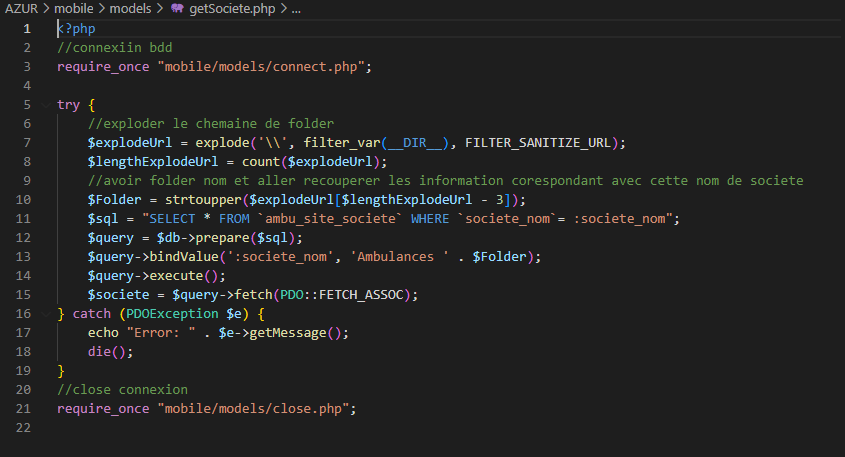


Dans la fichier Router.php on verify si Get[‘url’] exicte, si oui on explode le url, filtrer et puis en voir dans la dossie controller existe une fiche pour cette url et integrer sinon throw une nouvelle exception comme quoi page est introuvable, et si y’a pas de url on integre le fichier controller pour le page d’accueil(ControllerAccueil.php)



dans notre ControllerAccueil.php on avait besoin recouper la société, donc on inclue

mobile/models/getsociete.php

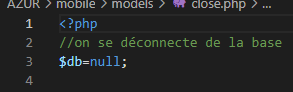
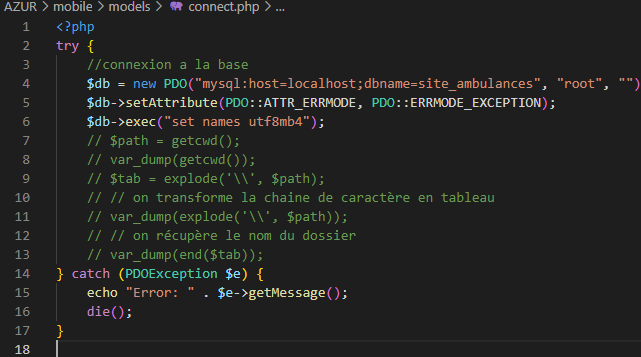


Dans getSociete.php on inclue connect.php pour faire une connexion a la data base, on localise le dossier ou on est actuellement ainsi que le nom de dossier , et recouper les

données qui correspond avec cet nom qui est également la nom de la société.

Une foir qu’on recouper la data, on incluse close.php pour faire la déconnexion de la data base.

connect.php close.php



Dans Controlleraccueil.php , apres avoir recouperer les datas rencaisser , on incluse viewAccueil.php.

Le principat reste pareille pour les autres urls.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Les views**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Pour la version mobile , j’ai separre une partie qui est en generale pour tout les views.



Dans mes code, c’est présente la le fichier templete.php et footer.php qui sont dans le dossier includesView



**Projet pour la partie back-end**

**Cahier des charges :**

Le deuxième projet c’est la développement d’un site responsible qui simplifie le processus du règlement des factures par chèque.Le site permet d’enregistrer les factures , imprimer le(s) chèque(s) , avoir le statut , la date de la signature et de l’expédition ainsi que la possibilité d’une description si celui-ci est finalement annulé.

***Les pages du site:***

* Accueil
* Impression
* Signature
* Expédition
* Historique

***La structure du site doit répondre au format suivant:***

Dans la page d’accueil possibilité de création une nouvelle facture et dynamiquement voire l’affichage sur le chèque dans les endroits correspondants, possibilité enregistrement la facture ou impression le chèque en enregistrement la facture parallèlement.

Dans la page d’impression l’y a la liste des factures qui sont en attends a impression en chèque., Ainsi que possibilité de faire recherche dans la liste soit par numéro du facture soit par société et encore possibilité faire modifier le(s) facture(s) , supprimer et imprimer plusieurs factures d’un coup .

Dans la page de signature l’y a la liste des chèques déjà imprimés, qui sont en attends a signer.Pour pouvoir signer la facture on doit mettre numéro du facture(s). Ainsi que possibilité de faire recherche dans la liste soit par numéro du facture soit par société.

Dans la page d’expédition on a la listes des chèque déjà signé , qui attendent a expédier.

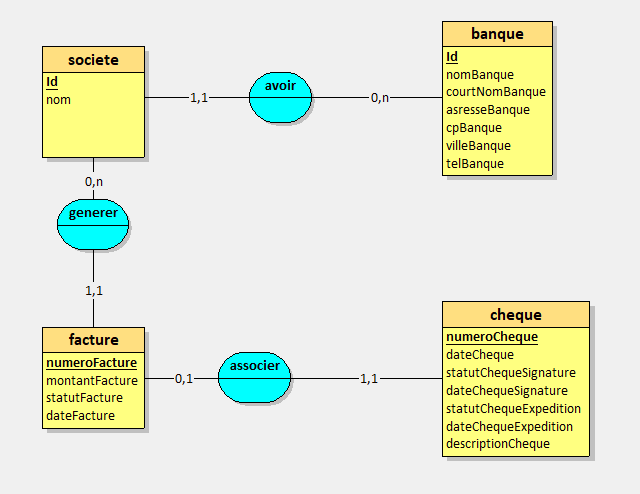
On a possibilité dans cette page annuler ou expédie la facture, ainsi que expédition plusieurs d’un coup.Encore possibilité de faire recherche dans la liste soit par numéro du facture soit par nom du société soit numéro du chèque. Si jamais on annule le chèque, on a possibilité de mettre la motive d’annulation.

Dans la page historique, on voir la liste des facture complets qui sont soit déjà envoyés soit annulés, Ainsi que voir la motive d’annulation si le chèque a été annulé.Encore possibilité de faire recherche dans la liste soit par numéro du facture soit par nom du société soit numéro du chèque.

***Pour la partie du maquettage, les couleurs et la police étaient libres d’utilisation, ainsi que les images.***

**Création la base de données**

Pour concevoir une base de données dans sont ensemble il faut dans un première temps la modéliser.J’ai utilisé la logiciel Looping pour faire un MCD(Modèle De Conceptuel De Donnée).



Donc j’ai crée quartes tables , qu’ils sont *banque*, *société*, *facture*, *chèque* pour gérer mon site.

***Normalisation dans la base de données***

Pour éviter les anomalies dans la base de données, il faut que notre nos tables seront normalises et pour ça y’a trois règles a respecter

**a. 1FN**

*Une base de données est en première forme normale* ***1FN*** *si elle satisfait les conditions suivantes:*

* *Ne contient que des valeurs atomiques*
* *Il n'y a pas de groupes répétés*

**b. 2FN**

Une base de données est en première forme normale ***2FN*** si elle satisfait les conditions suivantes:

* **1FN**
* Chaque attribut de clé non primaire dépend entièrement fonctionnellement de la clé primaire.

***On dit attribut B dépend fonctionnellement de l’attribut A, quand la connaissance de la value de A implique la connaissance de la valeur de B.***

***On note A->B***

**b. 3FN**

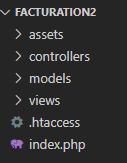
Une base de données est en première forme normale **3*FN*** si elle satisfait les conditions suivantes:

* **2FN**
* il n’y a pas de dépendance transitive pour les attributs nom premiers.

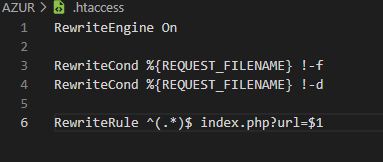
J’ai cré la base de données en respectent les règles la normalisation. Ça nous assure que on n’a pas des anomalies dans la base de données.

**La structure:**

J’ai fait mon site avec le design pattern MVC et en PHP Orienté Objet.

.

**.htaccess**(l’abréviation de Hypertext Access) désigne un type de fichier utilisé par le logiciel libre [Apache HTTP Server](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203241-apache-definition/). Ce dernier fait tourner plus de la moitié des serveurs d'Internet. Ces fichiers permettent de personnaliser la configuration d'Apache principalement pour gérer la confidentialité et la sécurité.

Dans le fichier .htacces , j’ai écris une roule en précisant que pour n’importe quel URL , il passe en premier index.php en passent l’URL.

Dans le fichier index.php je incluse Router.php pour pouvoire créer new Router class et accès son routeReq() méthode, ansi que je define une URL, qui montre la directoire actuelle.

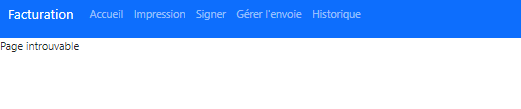


Maintenant on regarde le fichier Router.php et voir son rôle.

Class Router contiens deux attribues prives $\_ctrl, $\_view et une méthode routeReq(), qu’on l’a appelée dans notre index.php.Une foir qu’on appele la methode routeRoq(), il essaye incluede tout les fichier qui sont dans la dossier models.

Et ensuite il verifi si on l’a passe un url en GET, si oui on verifi si fichier contrôller corespand a cette URL il exit on including cette fichier sinon on arrete la programme avec an Exception comme quoi avec l’url , la page est introuvable., et si on n’a pas d’url , il charge automatiquement la controller pour la page d’accueil.

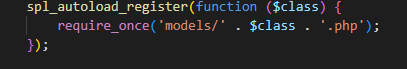
Si on essayer le donner un url inexistant , on va voire que il va dire que la page est introuvable.



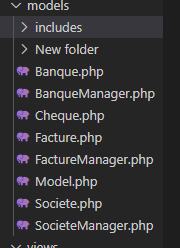
Donc ca nous assure que justqu’a ici la logique de programe est bon.

Maintenant par exemple pour un url accueil ,on voit la suite des étapes, qui sera logiquement pareille pour les autre url existant.

Donc dans notre methode routeReq(), avec la function il va charger tout les fichier qui son dans notre dossier models et qui’ls ont une classe avec la même nom que aux .



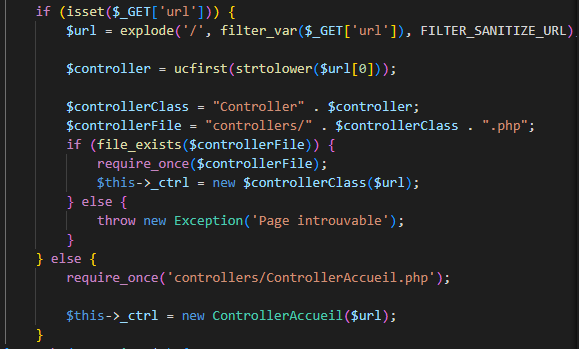
Voici les fichier dans la models



On remarque que le fichier Societe.php, Banque.php , Facture.php et Cheque;php ont les meme nom que nos tablles et a l’nterieur on la une classe avec le nome nom que la table , avec les meme paraametres ;;;;;

Ensuite la verification si url existe, va donne une resulatat positive, donc on va rentrer dans if.En verifiant que dans notre url y’a pas des script c’est qui est dans notre cas, avec la fonction expode on crer un tableau en separent le url par ‘/’, et apres on prend le premiere element.dans notre cas ,notre url est simple ‘accueil’, danc notre tableau va avoir un élément.Une fois qu’on recouper le premier element, avec strtolower() fonction on mis tout les lettres minuscule, et avec ucfirts() function, on fair le preliere lettre majuscule, dans notre cas pour un url accueil, ca sera **Accuei**l.Mais si on passe par example ***accueil/test* ,**  danc notre tableau $url va avoir deux elements , et dans ce cas la on prend toujour le premier.Alors, on peut dire même si on passe un url se separer avec ‘/’ avec plusieurs elements , il prend toujours le première élément et il fait la suite des étapes avec.

Une fois on a le url changé, on concatène par avant avec ’Controller’ , qui va nous donner le nom du class,et puis en cancatenant ‘.php’ a la fin , on recoit le nom du fichier qui est dans le dossier Controller, et Dans notre cas ca sera ControllerAccueill.php.Apres avoir definir le fichire on verifie si cette fichier exist dans la dossier Controller, si oui on fait la suite des etapes, sinon on sort le programe avec une meaage exception, comme quoi le page est introuvable.Danc notre cas ControlleurAccueil.php exist , donc ca va continuer faire les etapes souvents.On va integrer ControllerAccuel.php dans notre index.php et on va créer nouvel objet de la class ControllerAccueil.

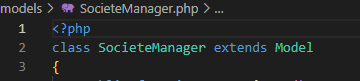


Dans notre ControllerAccueil.php , dans le première temps on inclure view/View.php, pour pouvoir access la Class View. Et puis on a une class avec le nom ControllerAccueil.la class a é attributes privet $sociteManager et $view, egalement on a un constructeur ou on apper la member function societes(), avec on recoupere les information besoin pour trannsmetre a notre view qui sera Accueil. Dans cette cas les information besoin sont les societes que on recoupere.Donc avec la membre argument societeManager on va crer une inctance de class SocieteManager et on apperl son member function getSociete() pour recouperer les societes..Apres avoir recouperer les sociétés ,avex la memebre arguments view, on va avoir une instance de class View en passant un argument Accueil, et puis avec an member function generate en passant argument array(‘societes’=>$societes) .



**Maintenant on voit les ccontenues SocietesManager et View class .**

En effet, pour chaque fichier qui ont le meme nom que notre table dans notre base de donnees societe, banque, facture, cheque on a une fichier corespande Sochiete.php, Banque.php, Facture.php, Cheque.php qui ont une class a leur interieur avec le nom corespendant Societe, Banque, Facture et Cheque.Dans la fichier models on a egalement SocitetManager, banqueManager, FactureMAnager et ChequeManager qui derige les opperations avec les instances des class coresdentant.



Chaque de fichier qui dirigent les opperation, ils hérite une abstract class Model.

**C’est quoi une class abstract.**

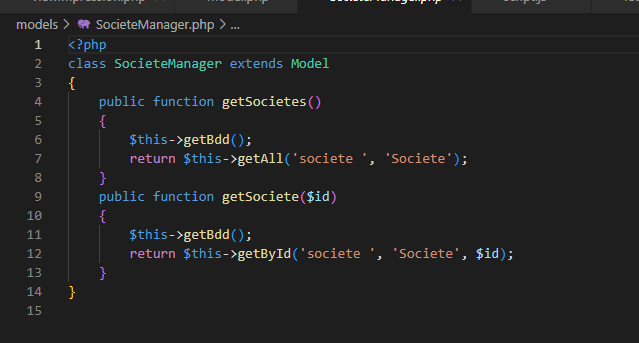
*Une classe abstraite est une classe qui contient au moins une méthode abstraite, qui est une méthode sans code réel, juste le nom et les paramètres, et qui a été marquée comme "abstraite".Le but est de fournir une sorte de modèle dont hériter et de forcer la classe héritante à implémenter les méthodes abstraites.Une classe abstraite est donc quelque chose entre une classe régulière et une interface pure. Les interfaces sont également un cas particulier de classes abstraites où toutes les méthodes sont abstraites.*

Autrement dire on peut pas avoire un inctance de class Modele, mais quand on herite la class Model, on peut faire appele ses meemer functions.Dans notre Model class on a 4 abstract class qui sont setBdd(), getBdd(), getAll()getByID().

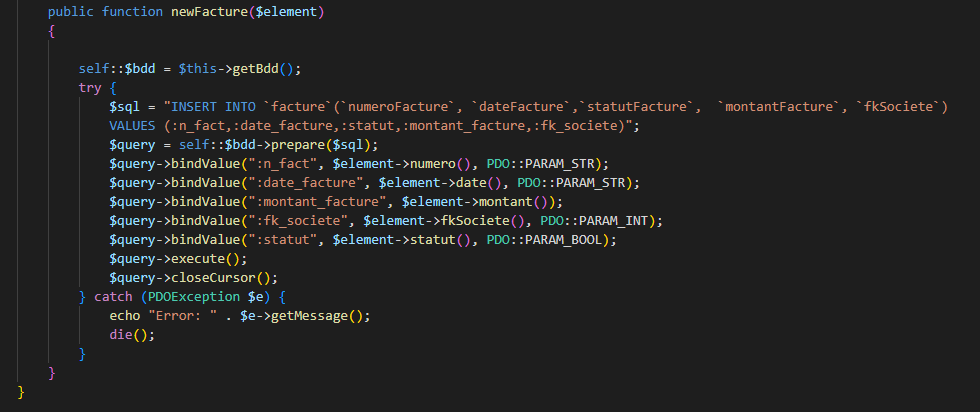
* SetBdd()-faire une connexio la base de donnes
* getBdd($table,$obj) -si il y’ a pas encore la connexion, il se connecte
* getAll($table,$obj)-return tout les objects de la table
* getById($table, $obj,$id)-returen l’objet de la table , qui id est $id

Alors, dans notre class SocieteManager.php on peut utiliser ces funnctios.

Dans la class SocieteManager , i’ya deux member function getSocietes() et getSociete()



On voit que pour les deux cas, dans les methodes il fait appele les members function de class Model.Mais la class peut avoir ca propre methode aussi, par example dans la class FactureManager, on a member function newFacture() qui crée new facture.



Donc quand on analyse tout ca , on voit qui dans la dossier models, on a les class avec les meme nom que nos table dans nos database, qui ont les class Manager corespendant qui derige les opperation avec a l’aide une abstract class Model.

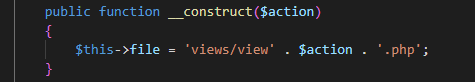
Maintenant on peut passer pour voir la contenu de View

La class View a deux memeber arguments $\_file et $\_t et trois member functions

* \_\_construct($action)
* generate(data)
* generateFile(date)

Si on revien arrier , dans notre ControllerAccueil.php , dans la membre function socitet on avait crée une inctance de View et puis fait appel son methode generate

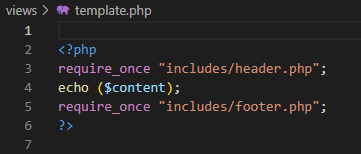
Quand on crée une instance de class , on le fait avec la construct de cette class.



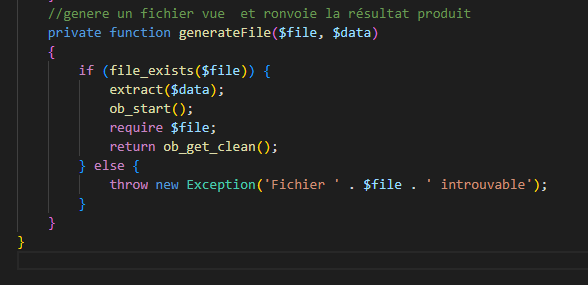
Dans le construct View, on donne value de notre membre argument **file** value la view que on veux afficher.Pour l’example accueil ca sera views/viewAccueil.php.Apres on avais fait l’appel member function generate, on passant comme argument array(’societe’=>$societes) .



La memeber function generate appele a son tour la member function generateFile une fois avec les argument class member argument file et la data qu’on la passé, c’est a dire avec views/viewAccueil.php et avec argument array(’societe’=>$societes)  et un deuxième fois avec les arguments ‘views/template.php’ et array(‘t’ => this→\_t, ‘content’=>$content) when $content c’est la resulat de première appel , this→t cest le tire notre view.le fichier template.php qui se trouve dans le dossier view , contien un incluse le hieader de page , le footer de la page et au milieux la contenue de la page.



Maintenant on voit la functionnalité de generateFile().



Pour le premier tempe on vérifie si le file existe, si oui on passe les étapes soutes, sinon on sort la programe avec une exception comme quoi la fichier est introuvale.Si le fole exist, dabord on extract la data .

**C’est quoi extraci in php**

*La fonction extract () est une****fonction intégrée en PHP. La fonction extract() effectue une conversion de tableau en variable****. C’est-à-dire qu’il convertit les clés de tableau en noms de variable et les valeurs de tableau en valeur de variable.*

*Une fois apres la extraction data, on appele php function* ***ob\_start()*** *et* ***ob\_get\_clean()****,entre ces deux* ***require $file.***

*La temporisation de sortie sera débutée par un appel à la fonction ob\_start(). La temporisation de sortie sera terminée par un appel à ob\_get\_clean().*

*En conclusion, pour construire nos view pages , pour le premiere temp on tronsver les datas naissasere , on require template , qui requir de son toure header.php et footer.php , et au milieux de ces deux on integre la contenue de la page.*

*Donc maintenatnt on peux voir les pages views et leurs fonctionnaliste .*

***viewAccueil***