**Documentation technique**

**Projet GSB – Gestion des visites**



**Projet réalisé par :** Alpha Mamadou BALDE

**Année :** *2021*

**SOMMAIRE :**

1. Présentation du projet…..………3
2. Connexion…………….………….6
3. Accueil…………………………....7
4. Profil………………………………9
5. Mes Rapports……………….......15
6. Modification d’un rapport….…....18
7. Création d’un rapport……..……..22
8. Recherche d’un médecin…..……26
9. Médecin…………………………...28
10. Déconnexion-Entête……………..32

**PRESENTATION DE PROJET**

1. **Le laboratoire Galaxy Swiss Bourdin :**

Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy (spécialisé dans le secteur des maladies virales dont le SIDA et les hépatites) et le conglomérat européen Swiss Bourdin (travaillant sur des médicaments plus conventionnels), lui-même déjà union de trois petits laboratoires.

En 2009, les deux géants pharmaceutiques ont uni leurs forces pour créer un leader de ce secteur industriel. L'entité Galaxy Swiss Bourdin Europe a établi son siège administratif à Paris.

Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux Etats-Unis.

1. **Présentation du contexte:**

### L'activité à gérer

L'activité commerciale d'un laboratoire pharmaceutique est principalement réalisée par les visiteurs médicaux. En effet, un médicament remboursé par la sécurité sociale n’est jamais vendu directement au consommateur mais prescrit au patient par son médecin.

Toute communication publicitaire sur les médicaments remboursés est d'ailleurs interdite par la loi. Il est donc important, pour l’industrie pharmaceutique, de promouvoir ses produits directement auprès des praticiens.

### Les Visiteurs Médicaux

L'activité des visiteurs médicaux consiste à visiter régulièrement les médecins généralistes, spécialistes, les services hospitaliers ainsi que les infirmiers et pharmaciens pour les tenir au courant de l’intérêt de leurs produits et des nouveautés du laboratoire.

Chaque visiteur dispose d’un portefeuille de praticiens, de sorte que le même médecin ne reçoit jamais deux visites différentes du même laboratoire.

Comme tous les commerciaux, ils travaillent par objectifs définis par la hiérarchie et reçoivent en conséquence diverses primes et avantages.

Pour affiner la définition des objectifs et l’attribution des budgets, il sera nécessaire d’informatiser les comptes rendus de visite.

### L’activité des visiteurs

L'activité est composée principalement de **visites** : réalisées auprès d’un praticien (médecin dans son cabinet, à l’hôpital, pharmacien, chef de clinique…), on souhaite en connaître la date, le motif (6 motifs sont fixés au préalable), et savoir, pour chaque visite, les médicaments présentés et le nombre d’échantillons offerts. Le bilan fourni par le visiteur (le médecin a paru convaincu ou pas, une autre visite a été planifiée…) devra aussi être enregistré.

### Les produits

Les produits distribués par le laboratoire sont des médicaments : ils sont identifiés par un numéro de produit (dépôt légal) qui correspond à un nom commercial (ce nom étant utilisé par les visiteurs et les médecins).

Comme tout médicament, un produit a des effets thérapeutiques et des contre-indications.

On connait sa composition (liste des composants et quantité) et les interactions qu'il peut avoir avec d'autres médicaments (éléments nécessaires à la présentation aux médecins).

La posologie (quantité périodique par type d’individu : adulte, jeune adulte, enfant, jeune enfant ou nourrisson) dépend de la présentation et du dosage.

Un produit relève d’une famille (antihistaminique, antidépresseur, antibiotique, …).

Lors d'une visite auprès d'un médecin, un visiteur présente un ou plusieurs produits pour lesquels il pourra laisser des échantillons.

### Les médecins

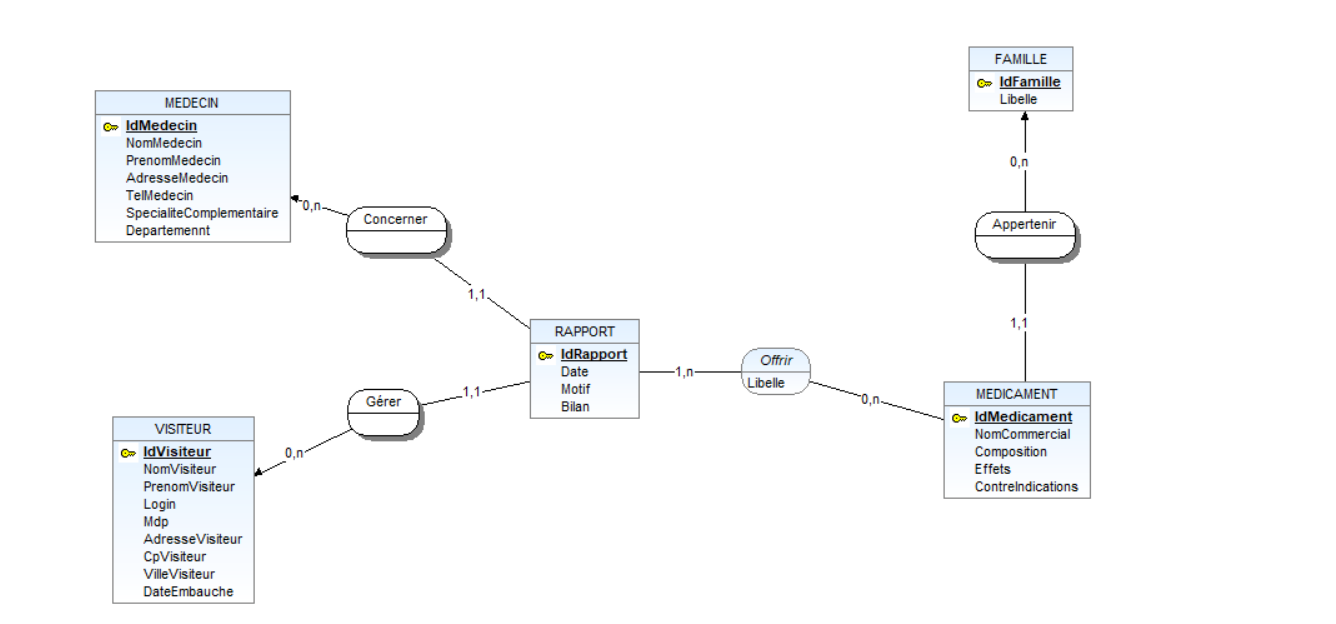
Les médecins sont le cœur de cible des laboratoires. Aussi font-ils l’objet d’une attention toute particulière.

Pour tenir à jour leurs informations, les laboratoires achètent des fichiers à des organismes spécialisés qui donnent, les diverses informations d’état civil et la spécialité complémentaire.

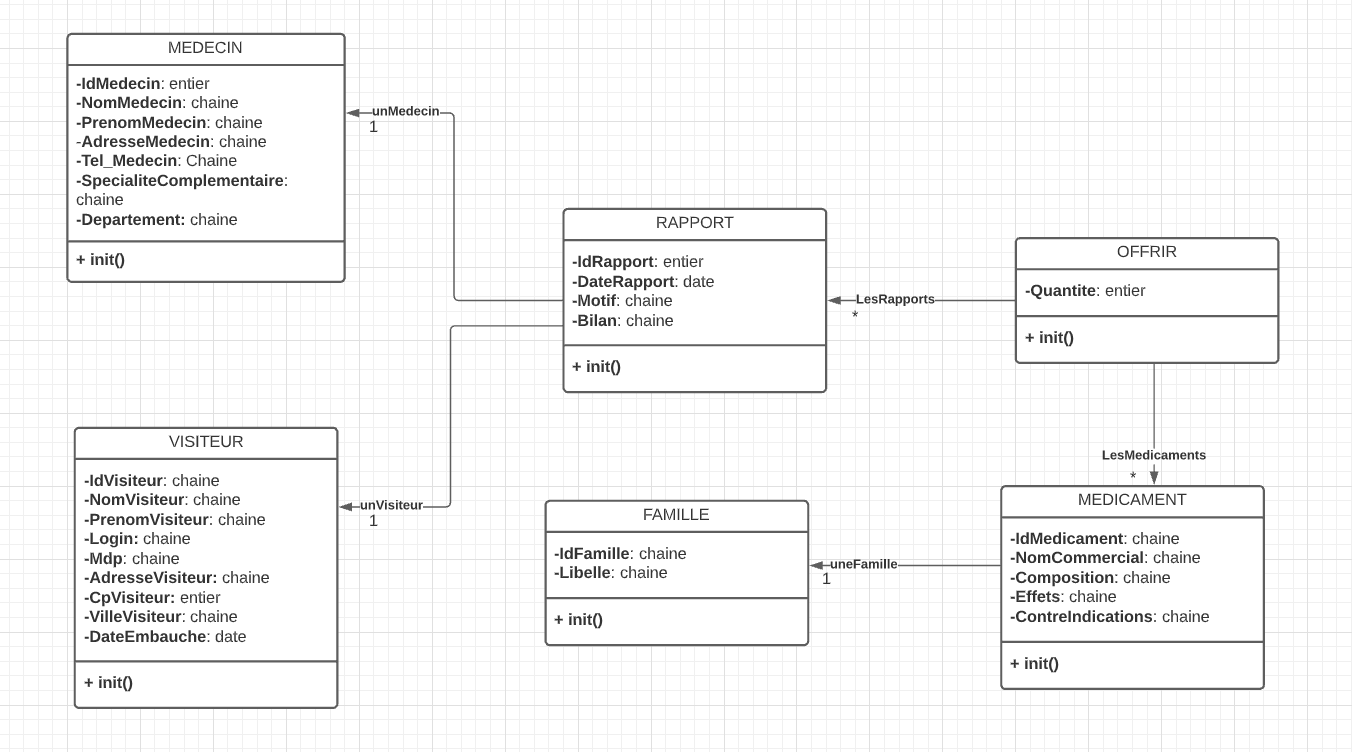
1. **L’application réalisé:**

Ainsi pour promouvoir les produits et assurer l’activité commerciale du laboratoire dans de bonnes conditions, l’application «**Gestion des visites de GSB** » a été mis en place et à la disposition des visiteurs. Cette application permet ainsi aux visiteurs de gérer les rapports de visite et les médecins. Le visiteur aura la possibilité de créer un nouveau rapport, de modifier un rapport, de voir les informations d’un médecin et de les modifier si nécessaire. Le visiteur pourra également mettre à jour son profil. Nous verrons en détail les différentes fonctionnalités de l’application dans les prochaines pages.

1. **Modèle Conceptuel de Données:**



1. **Diagramme de classe:**



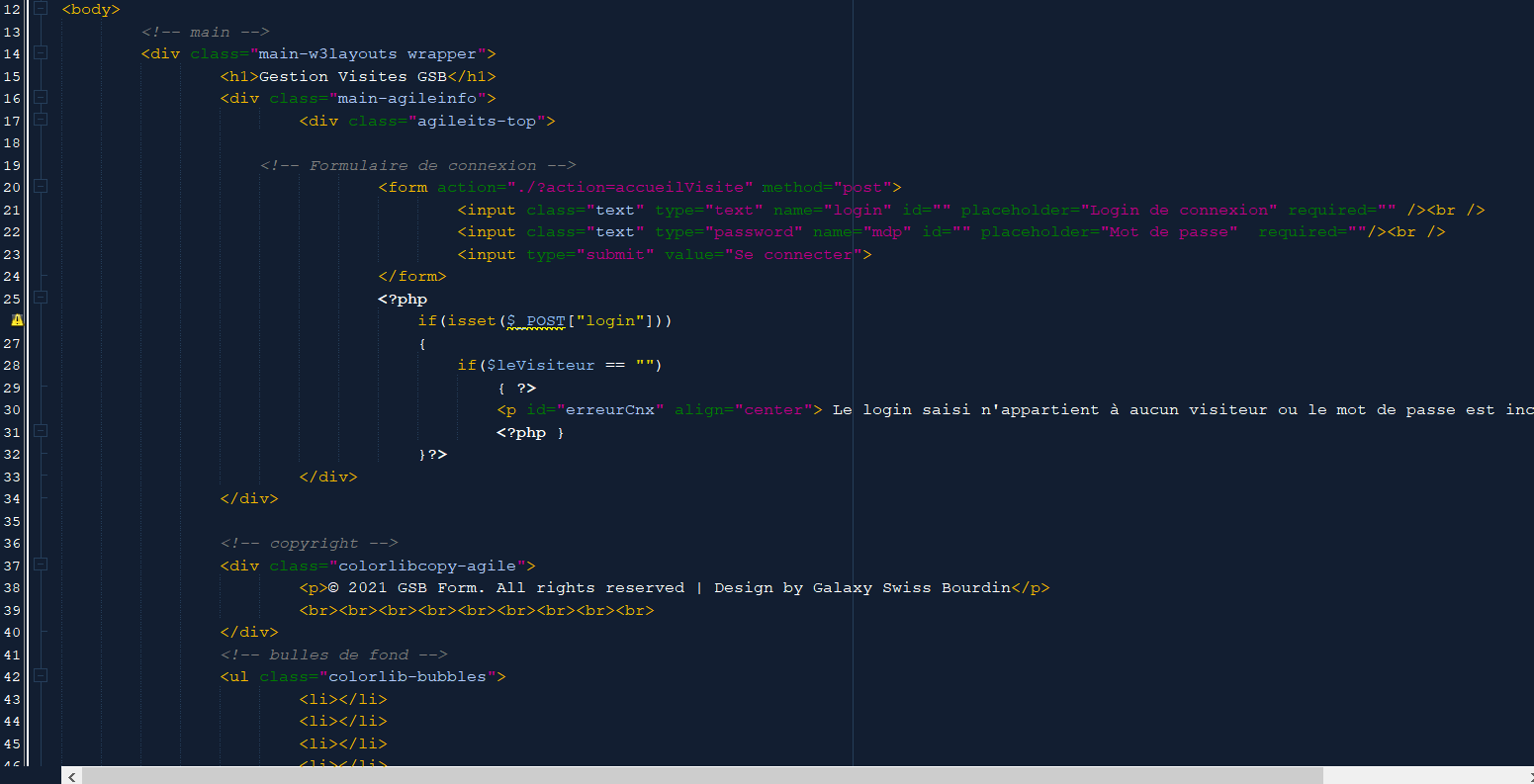
**CONNEXION**

1. **Vue:**

*Coté visiteur*



*Extrait de code (vueConnexion.php)*

****

Comme on peut le voir sur les deux images ci-dessus, pour accéder à son espace, le visiteur doit saisir ces données de connexion (Login et mot de passe). En cas de saisie incorrecte du login ou du mot de passe, le formulaire de connexion est reproposé au visiteur en lui précisant cette fois-ci que « Le login saisi n'appartient à aucun visiteur ou le mot de passe est incorrect ! ». Dans le cas où l’un des champs n’a pas été rempli, le formulaire demande au visiteur de le remplir.

1. **Contrôleur:**

Concernant le contrôleur « connexion.php » qui mène à la page de connexion, il n y a pas grandes choses à souligner. C’est le contrôleur par défaut, donc il inclut juste le fichier vue « vueConnexion.php ».

*Extrait de code (connexion.php)*

Une image contenant texte

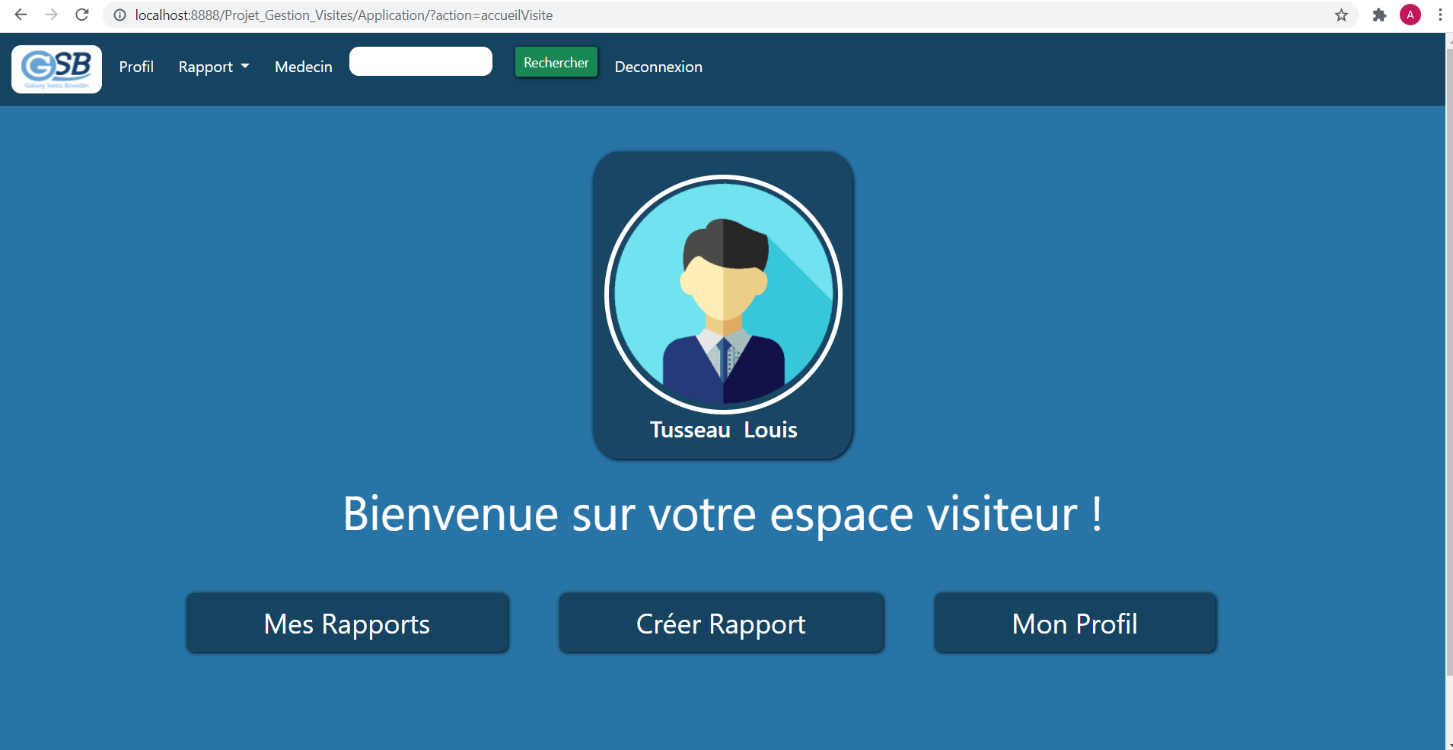
Description générée automatiquement

**ACCUEIL**

Dès la connexion du visiteur est établit, le visiteur est automatiquement dirigé vers la page d’accueil de l’application.

1. **Vue:**

*Coté visiteur*

****

Sur la page d’accueil de l’application plusieurs éléments de navigation sont mis à disposition du visiteur. A travers le menu sur la barre de navigation (entête) et le menu situé en bas de page, le visiteur peut : consulter son profil, voir ses rapports, créer un rapport, rechercher un médecin et se déconnecter.

*Extrait de code (accueilVisiteVue.php)*

Une image contenant texte

Description générée automatiquement

1. **Contrôleur:**

*Extrait de code (controleurAccueil.php)*



Au niveau de ce contrôleur, nous récupérons les données (Login et Mot de passe) saisies par l’utilisateur dans le formulaire de connexion. A partir de ces données récupérées ont charge le visiteur connecté avec la fonction getVisiteurByLogin() de la classe *visiteurDAO* pour ensuite le stocker dans la variable $leVisiteur. Dès que le visiteur a été chargé, la connexion est établit en faisant appel à la fonction login() de la classe *connexionUser.* Ainsi dès que la connexion est établit le contrôleur dirige le visiteur vers la vue de la page d’accueil (accueilVisiteVue.php).

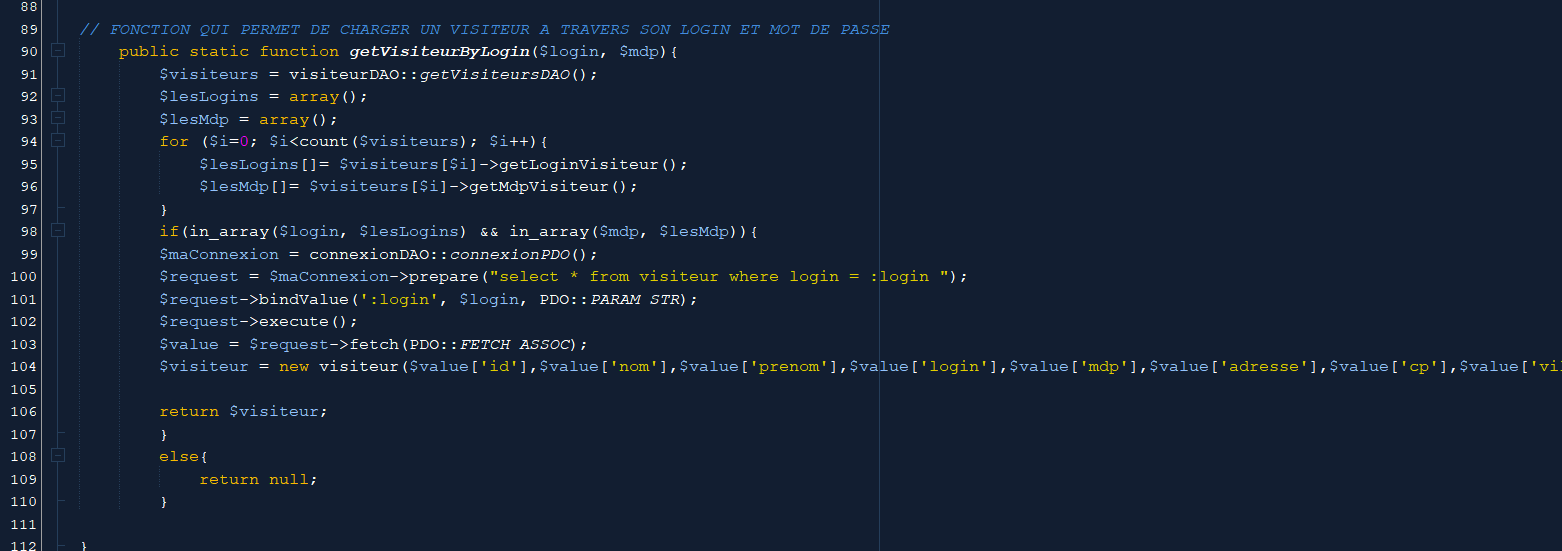
Si les données de connexion ne sont pas récupérés le contrôleur redirige le visiteur vers la page de connexion, jusqu’à ce que les bonnes données soient récupérées.

1. **Modèle:**

*Extrait de code ( fonction login() / connexionUser.php)*



*Extrait de code ( fonction getVisiteurByLogin() / visiteurDAO.php)*



**PROFIL**

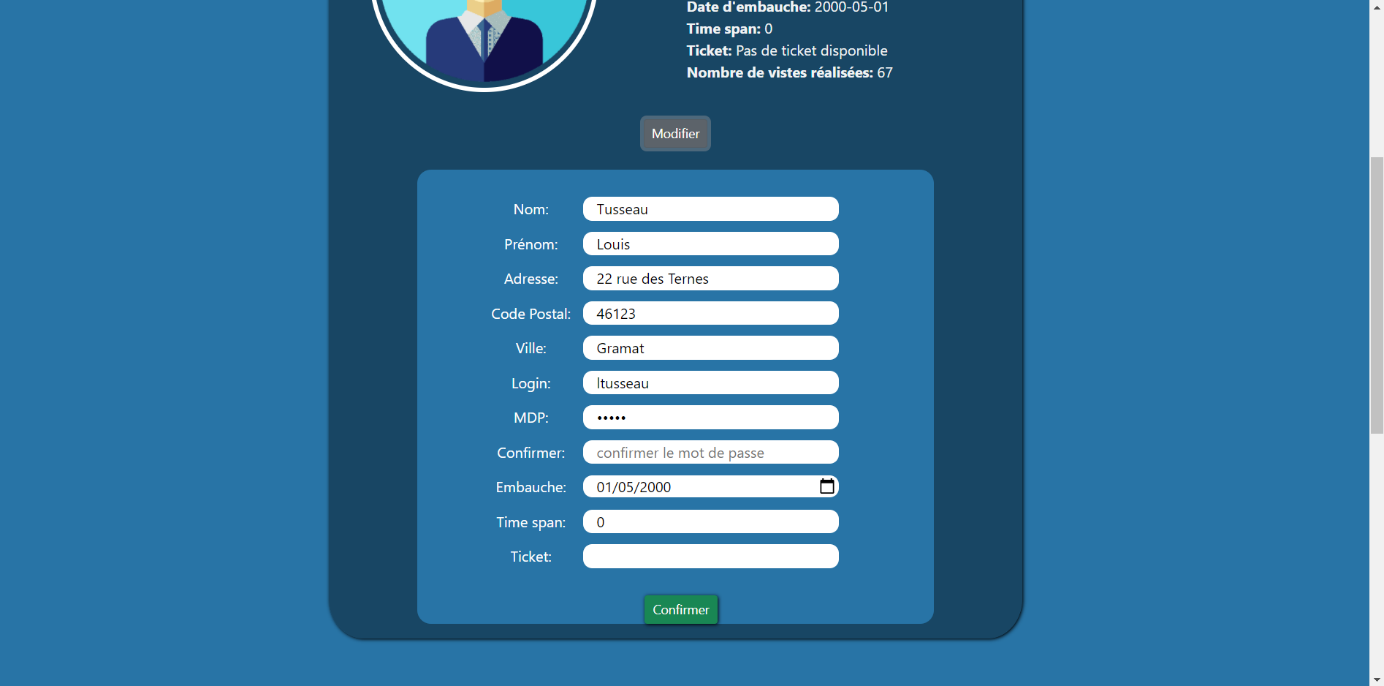
1. **Vue:**

****

****

Cette page « Profil » montre en détail les informations concernant le visiteur connecté et le rapport de sa dernière visite. Au-delà nous pouvons également remarquer la présence d’un bouton « Modifier » qui permet de mettre à jour les coordonnées du visiteur. Un clic sur ce bouton fait apparaitre automatiquement un formulaire sur lequel le visiteur peut saisir ces nouvelles coordonnées.

Vous trouverez ci-dessous une image du formulaire masqué derrière le bouton « Modifier ».



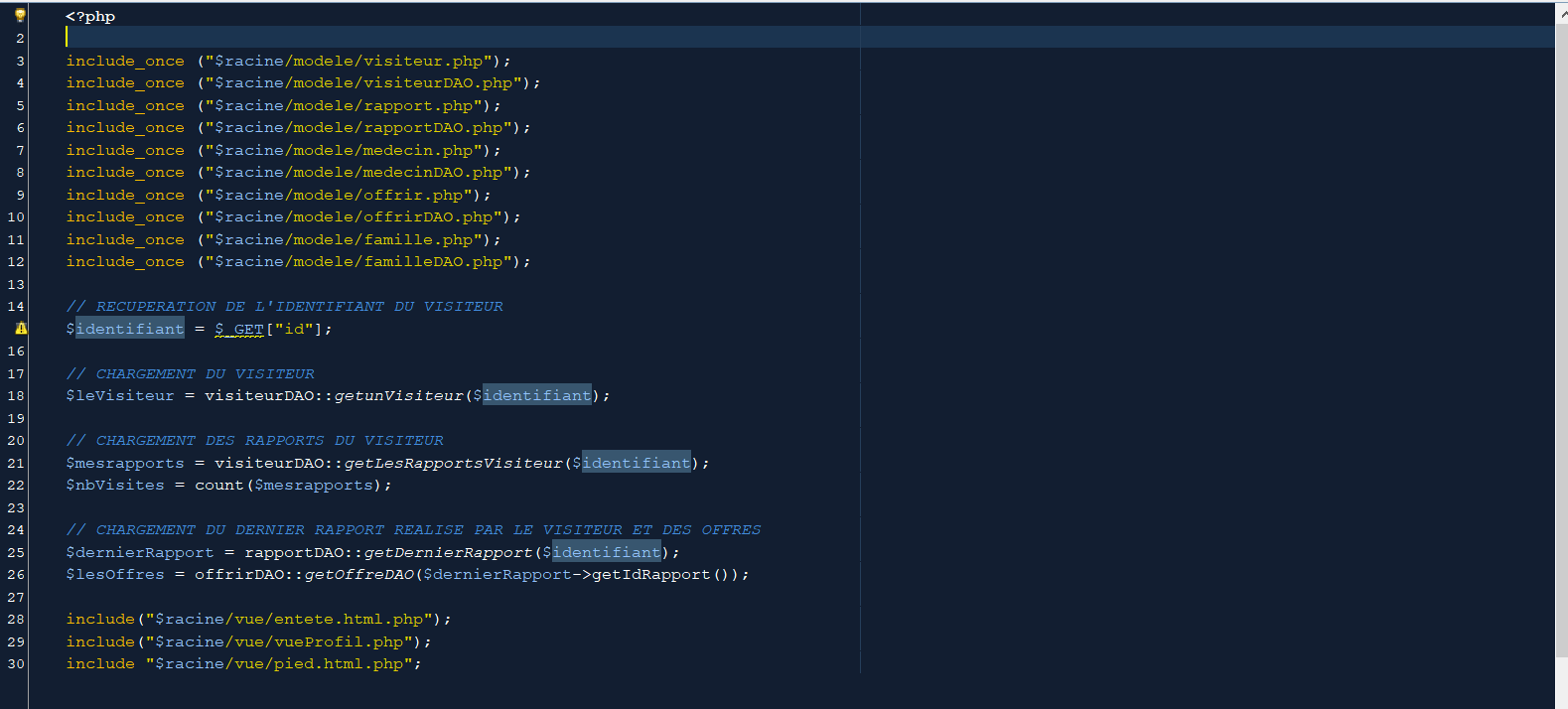
Après que le visiteur ait saisi ces coordonnées et confirmé la modification, il est aussitôt dirigé vers une page qui lui montre que la mise à jour à bien été effectué. Cette page permettra aussi au visiteur de vérifier si les données qu’il a saisi sont correctes. Le cas contraire le visiteur peut recliquer sur le buton « Modifier » pour apporter des corrections.

On peut voir cela sur l’image de la page concerné ci-dessous.



1. **Contrôleur:**

*Extrait de code (controleurProfil.php)*



Dans ce fichier « controleurProfil.php », on récupère en premier l’identifiant du visiteur qui a été transmis au contrôleur en cliquant sur l’élément de navigation « Profil ». Ensuite nous chargeons le visiteur connecté en faisant appel à la fonction getunVisiteur() de la classe *visiteurDAO* qui prend en paramètre l’identifiant du visiteur récupéré (*$identifiant*).

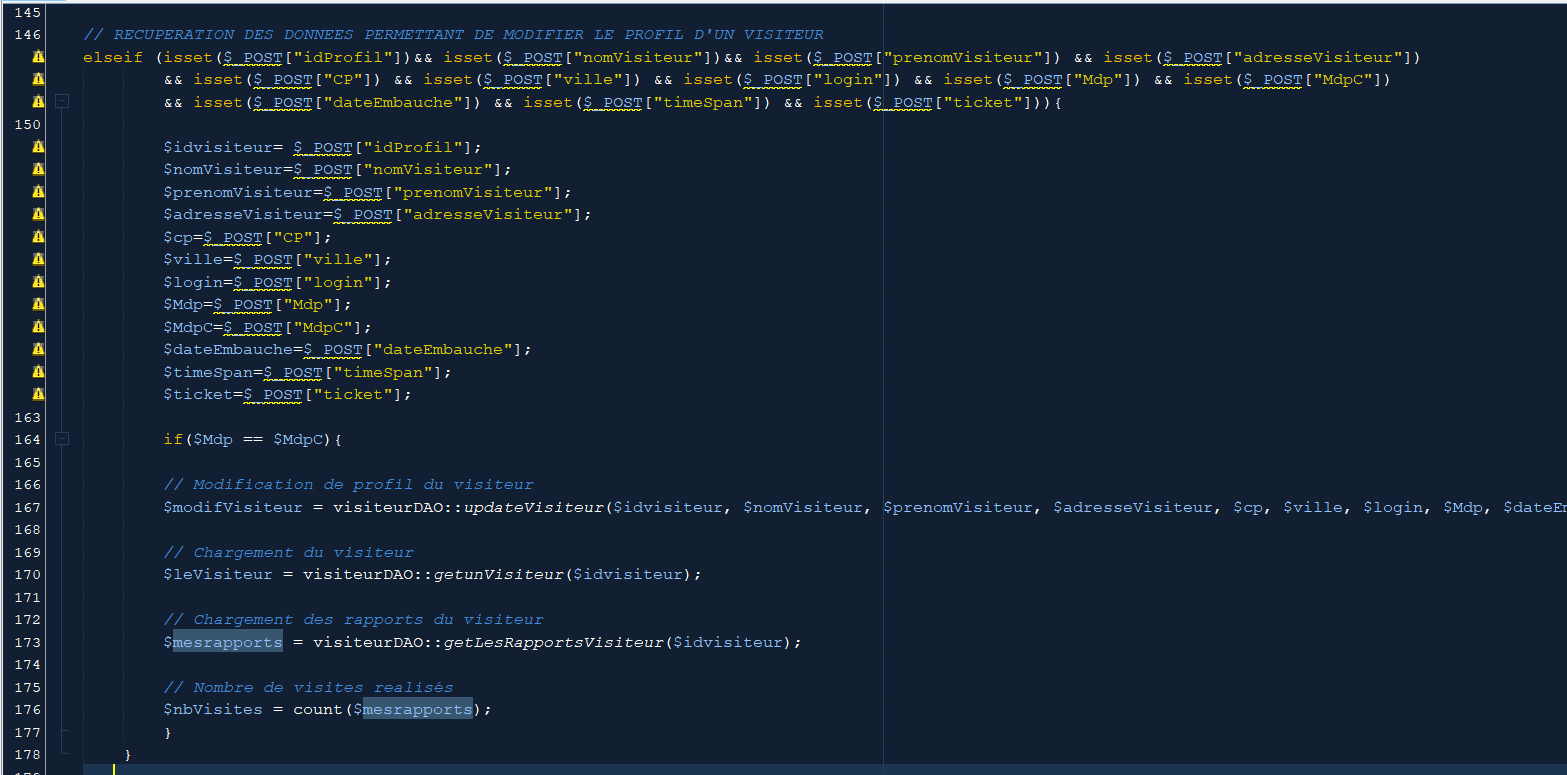
Après avoir chargé le visiteur, on charge ensuite les rapports effectués par le visiteur en faisant appel à la fonction getLesRapportsVisiteur() de la classe visiteurDAO qui prend aussi en paramètre l’identifiant du visiteur. Les rapports chargés nous permettent ensuite de stocker le nombre de rapports réalisés par le visiteur dans la variable *$nbVisites*.

A la ligne 25 on charge le dernier rapport réalisé par le visiteur en faisant appel à la fonction getDernierRapport() de la classe rapportDAO qui prend en paramètre l’identifiant du visiteur.

A ligne 26 nous chargeons les offres de médicament du dernier rapport en faisant appel à la fonction getOffreDAO() de la classe *offrirDAO* qui prend en paramètre l’identifiant du dernier rapport.

Enfin, en plus de l’entête et le pied de page, nous incluons le fichier « vueProfil.php » qui permet d’afficher la vue de la page « Mon Profil » vu précédemment.

*Extrait de code (miseAjour.php)*

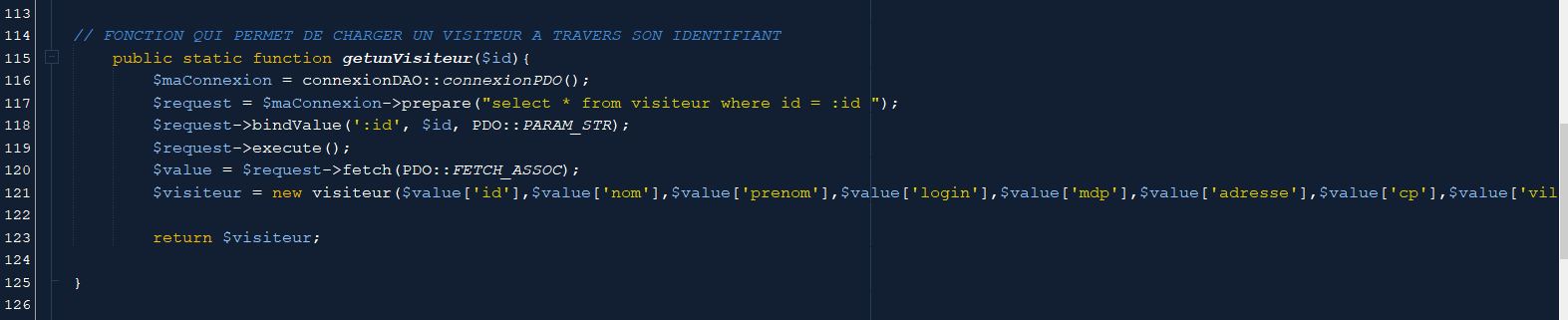


Ce extrait de code du contrôleur « miseAjour.php » permet de récupérer les données saisies par le visiteur dans le formulaire de modification, pour ensuite mettre à jour son profil. Pour effectuer le mis à jour on fait appel à la fonction updateVisiteur() de la classe *visiteurDAO*.

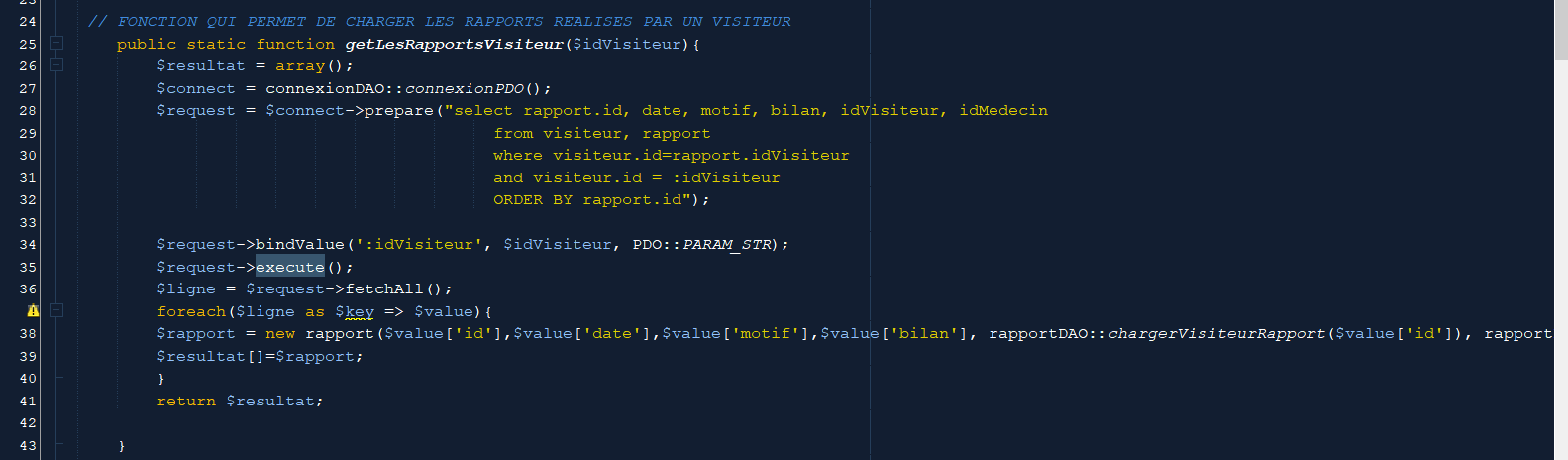
Les éléments du code qui sont entre la ligne 169 et 176 ont déjà été expliqués dans le précédent extrait de code (controleurProfil.php).

1. **Modèle:**

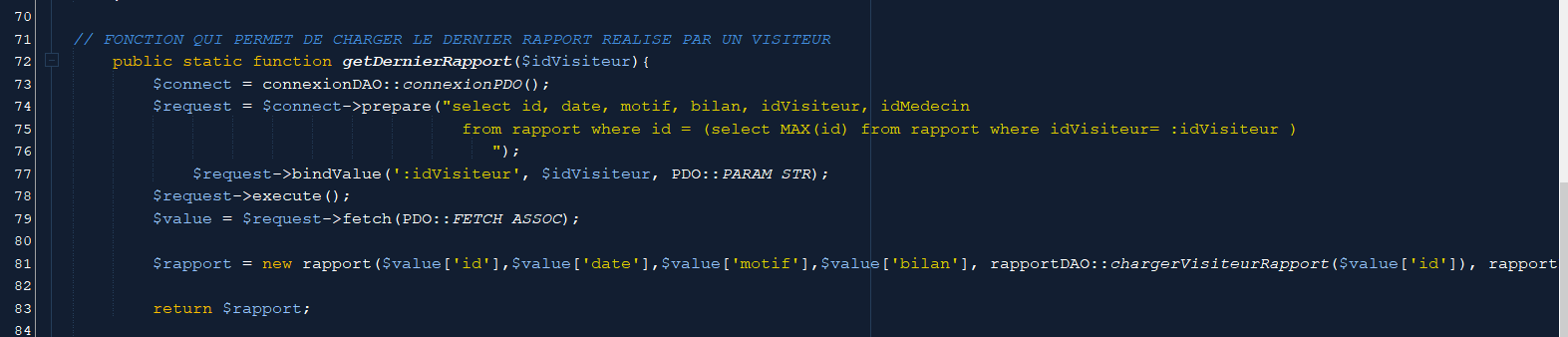
*Extrait de code ( fonction getunVisiteur() / visiteurDAO.php)*



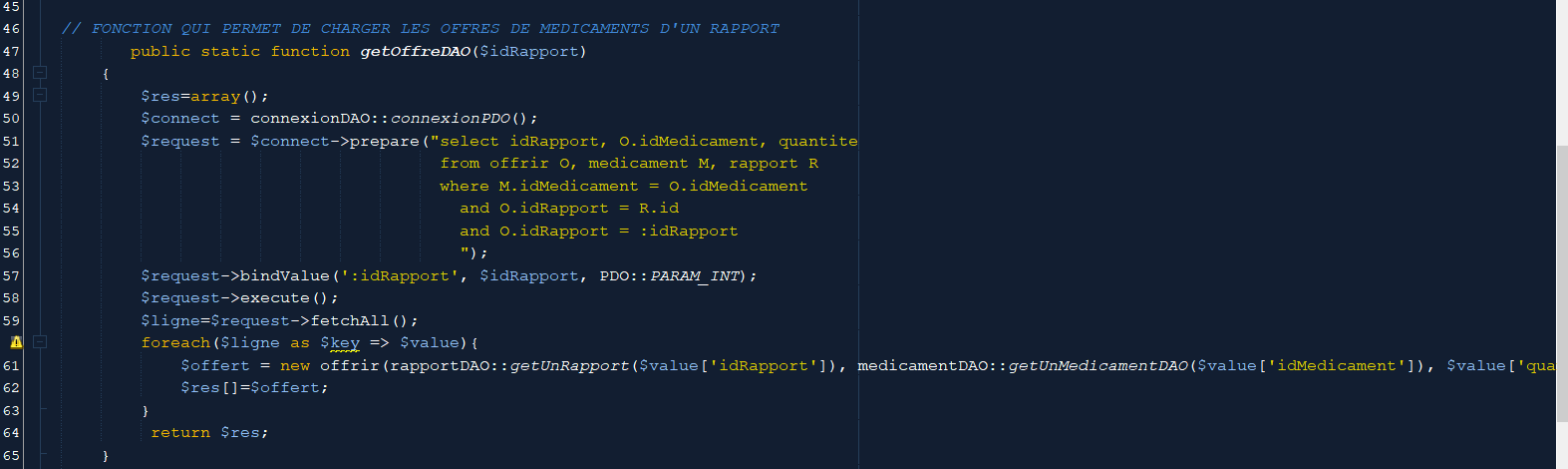
*Extrait de code ( fonction* getLesRapportsVisiteur() */ visiteurDAO.php)*



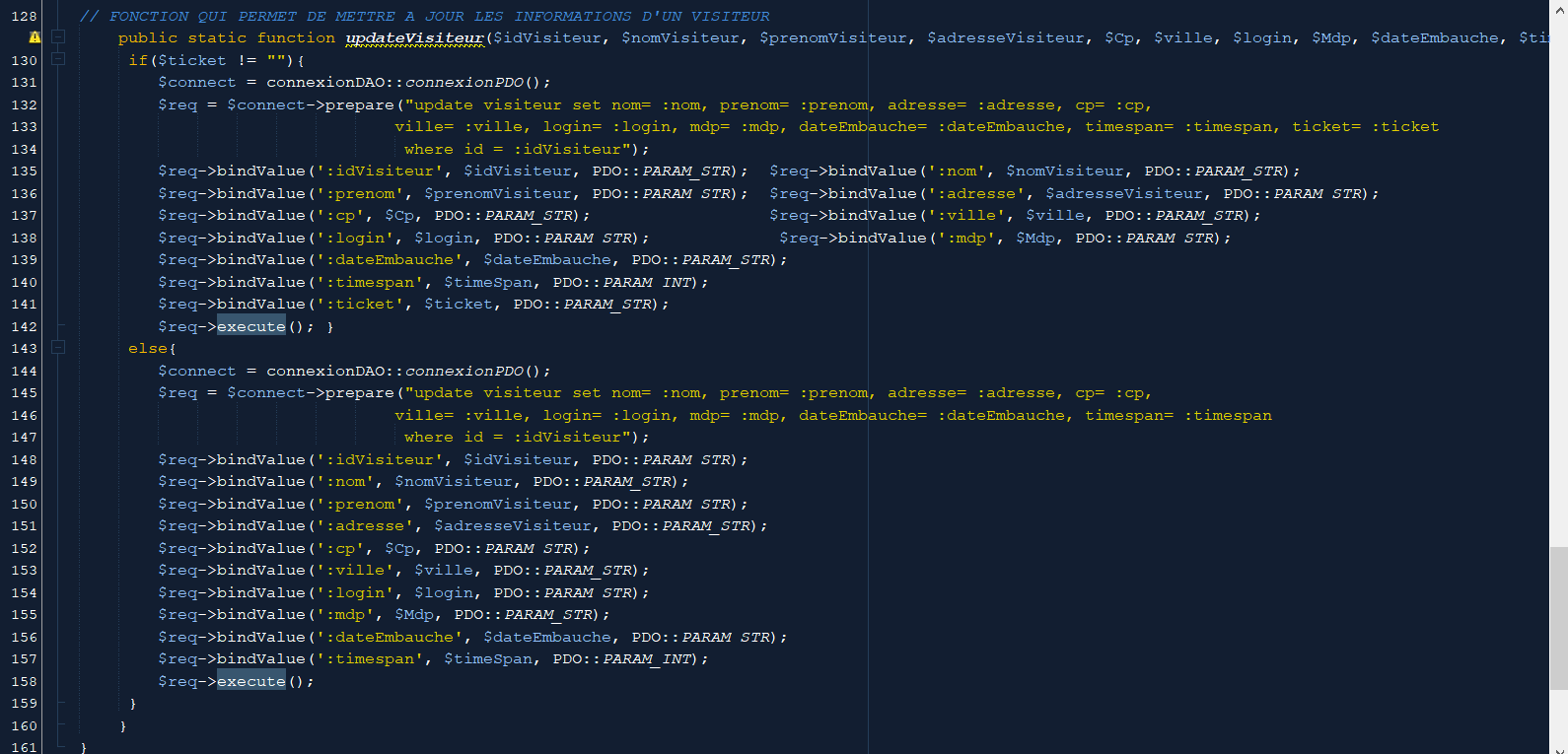
*Extrait de code ( fonction* getDernierRapport() */ rapportDAO.php)*



*Extrait de code ( fonction* getOffreDAO() */ offrirDAO.php)*



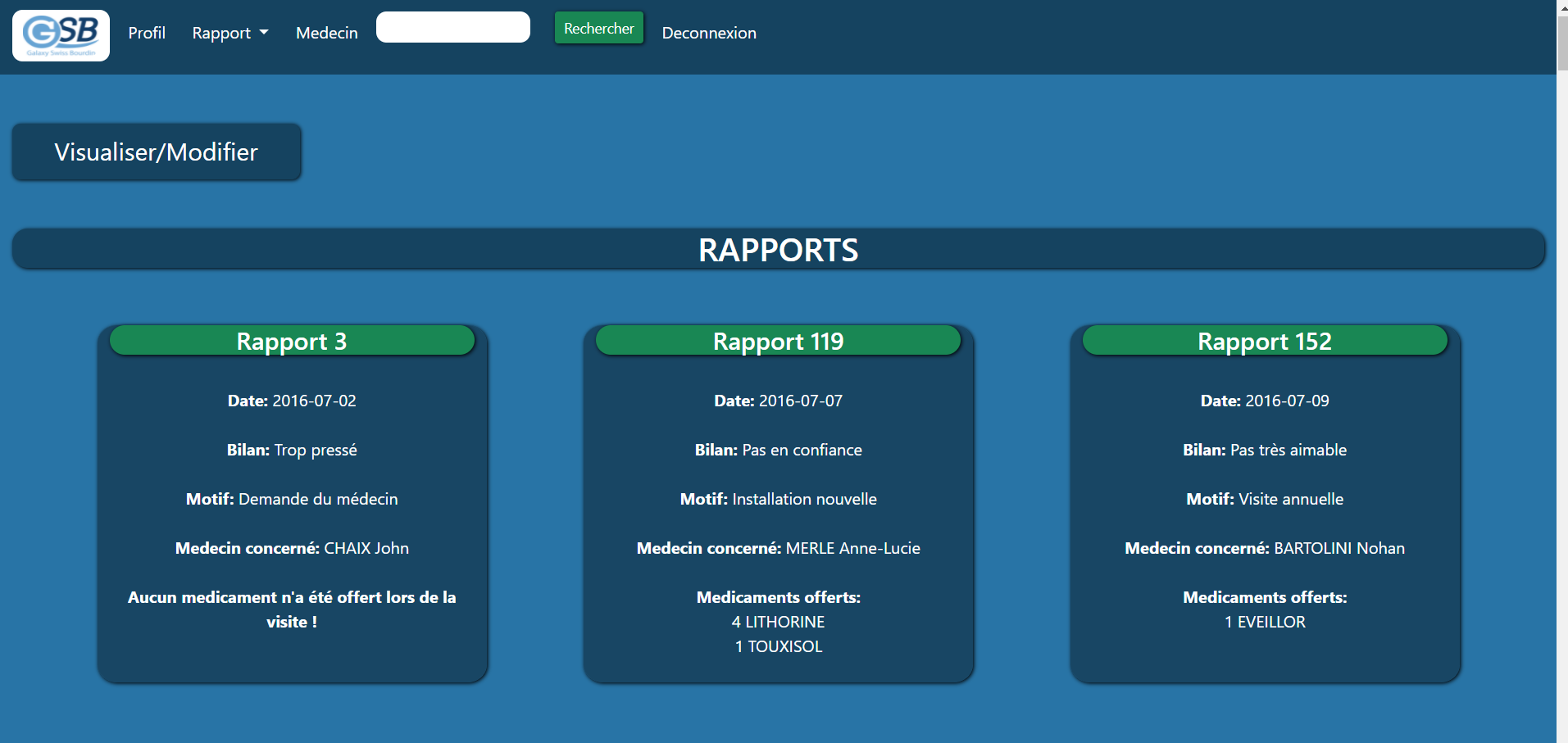
*Extrait de code ( fonction* updateVisiteur()  */ visiteurDAO.php)*



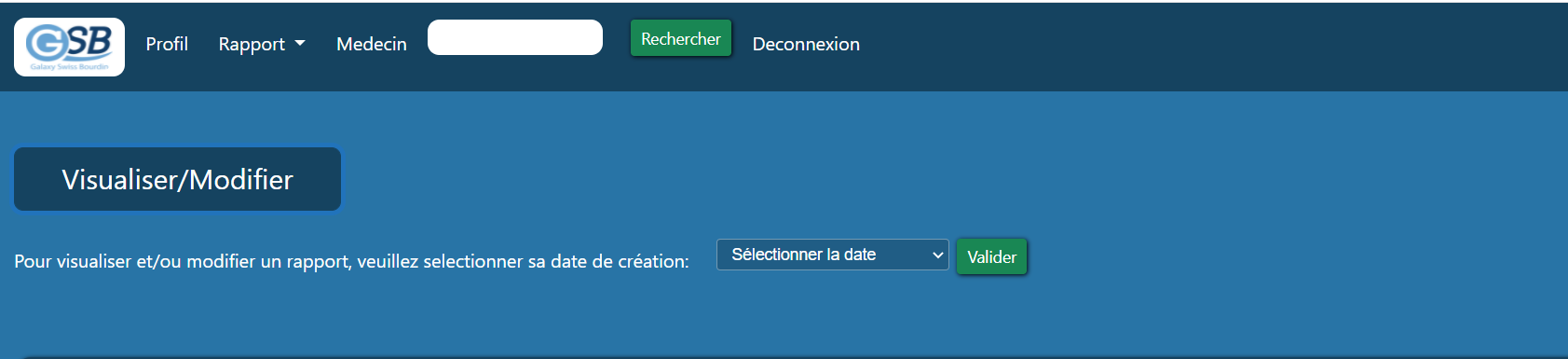
**MES RAPPORTS**

1. **Vue:**

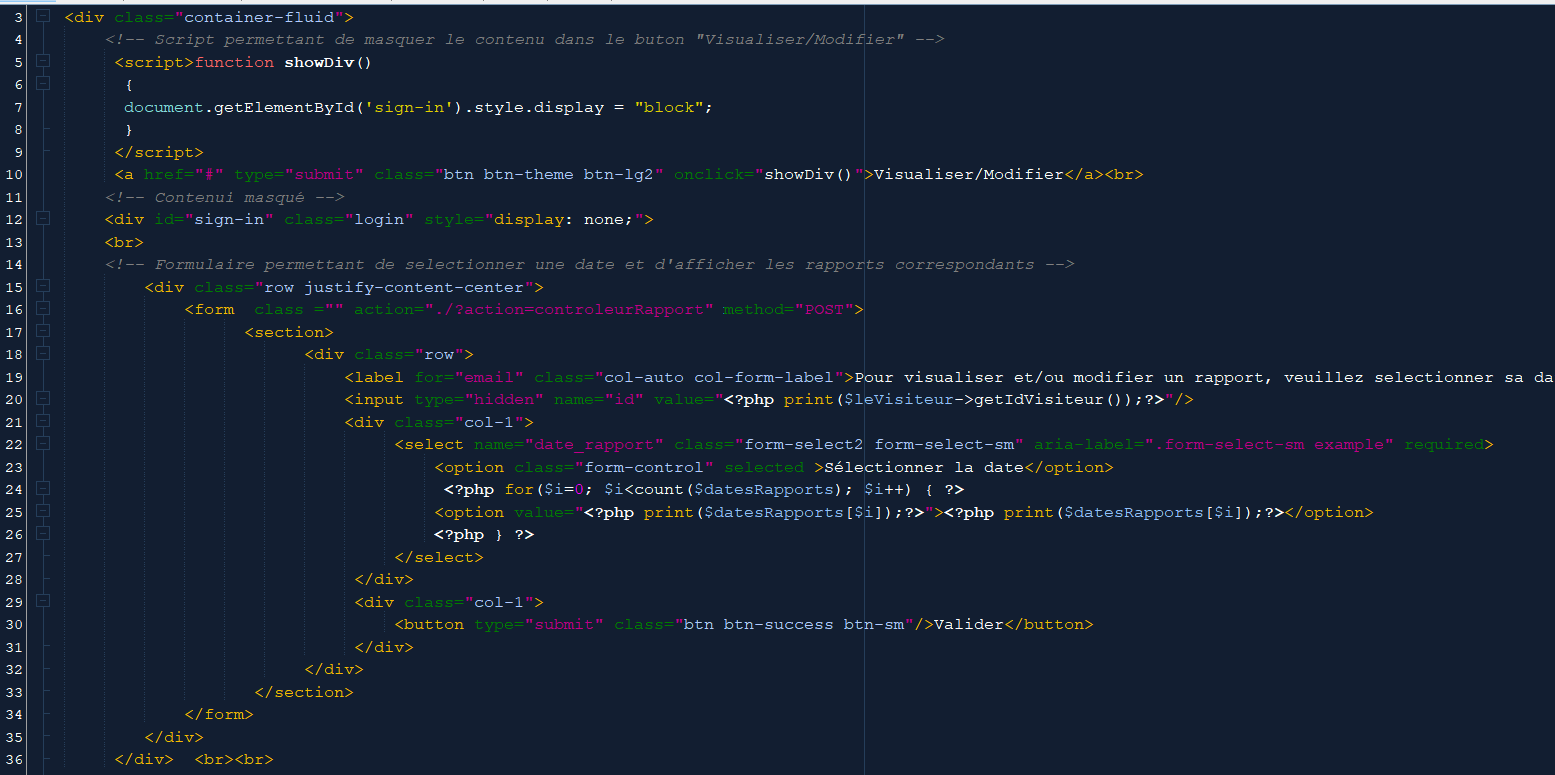
*Coté visiteur*

****

L’image ci-dessus montre une partie de la page « Mes Rapports » qui affiche, en quelques informations, les rapports réalisés par le visiteur. En haut de cette page nous pouvons également voir un bouton « Visualiser/Modifier » qui permet d’afficher un formulaire dans lequel on peut sélectionner une date. Après avoir sélectionné et confirmé une date, le visiteur peut accéder aux informations bien détaillées des rapports qu’il a réalisé à cette date. Il pourra aussi modifier ces rapports.



*Extrait de code (vueMesRapports.php)*

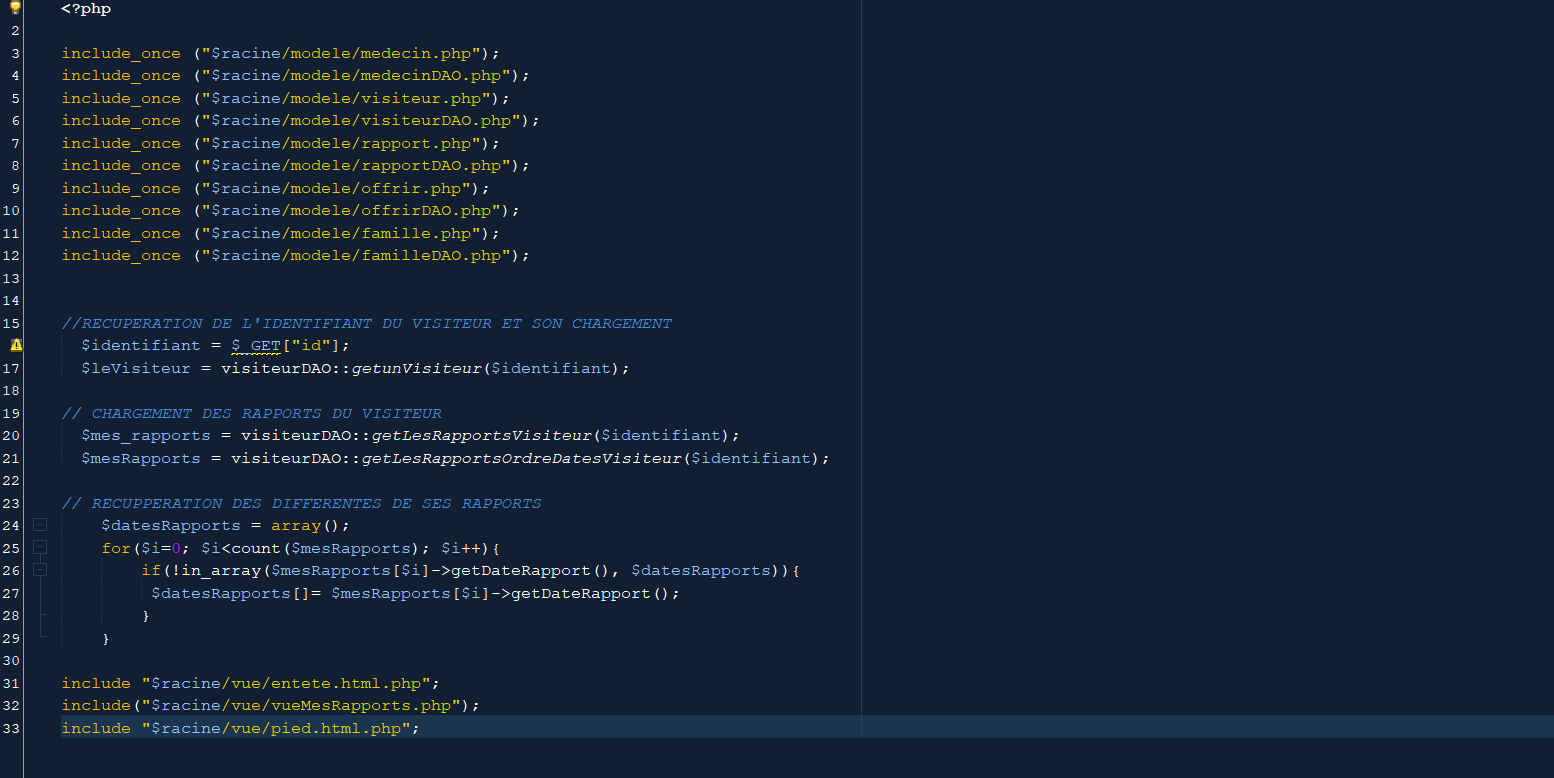
****

**Une image contenant texte

Description générée automatiquement**

1. **Contrôleur:**

*Extrait de code (controleurMesRapports.php)*



Dans ce fichier « controleurMesRapports.php », on récupère en premier l’identifiant du visiteur qui a été transmis au contrôleur en cliquant sur l’élément de navigation « Mes rapport » ou « voir mes rapports ». On charge ensuite le visiteur connecté en faisant appel à la fonction getunVisiteur() de la classe *visiteurDAO* qui prend en paramètre l’identifiant du visiteur récupéré (*$identifiant*).

Après avoir chargé le visiteur, nous chargeons les rapports effectués par le visiteur en faisant appel à la fonction getLesRapportsVisiteur() de la classe visiteurDAO qui prend aussi en paramètre l’identifiant du visiteur. On charge encore les rapports une seconde fois, mais cette fois ci en faisant appel à la fonction getLesRapportsOrdreDatesVisiteur() de la classe *visiteurDAO.* Cette fonction nous permet de charger les rapports en ordre croissant selon la date.

Ensuite nous parcourrons ces rapports chargés pour récupérer les dates des différents rapports pour les stocker dans le tableau *$datesRapports.* Ce sont ces dates que nous retrouvons dans le formulaire de la vue « Mes Rapports ».

Enfin nous incluons le fichier « vueProfil.php, en plus de l’entête et le pied de page, qui permet d’afficher la vue de la page « Mes Rapports ».

1. **Modèle:**

*Extrait de code ( fonction getunVisiteur() / visiteurDAO.php)*: Voir la page 11.

*Extrait de code ( fonction* getLesRapportsVisiteur() */ visiteurDAO.php) :* Voir la page 12.

*Extrait de code ( fonction* getLesRapportsOrdreDatesVisiteur() */ visiteurDAO.php)*

Une image contenant texte

Description générée automatiquement

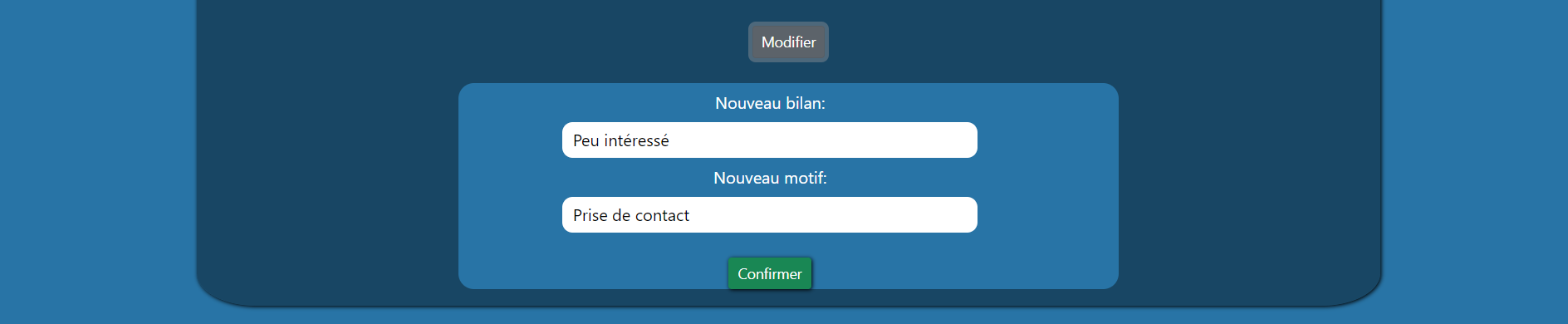
**MODIFICATION D’UN RAPPORT**

1. **Vue:**

*Coté visiteur*



Si le visiteur sélectionne par exemple la date « 2016-07-10 » dans la page précédente « Mes Rapports », il sera redirigé sur la page ci-dessus. Sur cette page le visiteur peut voir les informations en détail des rapports qu’il a réalisé à cette date. Il peut aussi modifier un rapport. En cliquant sur le bouton « Modifier », un formulaire qui permet de modifier le bilan et le motif du rapport apparait.

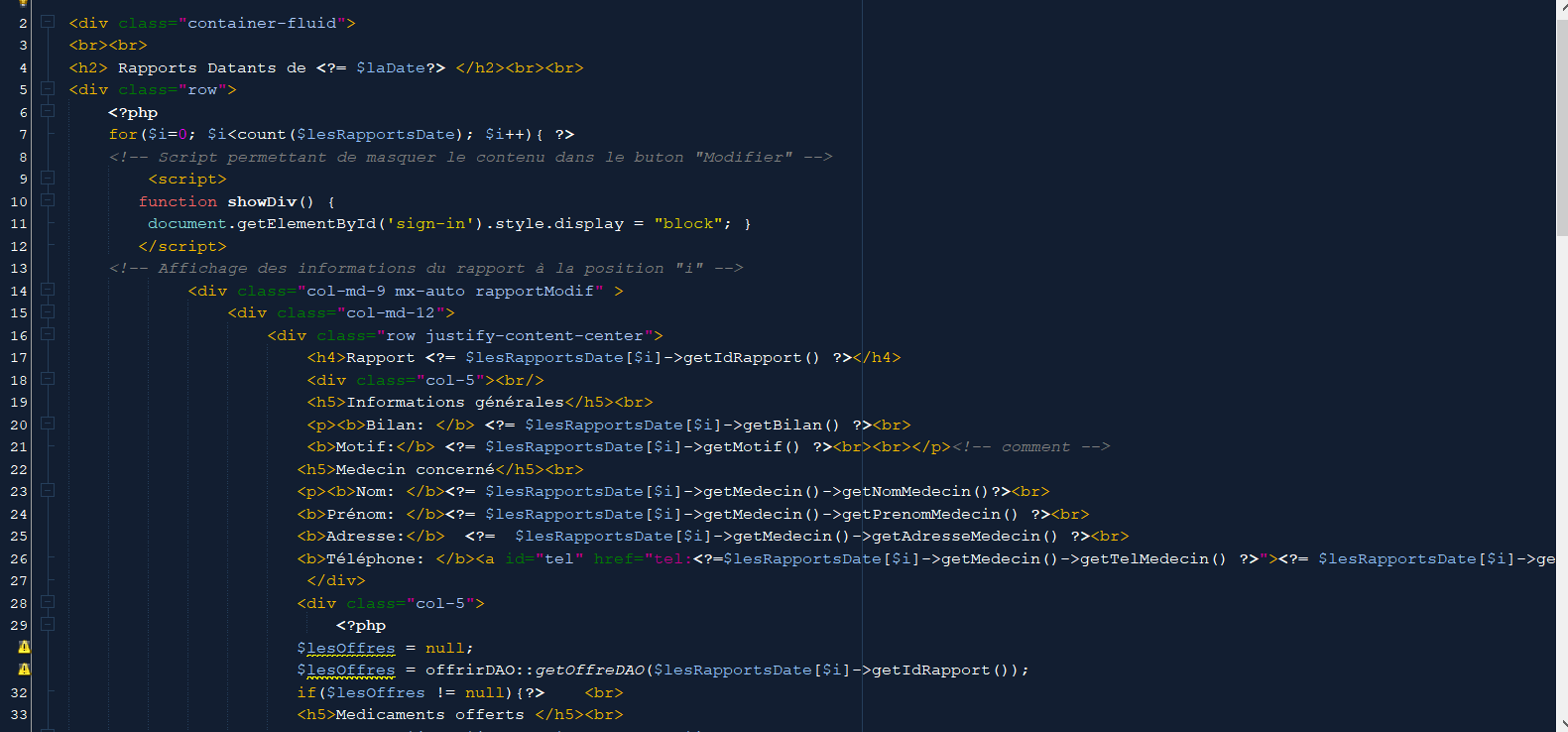


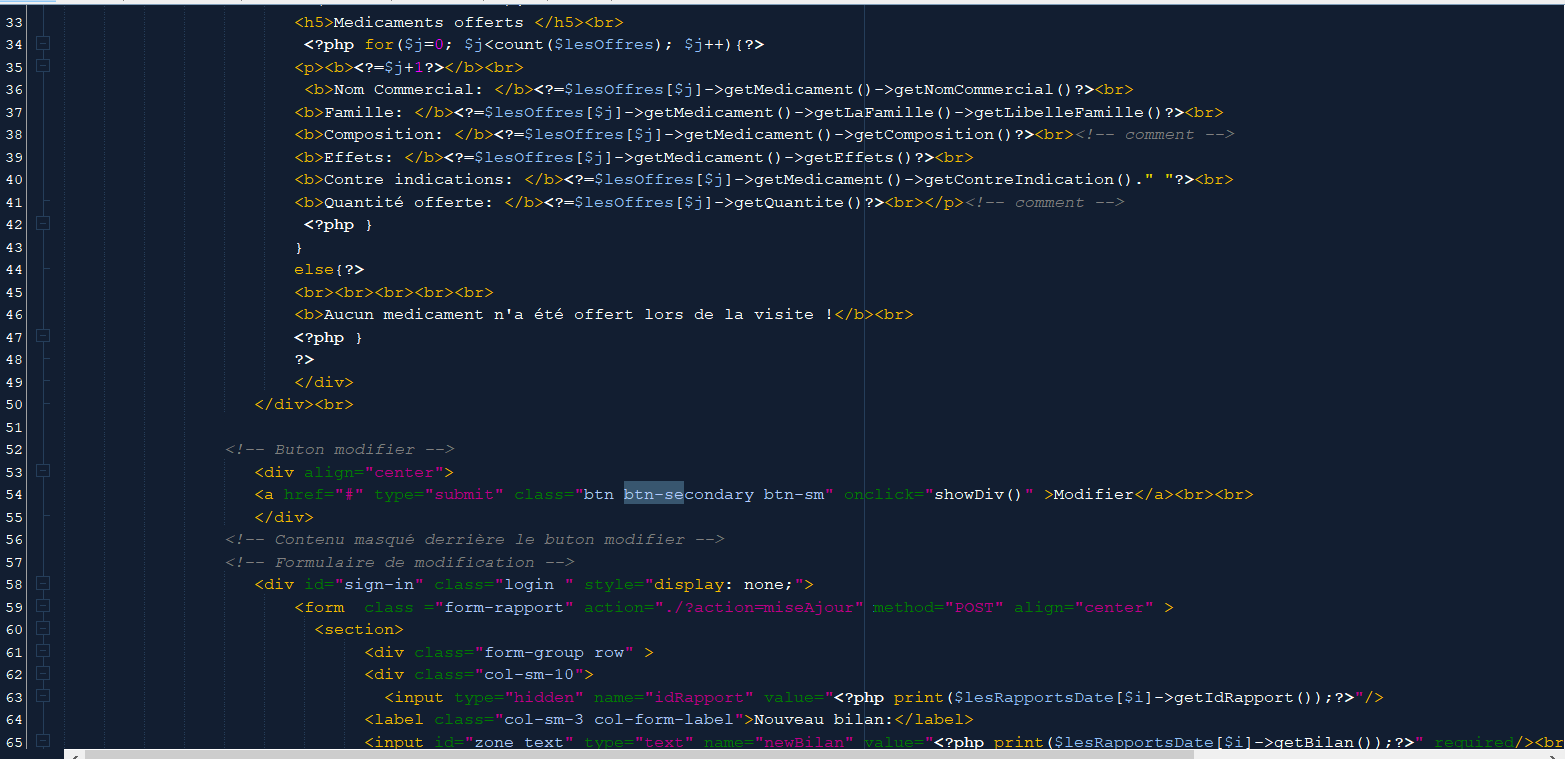
Après avoir saisi et confirmé les nouvelles données le visiteur est dirigé vers une page qui lui confirme que la modification a bien été effectué. On peut voir un extrait de cette page sur l’image ci-dessous.



Comme on peut le voir sur cette cette page le visiteur à la possibilité d’apporter des corrections en cas d’erreur en cliquant encore une nouvelle sur le bouton « Modifier »

*Extrait de code (vueRapport.php)*

****

****

Une image contenant texte

Description générée automatiquement

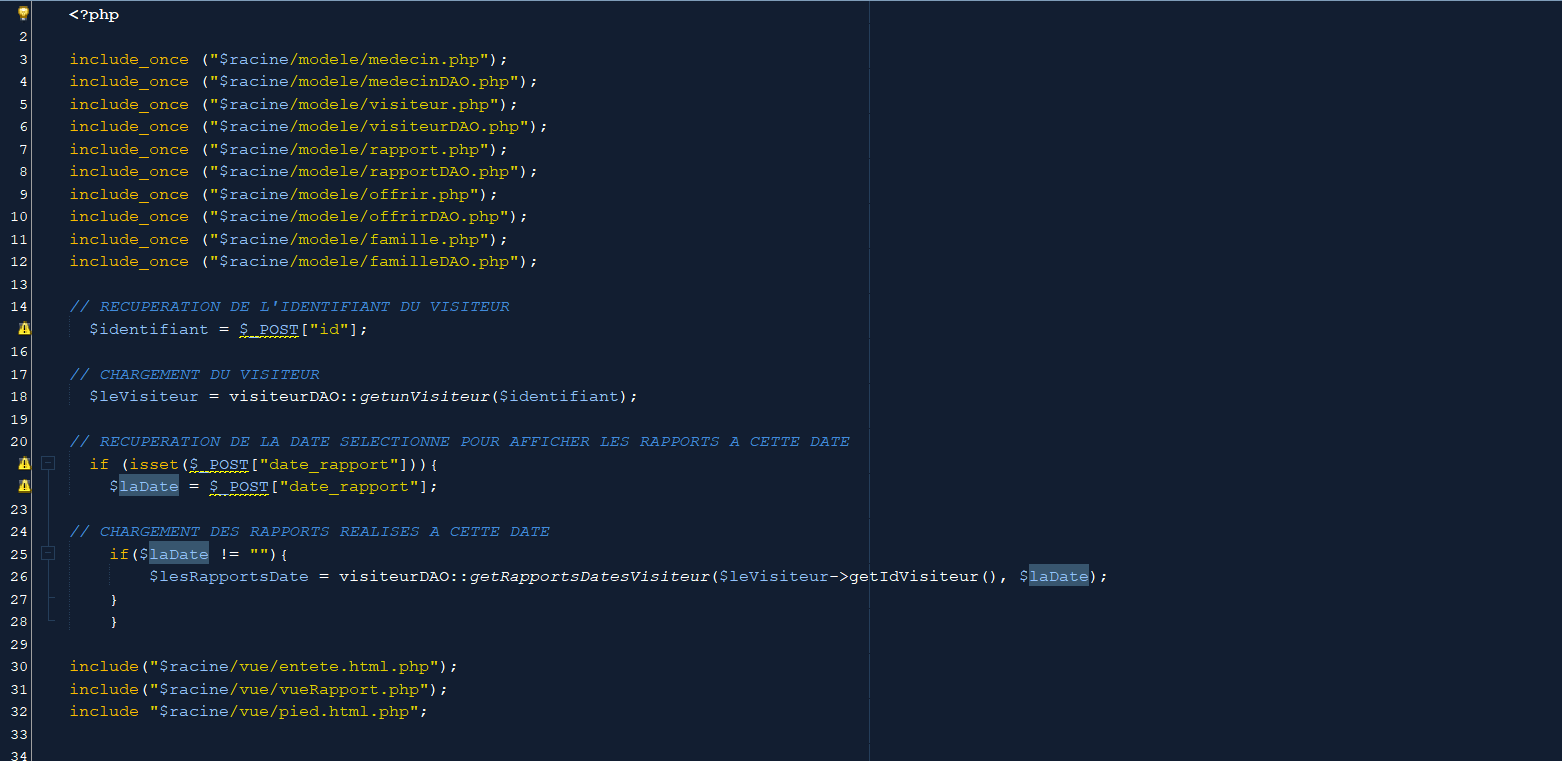
*Extrait de code (vueMiseAjour.php)*

****

Cet extrait ci-dessous représente juste une partie du code qui permet d’afficher la page qui confirme la modification d’un rapport. Le reste du code est pareil que celui de la vue «vueRapport.php».

1. **Contrôleur:**

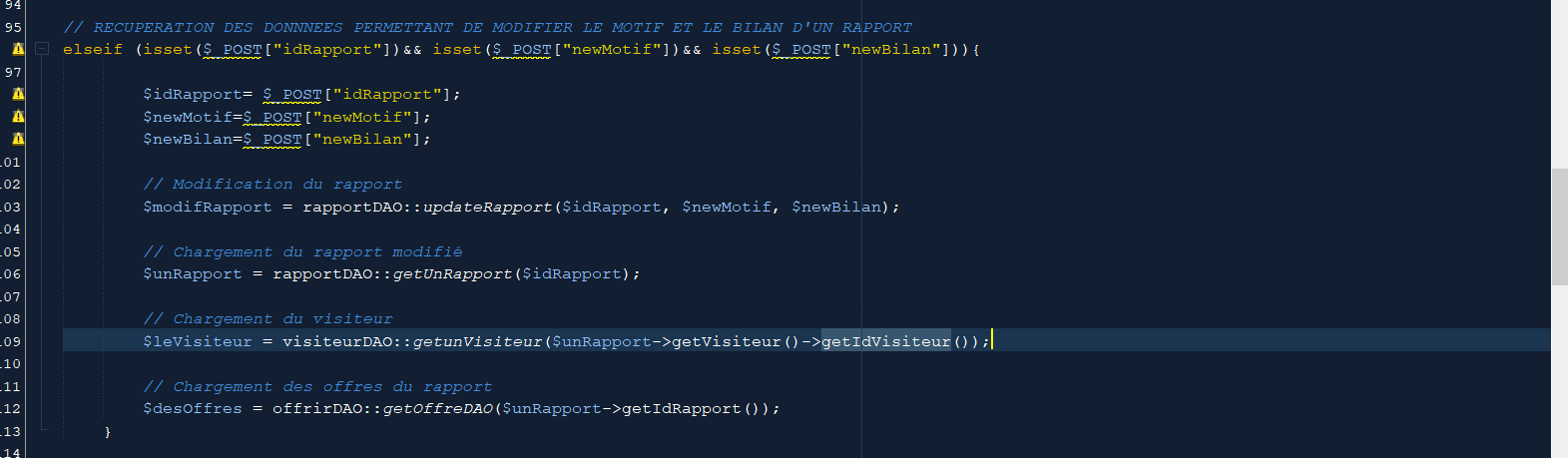
*Extrait de code (controleurRapport.php)*



Les nouveaux éléments sur ce extrait de code qui n’ont pas été vu précédemment sont la récupération de la date et le chargement des rapports réalisés à cette date. De la ligne 21 à 22 on récupère la date transmis au contrôleur après l’envoie du formulaire de la page « Mes Rapports » qui permet de sélectionner une date. Cette date est stocké dans la variable *$laDate.*

Ensuite, sur la ligne 26, nous chargeons tous les rapports réalisés par le visiteur à la date récupérée en faisant appel à la fonction getRapportsDatesVisiteur() de la classe visiteurDAO(). Cette fonction prend en paramètre l’identifiant du visiteur et la date.

*Extrait de code (vueMiseAjour.php)*



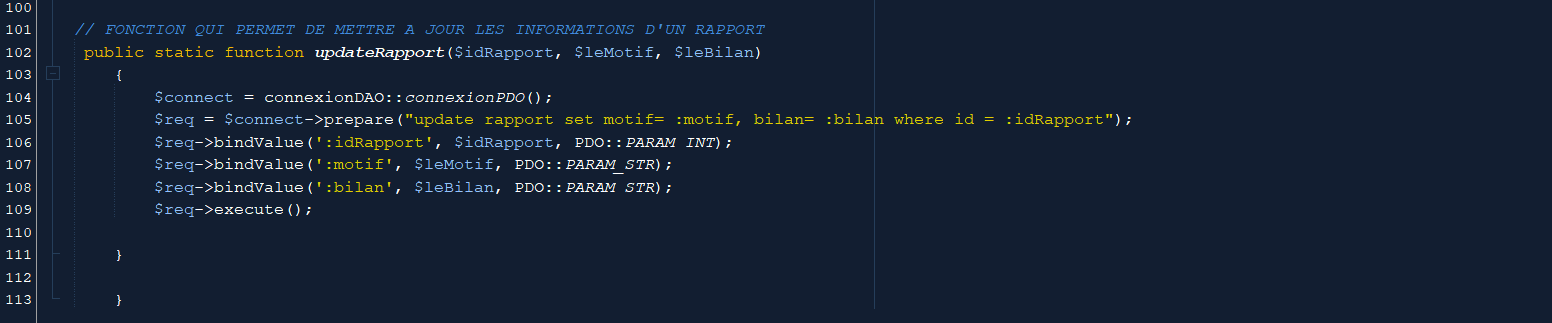
Sur cet extrait nous allons souligner la récupération des données (idRapport, newMotif, newBilan) transmis au contrôleur qui permettent de mettre à jour un rapport. Après avoir récupéré les données, on fait appel à la fonction updateRapport() de la classe rapportDAO pour modifier le bilan et le motif du rapport.

1. **Modèle:**

*Extrait de code ( fonction getRapportsDatesVisiteur() / visiteurDAO.php)*



*Extrait de code ( fonction updateRapport() / rapportDAO.php)*



**CREATION D’UN RAPPORT**

1. **Vue:**

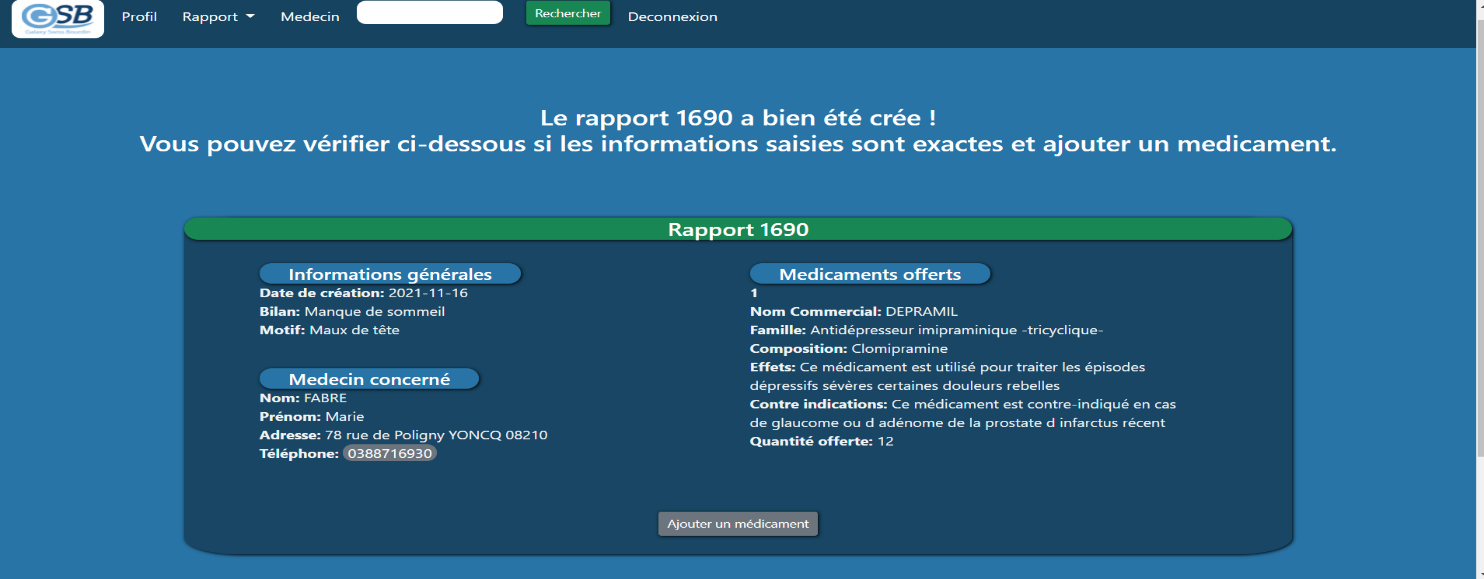
Quand le visiteur clique sur l’élément de navigation « Rapport / Créer un rapport » au niveau de la barre de navigation ou sur la page d’accueil, il est automatiquement dirigé vers la page de création d’un rapport.

*Coté visiteur*

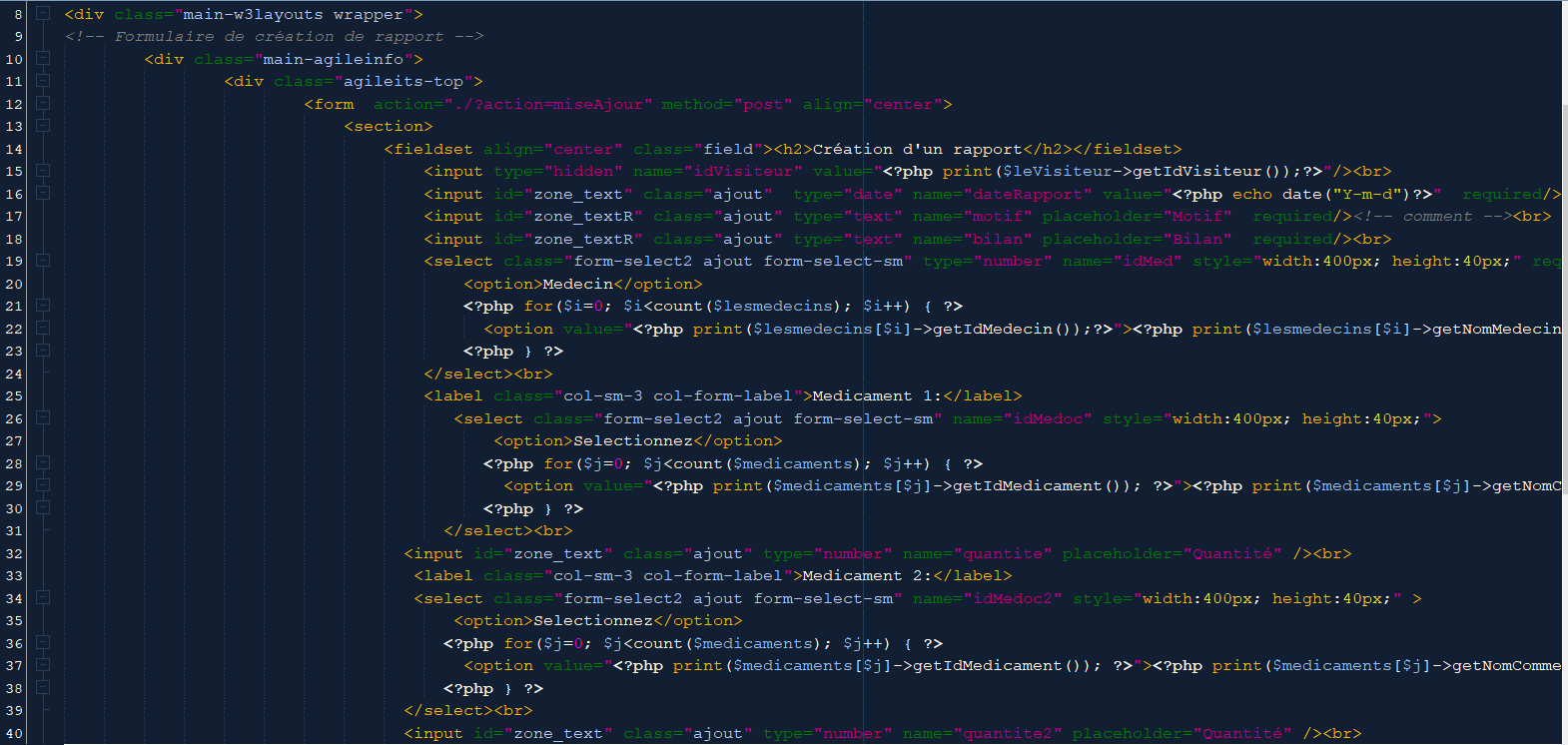


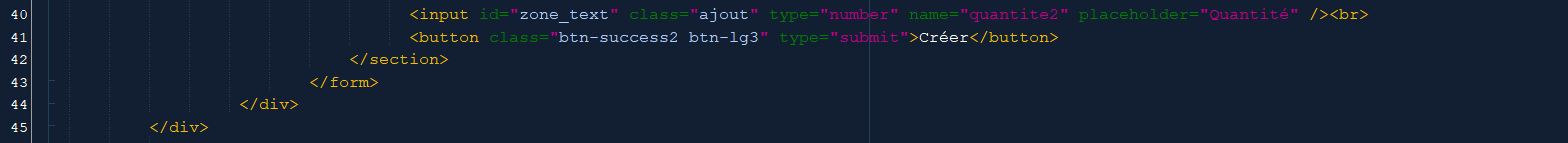


Comme vous l’auriez compris, la page ci-dessus permet de créer un nouveau rapport en remplissant les différents champs. Après avoir remplis les différents champs et cliqué sur le bouton « Créer », le visiteur est dirigé vers la page qui lui confirme que la création a bien été effectué. Sur cette page le visiteur pourra également ajouter un médicament au rapport. Comme on peut le voir sur l’image ci-dessous.

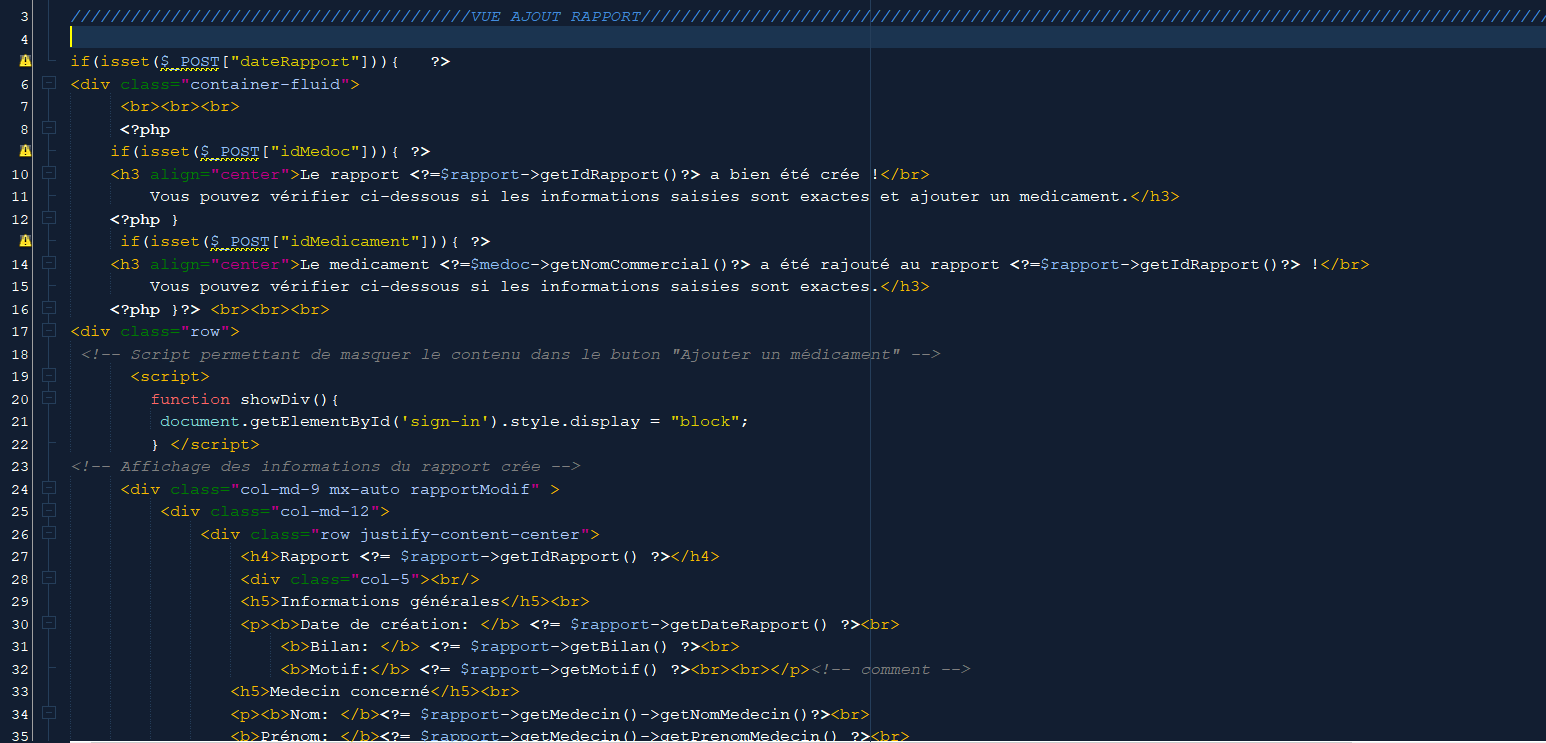


*Extrait de code (vueAjoutRapport.php)*

****

****

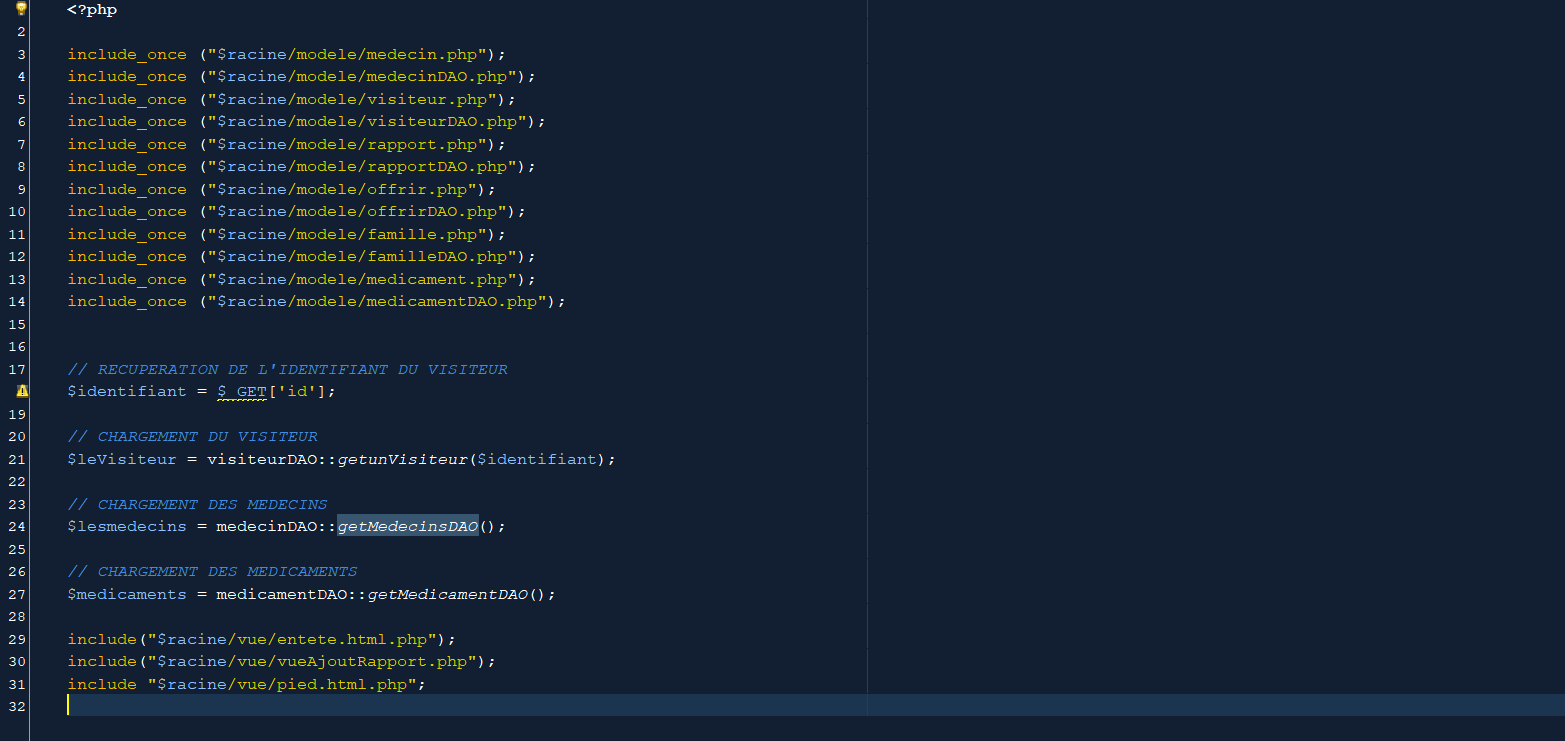
*Extrait de code (vueMiseAjour.php)*

****

Cet extrait ci-dessous représente juste une partie du code qui permet d’afficher la page qui confirme la création d’un rapport. Le reste du code est pareil que celui de la vue «vueRapport.php» qui affiche les informations détaillées d’un rapport.

1. **Contrôleur:**

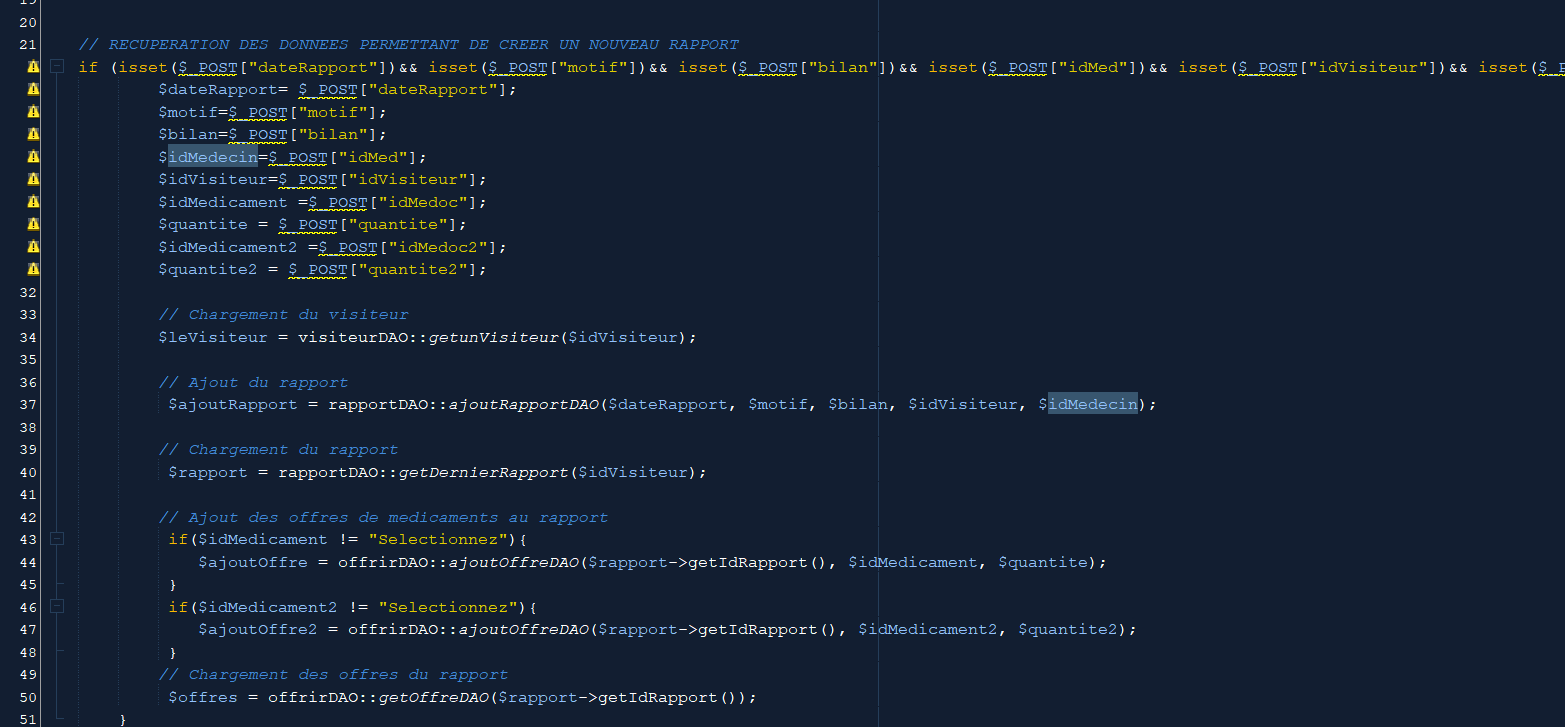
*Extrait de code (controleurAjoutRapport.php)*



Dans cet extrait nous mettrons l’accent sur le chargement des médecins et des médicaments. On charge les médecins en faisant appel à la fonction getMedecinsDAO() de la classe medecinDAO. Ensuite on charge les médicaments en faisant appel à la fonction getMedicamentDAO() de la classe medicamentDAO.

Après on inclut les différentes vues concernées, à savoir *: entete.html.php, vueAjoutRapport.php et pied.html.php*

*Extrait de code (miseAjour.php)*

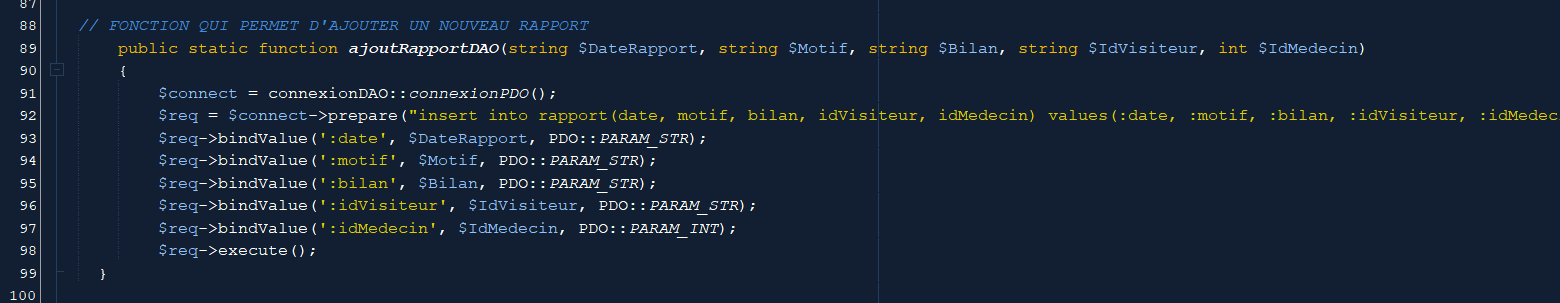


Dans l’extrait de code ci-dessous nous allons évoquer l’appel aux fonctions qui n’ont pas été vu précédemment, à savoir : ajoutRapportDAO() et ajoutOffreDAO() .

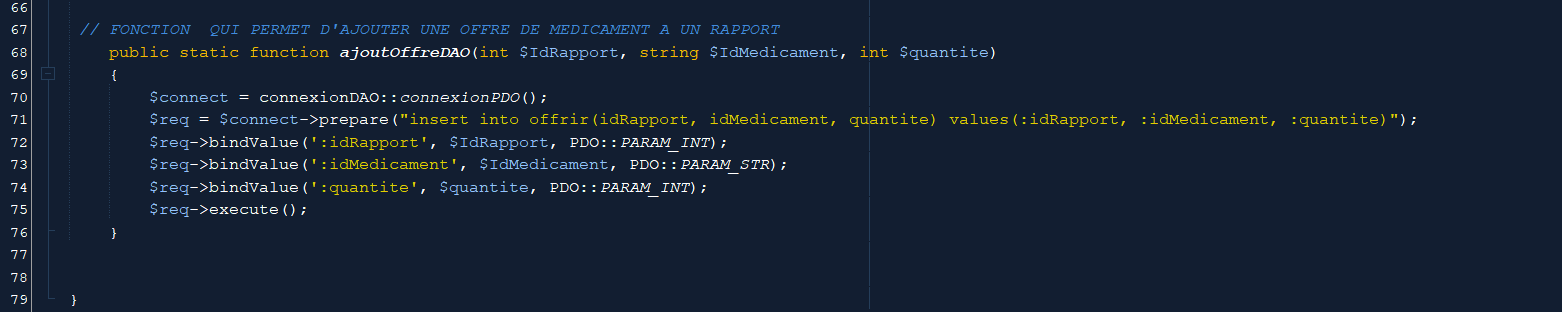
Après avoir récupéré toutes les données transmis par le formulaire « Création d’un rapport », nous procédons aux chargements du visiteur et du rapport crée (qui est le dernier rapport). Nous effectuons l’ajout du rapport dans la base de données en faisant appel à la fonction ajoutRapportDAO() de la classe rapportDAO. Ensuite nous faisons appel à la fonction ajoutOffreDAO() de la classe offrirDAO() pour ajouter les médicaments offerts et leur quantité dans le rapport. Cette fonction prend en paramètre l’identifiant du médicament sélectionné et la quantité offerte.

1. **Modèle:**

*Extrait de code ( fonction ajoutRapportDAO() / rapportDAO.php)*



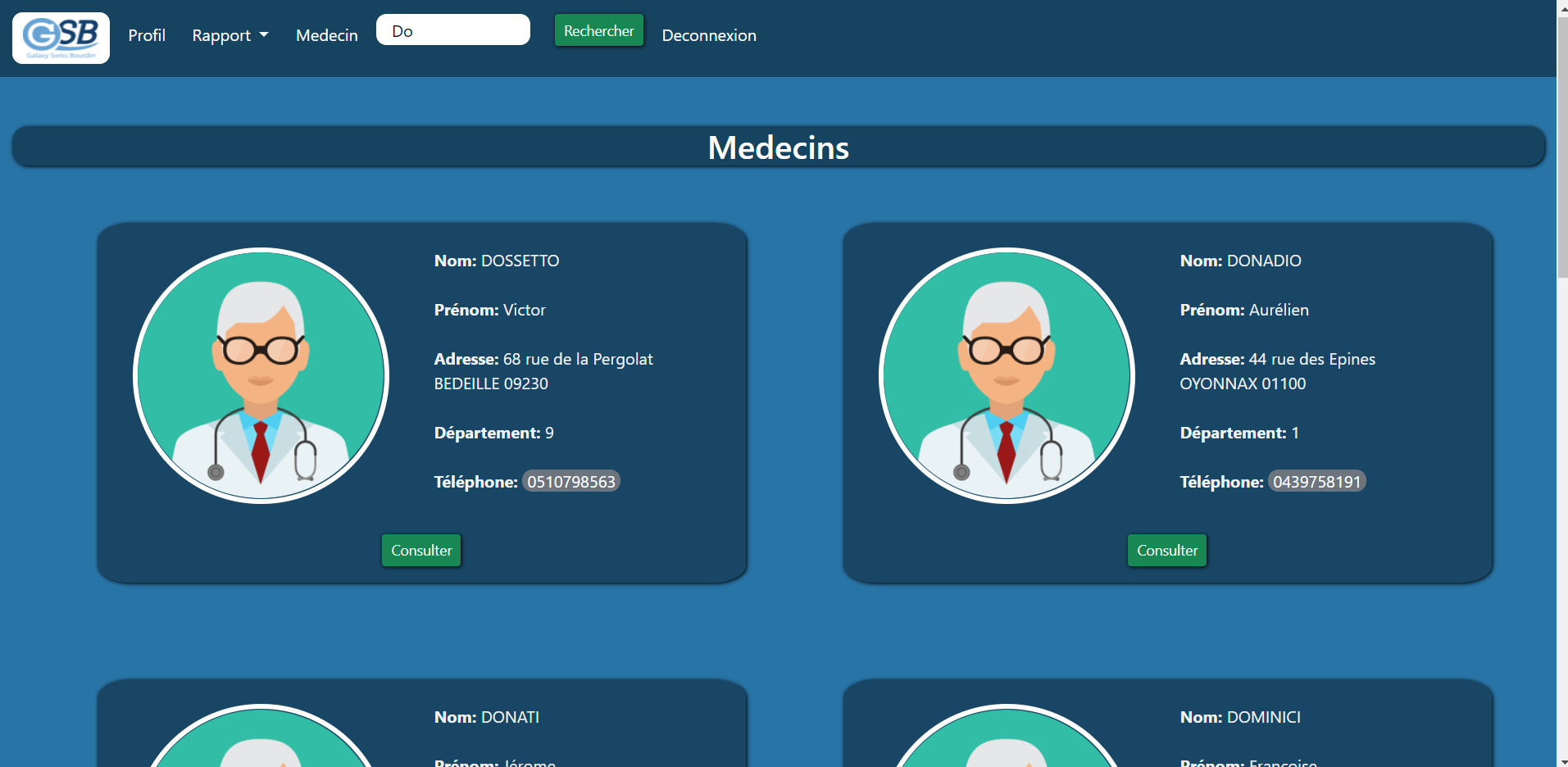
*Extrait de code ( fonction ajoutOffreDAO() / offrirDAO.php)*



**RECHERCHE D’UN MEDECIN**

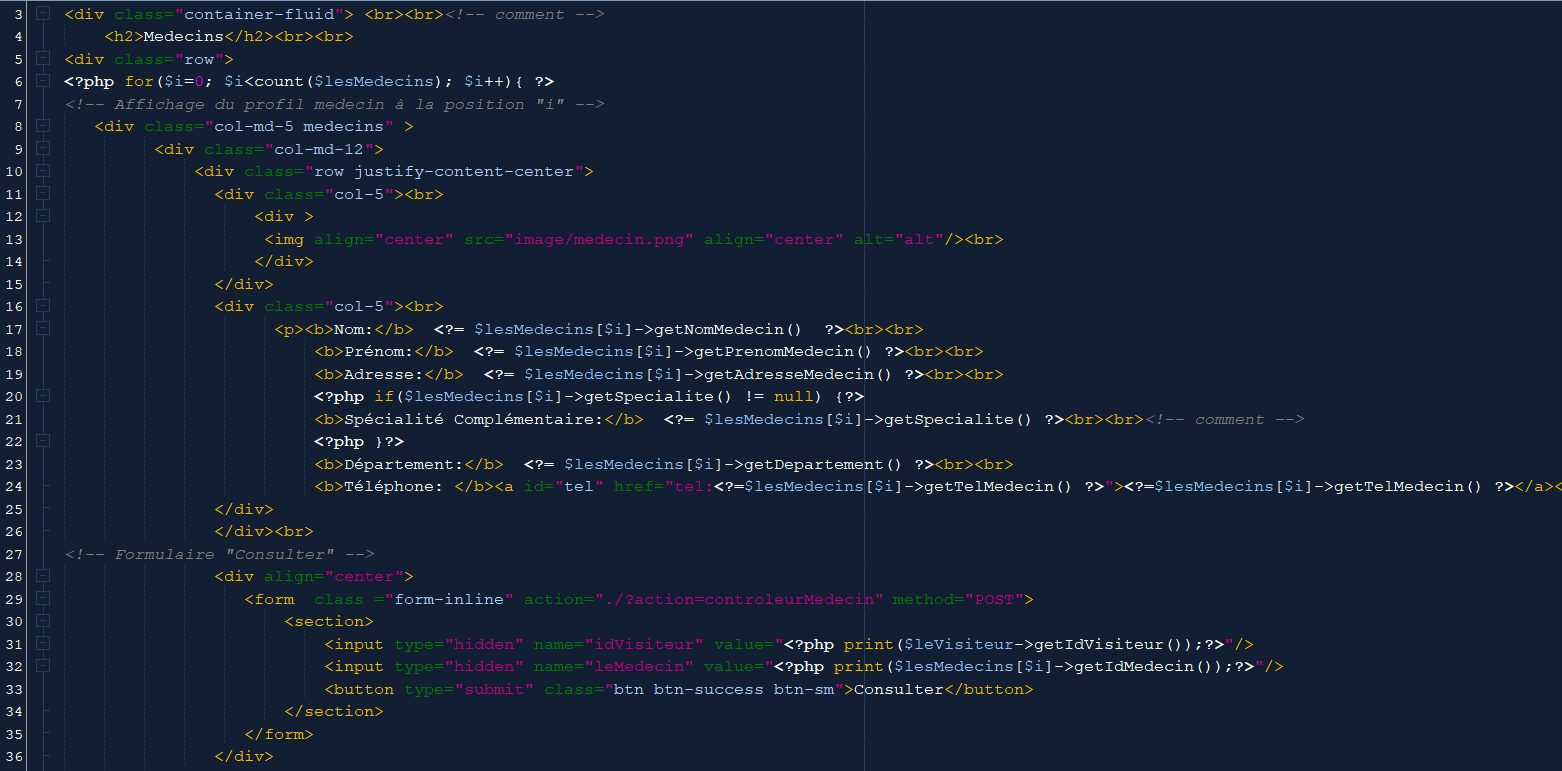
1. **Vue:**

*Coté visiteur*



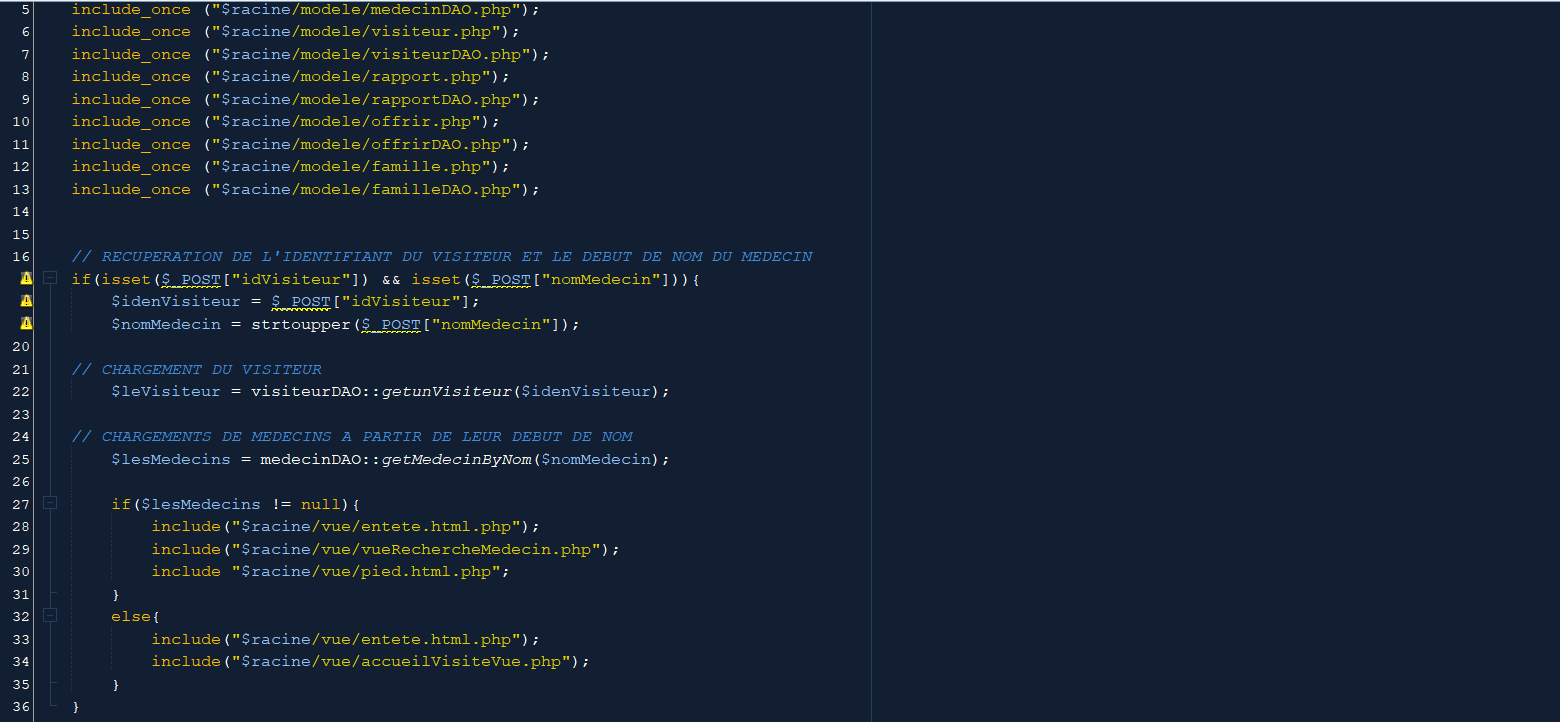
Lorsque le visiteur fait une recherche de médecin sur la barre de navigation en saisissant le début du nom du médecin, il est dirigé sur la page « Médecins » qu’on voit ci-dessus. Par exemple s’il saisit « do », le visiteur verra sur cette page tous les visiteurs dont le nom commence par « do ». Ensuite il peut consulter un médecin en particulier en cliquant sur le bouton « Consulter ».

*Extrait de code (vueRechercheMedecin.php)*

****

1. **Contrôleur:**

*Extrait de code (controleurRechercheMedecin.php)*



La fonction à souligner en particulier dans cet extrait de code est getMedecinByNom(). Après avoir récupéré les différentes données transmis par le formulaire de recherche, on fait appel à la fonction getMedecinByNom()  de la classe medecinDAO. Cet appel permet de charger les médecins dont le début de nom correspond au nom medecin (« nomMedecin ») posté par le formulaire.

Si des médecins sont retournés par la fonction et que la variable $lesMedecins n’est pas nul, on inclus les vue suivantes : *entete.html.php, vueRechercheMedecin.php, pied.html.php*. Le cas contraire le visiteur reste sur la page d’accueil.

1. **Modèle:**

*Extrait de code ( fonction getMedecinByNom() / medecinDAO.php)*



**MEDECIN**

1. **Vue:**

*Coté visiteur*



****

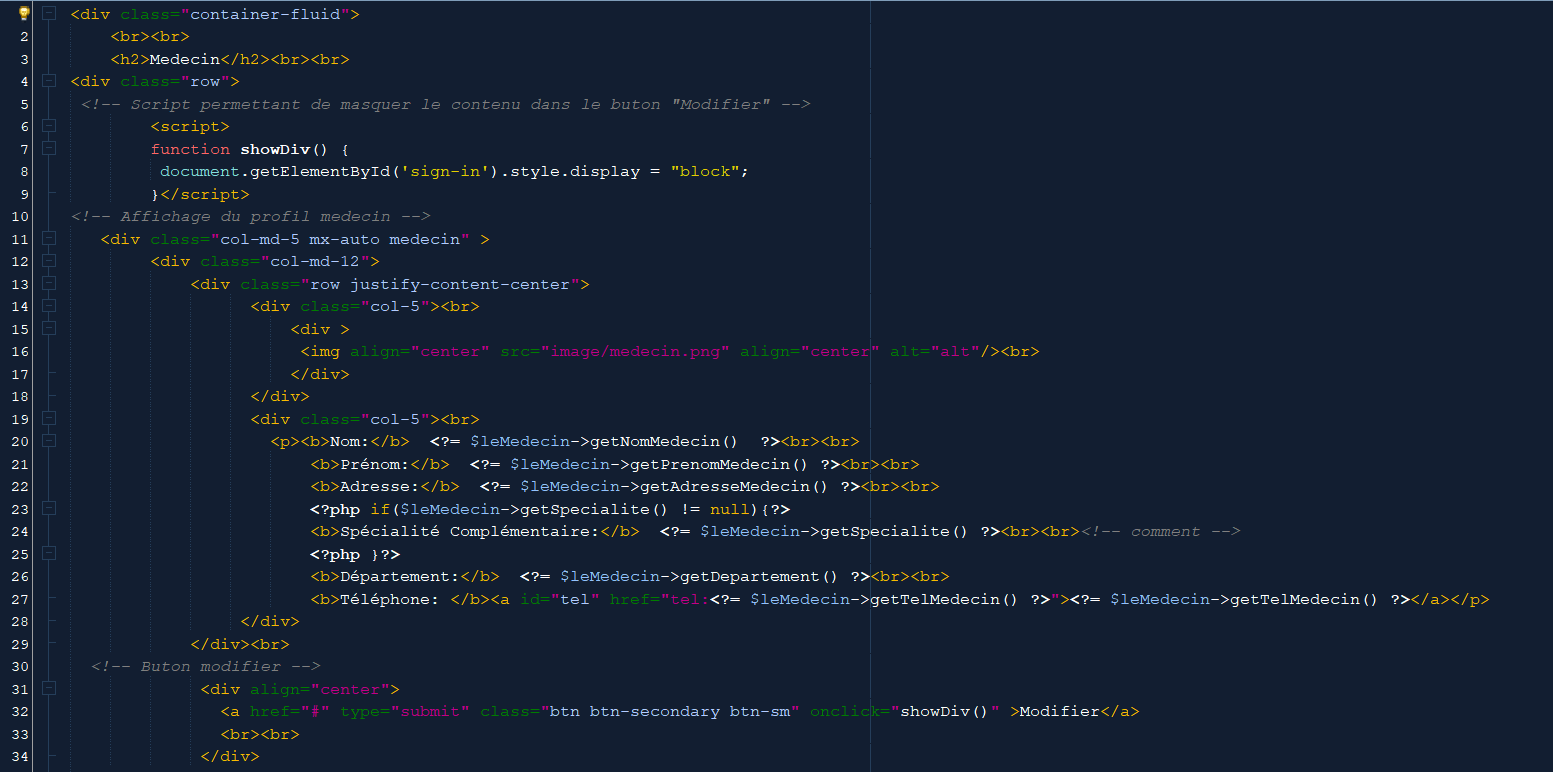
Comme on peut le voir sur ces deux images ci-dessus, quand le visiteur clique sur le bouton « Consulter » sur la page précédente, le visiteur accède à cette page. Sur cette page le visiteur, peut modifier les informations du médecin en cliquant sur le bouton « Modifier ». Il peut également voir les rapports concerné par ce médecin. En cliquant sur le bouton « Modifier », un formulaire apparait qui permet au visiteur de saisir les nouvelles données du médecin.



Après avoir saisi et confirmé les données, le visiteur est dirigé vers la page qui lui confirme que la modification a bien été effectué. Il aura aussi la possibilité d’apporter des corrections sur cette page en cas d’erreurs de frappe, comme on peut le voir sur l’image ci-dessous.



*Extrait de code (vueMedecin.php)*

****

Une image contenant texte

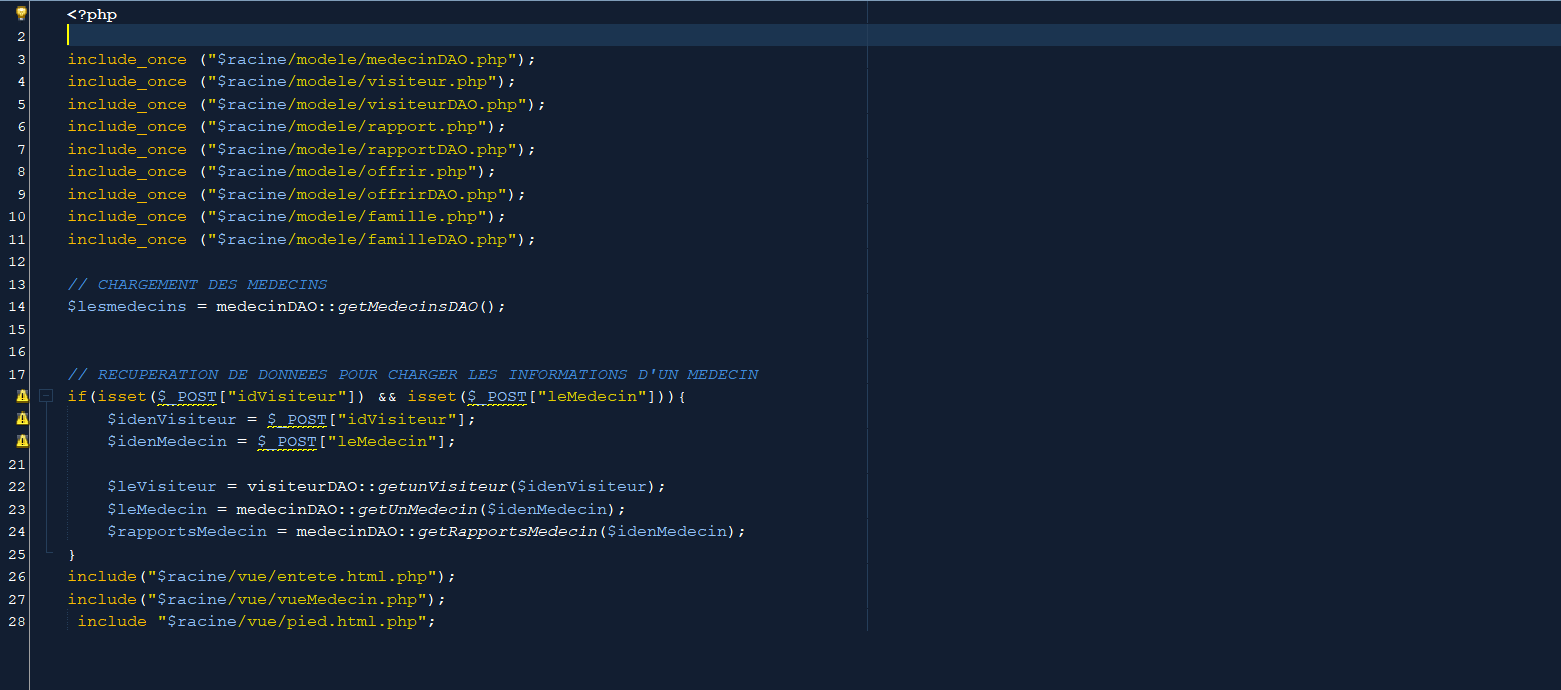
Description générée automatiquement

Une image contenant texte

Description générée automatiquement

1. **Contrôleur:**

*Extrait de code (controleurMedecin.php)*



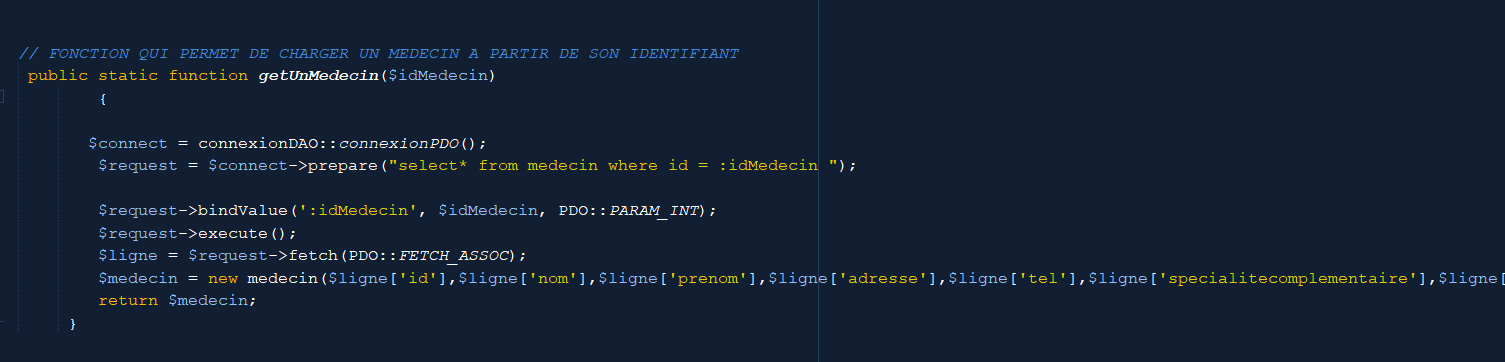
Dans cet extrait après avoir récupéré l’identifiant du visiteur et celui du médecin, on charge le médecin en faisant appel à la fonction getUnMedecin() de la classe *medecinDAO* qui prenden paramètre l’identifiant du médecin. Ensuite nous chargeons les rapports concernés par un medecin en faisant appel à la fonction getRapportsMedecin() de la classe medecinDAO qui prend aussi en paramètre l’identifiant du médecin.

Après on inclut les différentes vues concernées, à savoir *: entete.html.php,*

*vueMedecin.php et pied.html.php .*

1. **Modèle:**

*Extrait de code ( fonction getunMedecin() / medecinDAO.php)*



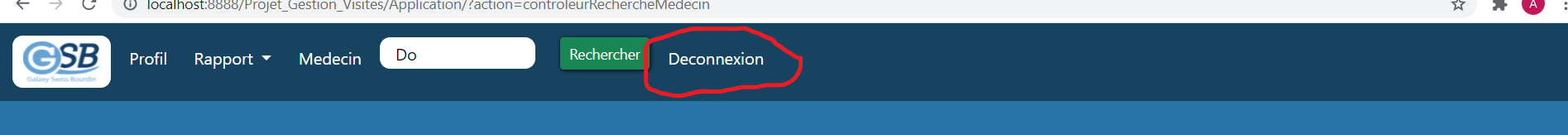
*Extrait de code ( fonction getRapportsMedecin() / medecinDAO.php)*



**DECONNEXION - ENTETE**

1. **Vue:**

*Coté visiteur*

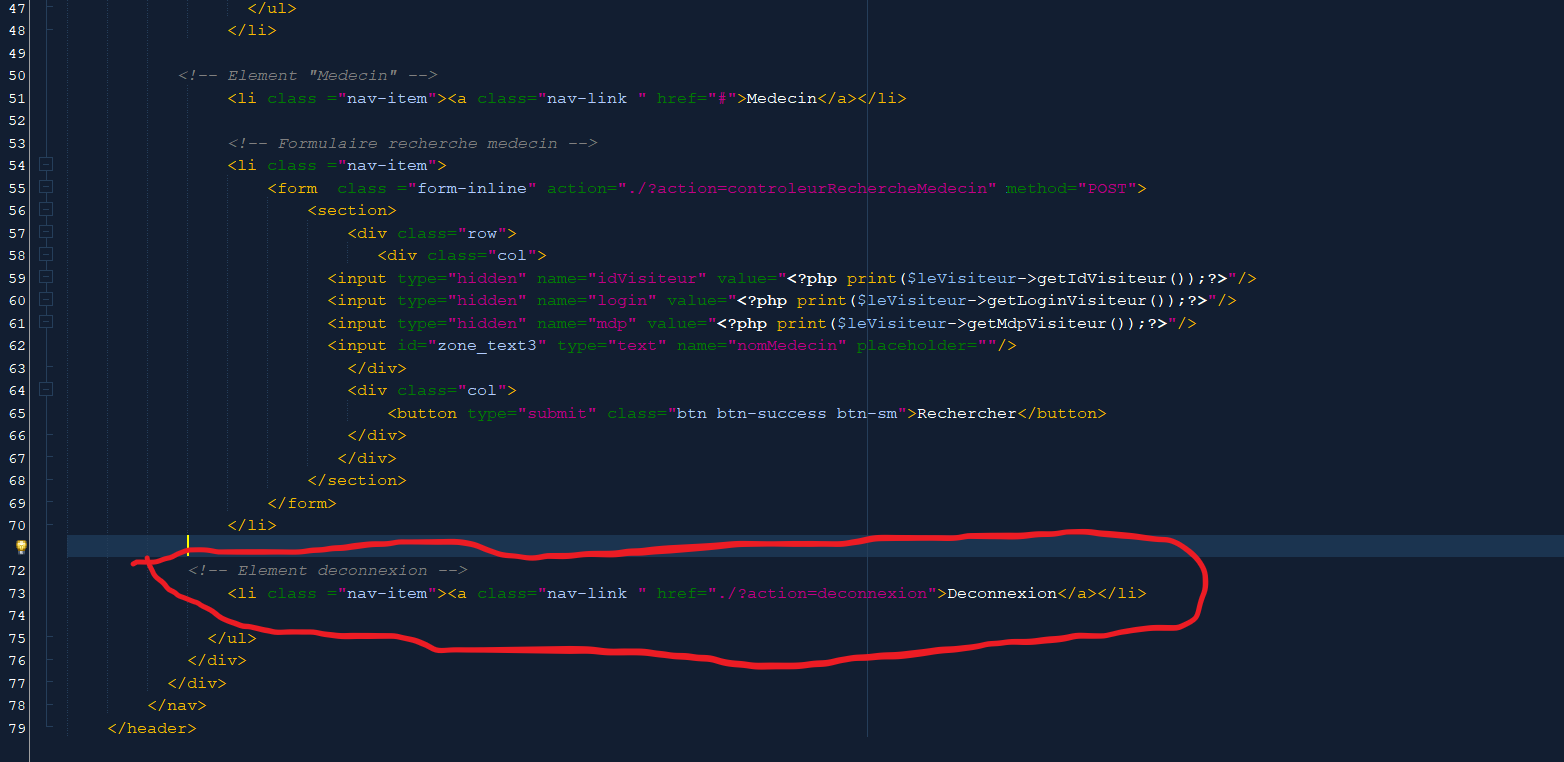


*Extrait de code (entete.html.php)*

Une image contenant texte

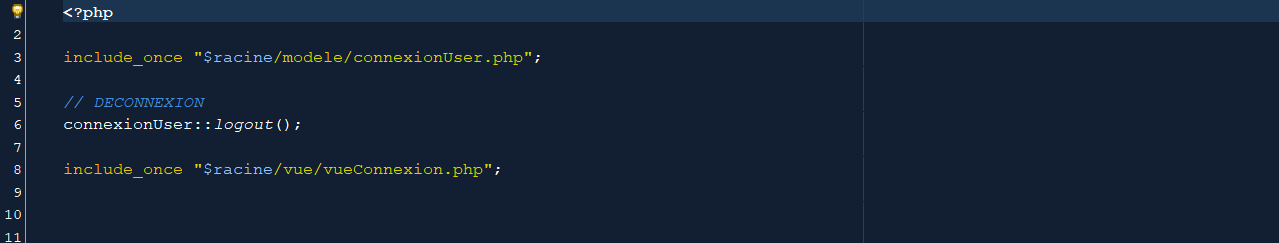
Description générée automatiquement

Une image contenant texte

Description générée automatiquement

1. **Contrôleur:**

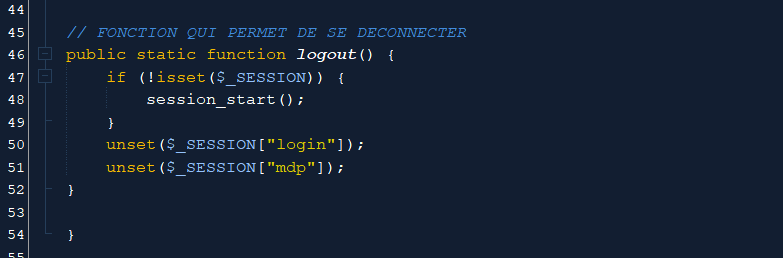
*Extrait de code (deconnexion.php)*



Dans ce contrôleur nous faisais appel à la fonction logout() de la classe connexionUser pour réaliser la déconnexion. Ensuite nous incluons la vue « *vueConnexion.php* » pour diriger le visiteur vers la page de connexion.

1. **Modèle:**

*Extrait de code ( fonction logout() / connexionUser.php)*



*En cas de question ou d’incompréhension vous pouvez me contacter sur mon mail : alphabalde0102@gmail.com*