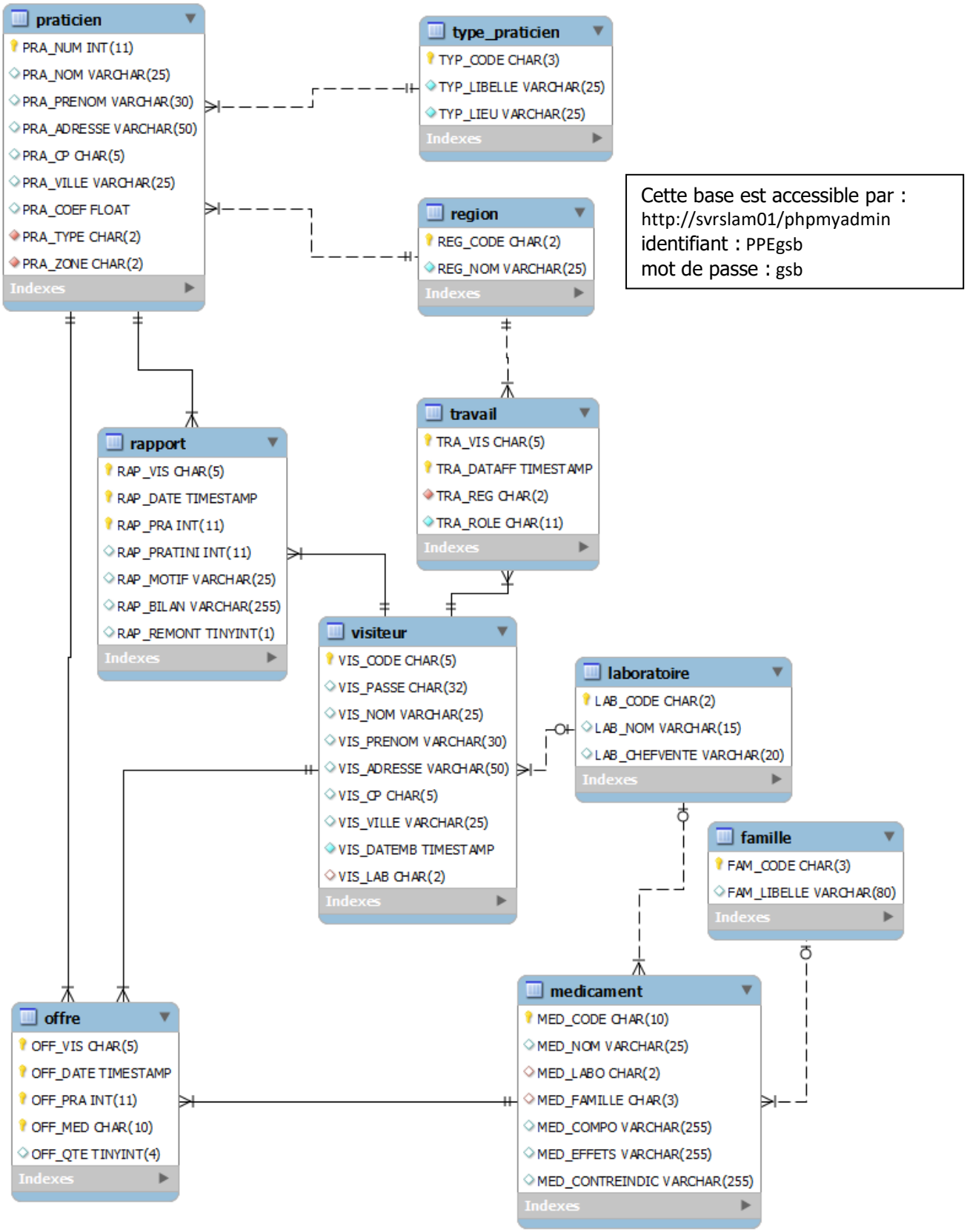
La base de données gsb



Cette base est accessible par :
http://svrslam01/phpmyadmin
identifiant : PPEgsb
mot de passe : gsb

L'authentification

Le site que vous êtes en train de développer est un **extranet**, un site destiné aux collaborateurs situés à l'extérieur d'une entreprise. Inversement on appelle **intranet**, un réseau d'entreprise utilisé à l'intérieur d'une entreprise ou de toute autre entité organisationnelle reprenant les standards internet.

Un extranet ne doit être accessible que par des utilisateurs autorisés. Il est indispensable que son accès soit protégé efficacement par un système de comptes d'utilisateurs protégés par mot de passe.

Une fois connecté à l'application, l'utilisateur devra disposer d'une option de menu pour se déconnecter quand son travail sera terminé.

1 Le stockage des mots de passe dans la base de données

Dans une base de données les mots de passe ne doivent pas être stockés tels quel pour des raisons de sécurité. A la place on leur applique un traitement nommé "hachage" qui calcule une **empreinte** du mot de passe. Une fonction de hachage (*hash function* en anglais) prend des données en entrée (un mot de passe, ou des données binaires comme une image par exemple) et calcule une chaine de caractères de taille limitée ou fixe : l'empreinte.

C'est cette empreinte qui sera stockée dans la base de données. Comme il n'est pas possible à partir de l'empreinte de retrouver la donnée initiale, on déterminera si un mot de passe saisi est valide en comparant son empreinte à celle stockée dans la base de données.

Plus d'informations à la page : `http://www.culture-informatique.net/cest-quoi-hachage/`

Dans la base de données `GSB`, ce sont bien les empreintes des mots de passe, hachés avec l'algorithme MD5, qui sont stockées dans la table `visiteur`.
Pour cette première application, et bien que la documentation de PHP précise que son utilisation n'est plus recommendée, vous utiliserez la fonction de hachage `md5()`, qui retourne une empreinte de 32 caractères.

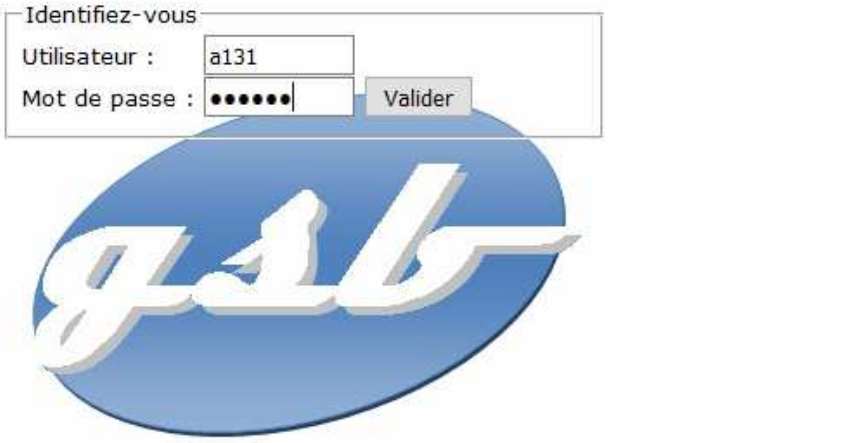
C'est le code visiteur qui servira d'identifiant à chaque visiteur pour se connecter. Demandez à votre chef de projet le mot de passe en clair de l'utilisateur que vous utilisez pour vos tests.

L'application doit prendre en compte 3 demandes :

1. affichage de la page d'identification ;
2. validation de l'identifiant et du mot de passe saisis ;
3. fermeture de la session de l'utilisateur.

2 Première demande : affichage de la page d'identification

C'est dans cette page que l'utilisateur va saisir son identifiant et son mot de passe :



Ajoutez un nouveau fichier nommé `Identif.php` à votre projet, sans pour l'instant écrire de HTML. Cette page n'inclut pas l'entête des autres pages du site puisqu'elle ne comporte pas de menu.

Sa présentation est particulière et va faire l'objet d'une autre feuille de style à nommer `gcr2.css`.

Pour l'image de fond, ajoutez pour la balise `body` (le chemin relatif de l'image doit être donné à partir de l'emplacement de la feuille de style) :

```
background-image:url(../Images/logo.jpg);
```

Il faut empêcher la répétition de l'image et la positionner :

```
background-repeat:no-repeat;
background-position:50% 0%;
```

On choisi une police de caractères adaptée à un affichage sur écran, dans une taille un peu inférieure à la normale :

```
font-family:Verdana,sans-serif;
font-size: 0.8em;
```

Une fois le logo positionné, on peut passer au codage de la page. Elle n'affichera pas le menu de l'application et utilisera donc un entête spécifique. Placez le code suivant dans un fichier nommé `entete2.inc.php` :

```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="UTF-8" />
        <link rel="stylesheet" href="./Styles/gcr2.css" />
        <title>GSB : GCR Gestion des comptes rendus de visite</title>
    </head>
    <body>
```

Incluez ce fichier dans votre page.

Ajoutez le formulaire (nommé `frmIdentification`) permettant à l'utilisateur de s'identifier. Il doit être soumis en utilisant la méthode `post`.
La saisie de toutes les informations est obligatoire. HTML 5 a ajouté l'attribut `required` (`required="required"`) à la balise `<input>`.

Modifiez votre fonction `formInputText()` en lui ajoutant un paramètre booléen qui permettra de gérer l'ajout ou non de cet attribut au code généré par la fonction. Une fois cette modification effectuée, corrigez les appels à la fonction car un paramètre supplémentaire est maintenant attendu.

2.1 Saisie du mot de passe

La saisie d'un mot de passe doit s'effectuer en utilisant un contrôle de saisie `<input>` de type `password`. Ajoutez à votre bibliothèque une fonction nommée `formInputPassword()` qui retourne le code HTML d'un tel contrôle de saisie. Elle prendra en paramètre :

- le label de la zone, - le nom de la zone, - son id,	- sa valeur, - sa taille, - sa longueur maximale,	- son index de tabulation, - un booléen qui indique si la saisie est obligatoire ou pas.

Elle fonctionnera presque de la même manière que la fonction `formInputText()`, donc prenez modèle sur cette fonction.

Terminez ensuite le codage des contrôles de saisie du formulaire.

2.2 Mise en forme du formulaire

Ajoutez les directives suivantes dans votre feuille de style afin de centrer le formulaire sur la page et de dimensionner ses labels :

```
#frmIdentification {
    width:350px;
    margin: 0 auto ;
}

label {
    display: inline-block;
    width:100px;
}
```

Lorsque le formulaire sera soumis, une demande devra être adressée au point d'entrée de l'application pour valider les informations saisies. A vous de préciser l'attribut `action` du formulaire, et d'adapter le fichier `index.php`.

ATTENTION ! Veillez à ce que l'affichage de la page d'identification soit systématiquement effectué lorsque `index.php` ne reçoit aucun paramètre ou reçoit une valeur d'action inconnue.

3 Deuxième demande : validation de l'identifiant et du mot de passe saisis

Ajoutez à votre projet un fichier nommé `Securite.inc.php`. Commencez le fichier par un appel à `session_start()`.

Ajoutez à ce fichier une fonction nommée `valideInfosCompteUtilisateur()` qui retourne un booléen indiquant si le code du visiteur et le mot de passe passés en paramètre sont valides.

Remarques : la fonction hachera ce dernier en MD5 à l'aide de la fonction `md5()`, avant d'appeler une fonction nommée `existeCompteVisiteur()` qui verifiera la présence des informations dans la base de données. Utilisez `rowCount()` pour exploiter le résultat de la requête ...

3.1 Cas 1 : la validation est négative

Il faut réafficher la page d'identification (le nom d'utilisateur saisi devra être affiché par défaut) avec un message d'erreur ☞

Ce dernier doit être contenu dans un paragraphe de la classe `erreur`, intégré au formulaire. Dans la feuille de style ajoutez les directives suivantes :

```
.erreur{
    background-color:rgb(237,210,229);
    color:rgb(203,28,128);
    width:100%;
    text-align: center;
}
```

3.2 Cas 2 : la validation est positive

Lorsque les informations d'ouverture de session sont validées, il faut conserver la trace de cette validation tant que l'utilisateur ne s'est pas déconnecté de l'application. Ceci afin de pouvoir vérifier que chaque demande de traitement reçue émane bien de l'utilisateur authentifié.

Pour conserver des informations d'une page à l'autre on utilise des variables de **session**. C'est pour utiliser les sessions que vous avez débuté `Securite.inc.php` par un appel à `session_start();`

Ici, une fois l'utilisateur validé, vous stockerez dans la variable de session `$_SESSION['utilId']` le code utilisateur saisi pour ouvrir la session. Ensuite à chaque demande de traitement, il suffira dans le point d'entrée unique de l'application de vérifier si cette variable existe ou pas.

Donc :
- aoutez à `Securite.inc.php` une fonction nommée `ouvreSessionUtilisateur()` qui crée la variable de session `utilId` à partir de l'identifiant qu'elle reçoit en paramètre.
- Modifiez `Identif.php` afin que cette fonction soit appelée une fois les informations d'ouverture de session de l'utilisateur validées.

3.2.1 Redirection vers le point d'entrée unique

Il faut ensuite afficher la page du menu général. L'affichage de cette page est une demande que doit prendre en compte l'application. L'affichage de la page s'effectue en appelant le fichier d'entête suivi du pied de page.

Pour que la page s'affiche automatiquement après la création de la variable de session, il faut effectuer une **redirection**. La fonction `header()` est utilisée pour cela.

L'exemple suivant redirige l'utilisateur vers `index.php` :

```
header("location: index.php");
```

L'exemple suivant redirige l'utilisateur vers `index.php` en lui passant un paramètre :

```
header("location: index.php?valeur=50");
```

ATTENTION ! La fonction `header()` envoie un entête HTML dans la page en cours de création, ce qui signifie qu'aucune information ne doit être envoyée avant l'appel à cette fonction. Il vous faut donc structurer votre code en tenant compte de cette contrainte …

N'oubliez pas non plus de passer la bonne valeur d'`action` au point d'entrée de l'application.

Dernier point : faites suivre l'appel à `header()` d'un appel à `exit()` afin que le code php qui reste ne soit pas interprété pour rien une fois la redirection effectuée.

3.2.2 Répondre aux demandes uniquement dans le cadre d'une session

Une fois la redirection effectuée, il reste à modifier `index.php` afin que les demandes de traitements ne soient exécutés que si elles émanent d'utilisateur validé (donc qui a bien ouvert une session).

- Ajoutez à `Securite.inc.php` une fonction nommée `estSessionUtilisateurOuverte()` qui retourne un booléen indiquant si la variable de session `utilId` existe.
- Modifiez `index.php` pour qu'il vérifie systématiquement qu'une session est ouverte avant de lancer le traitement demandé. Si aucune session n'est ouverte et si on ne demande pas la validation des informations d'identification il faudra afficher systématiquement la page d'identification.

4 Troisième demande : fermeture de la session de l'utilisateur

La fermeture de session doit se traduire dans l'application par la suppression de la session et l'affichage de la page d'identification.

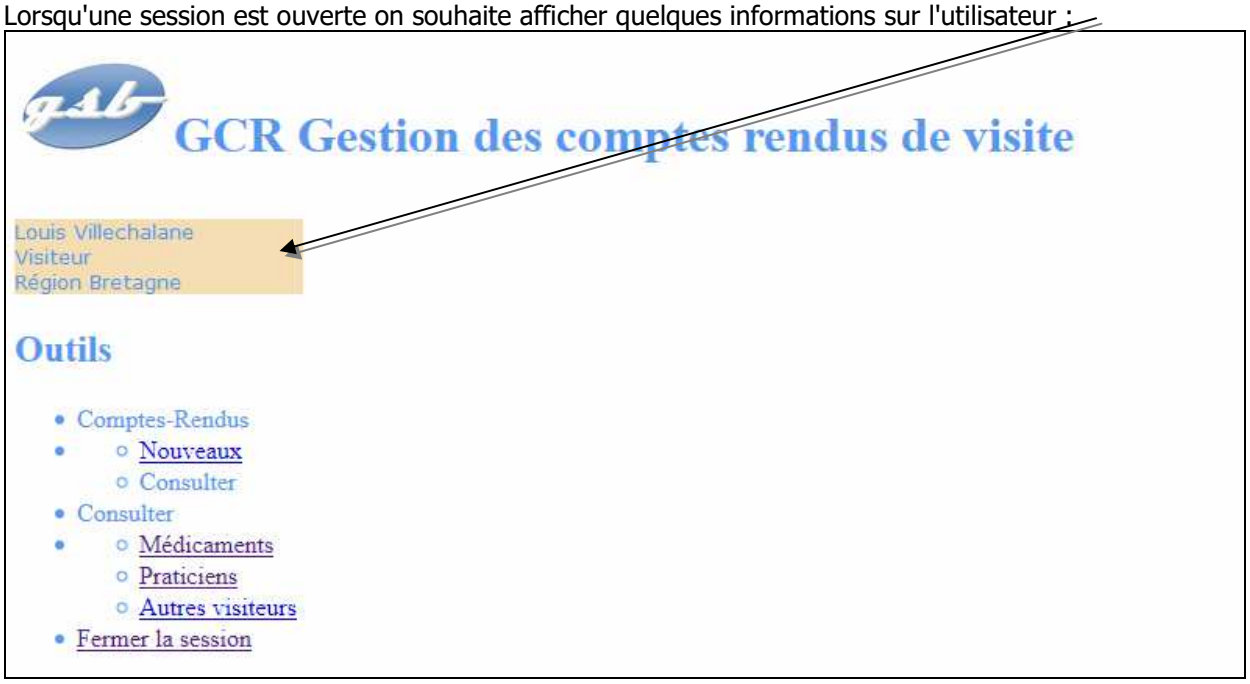
Ajouter dans `Securite.inc.php` la fonction `fermeSessionUtilisateur()` qui effectue la suppression de la session. Pour cela utilisez la fonction PHP `session_destroy()`.

Ajoutez au menu général de l'application l'option "Fermer la session".

Modifiez `Identif.php` et `index.php` pour prendre en compte la demande de fermeture de session : il faudra supprimer la session et demander l'affichage de la page d'identification.

5 Afficher des informations sur le visiteur

Lorsqu'une session est ouverte on souhaite afficher quelques informations sur l'utilisateur :



Pour cela :

- commencez par mettre au point la requête SQL nécessaire et créez la fonction adéquate dans `sourceDonnees.inc.php`. Si vous n'obtenez pas un rôle de visiteur, c'est une erreur, demandez un autre identifiant à votre chef de projet. Dans votre requête, tenez bien compte du fait qu'il peut y avoir plusieurs affectations successives et sélectionnez la plus récente.
- Modifiez la fonction `ouvreSessionUtilisateur()` afin qu'elle récupère les informations sur l'utilisateur et les place dans autant de variables de session.
- Modifiez votre fichier d'entête afin d'afficher ces informations. Placez-les dans un paragraphe qui précèdera le titre "Outils". Donnez à votre paragraphe l'identifiant "`infosUtil`".
- Ajouter les directives suivantes à votre feuille de styles :

```
#infosUtil {
    font-family:Verdana,sans-serif;
    font-size: 0.8em;
    width: 95%;
    background-color: wheat ;
}
```

Testez votre application.

☠ **Un dernier problème** : vous êtes un utilisateur mal intentionné et vous tapez l'url suivante :

`http://localhost/GCR/Index.php?action=10&txtUtilisateur=`<votre nom d'utilisateur>`&pwMdP=`<votre mdp>

… où `txtUtilisateur` et `pwMdP` sont les noms (`name`) des zones de saisie de votre formulaire `frmIdentification` et <votre nom d'utilisateur> et <votre mdp> sont les informations visiteur pour l'ouverture de session qui vous ont été communiquées.

Le logiciel plante. A votre avis pourquoi ?

Pour corriger le problème, modifiez `Identif.php` :
- avant de les valider, testez si le nom d'utilisateur et le mot de passe sont bien passés par la méthode post.
- Si tel n'est pas le cas redirigez l'utilisateur vers la page d'identification.