

**DOSSIER 1 : Répondre par vrai ou faux (05pts)**

- 1-Le grand livre s'obtient après avoir enregistré les opérations dans le journal.
- 2-Pour corriger une erreur dans un livre comptable, il faut enregistrer la même opération avec les montants négatifs
- 3-Le livre-journal est un document qui enregistre chronologiquement toutes les opérations de l'entreprise.
- 4-C'est à partir du grand livre que sont établies périodiquement des balances.
- 5-Selon le Principe de la partie double toute opération simple intéresse deux comptes au moins : si un compte est débité, il y a au moins un compte crédité et inversement.
- 6-Les comptes du bilan se décomposent en comptes d'actif et du passif.
- 7-On distingue uniquement que deux types de comptabilité.
- 8-Les Comptes de l'actif immobilisé réunissent les emplois durables ou les biens destinés à rester, sous la même forme, dans l'entreprise pendant plusieurs exercices.
- 9-En comptabilité le débit représente une ressource et le crédit un emploi.
- 10-En comptabilité la ressource représente la destination et l'emploi l'origine.

**DOSSIER 1 (05 points)**

Stagiaire dans l'entreprise IUA, on vous demande de travailler sur le dossier suivant relatif à la TVA des mois de juin et juillet 2016.

Mois	TVA facturée sur ventes	TVA facturée sur prestations de services	TVA récupérable sur Achats	TVA récupérable sur immobilisations	TVA récupérable sur services extérieurs et autres charges
Juin	500 000	1 000 000	2 000 000	300 000	100 000
Juillet	6 000 000	500 000	1 000 000	150 000	1 500.000

TAF : Pour chaque mois :

- 1) Calculer les montants de la TVA collectée, récupérable et la différence (qualifier le montant de la différence).
- 2) Passer les écritures de déclaration nécessaire pour chaque mois.

**Dossier 3 : Enregistrement dans le journal (10 points)**

TAF : Passer les écritures suivantes dans le journal

01/06/N : l'entreprise GI SARL a acheté à crédit des marchandises pour 25000000f.

- 02/06/N : Achat d'un ordinateur 1000000f réglé par chèque à 50%, 1/4 en espèce et le reste à crédit.

- 03/06/N : Achat d'un brevet au comptant 10 000 000f.

- 04/06/N : Vente de marchandises au client MIAGE SARL 300 000f dont 200 000f payés en espèces et le reste à crédit.

-05/06/N : Achats de matières premières à 3.200.000 F dont 200.000 F payés en espèces et le reste à crédit.

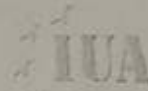
-06/06/N : Retrait de la banque pour alimenter la caisse : 2.800.000 f.

-07/06/N : Règlement des salaires du mois : 10.000.000 F dont 7.500.000 F par chèque et le reste en espèces.

-08/06/N : Achat d'un camion de livraison à crédit 25.000.000F (facture N°T46).

-09/06/N : Emission d'un emprunt obligataire remboursable sur 10 ans : 50.000.000.

-10/06/N : Remboursement du découvert de 5.000.000 F par avis bancaire N°16.



**EXAMEN**

Classe : **Licence 2 GI**

Matière : **Langage C++**

Durée : **1H10**

**EXERCICE 1: (10 points)**

Soit la matrice m1 ci-dessous :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

Ecrivez un programme qui affiche la matrice m1, son déterminant, la matrice inverse  $m1^{-1}$  et la copie m2 de la matrice m1.

Le code devra contenir une classe matrice qui intégrera un constructeur par défaut, un constructeur par copie, une fonction d'inversion de matrice, une fonction de calcul du déterminant et une fonction d'affichage. L'utilisateur devra saisir les coordonnées de la matrice m1.

**EXERCICE 2: (10 points)**

Ecrire un programme permettant de calculer la surface et le périmètre d'un cercle.

Le code doit contenir une *classe cercle* qui intégrera un *constructeur par défaut*, un *constructeur par copie*, un *destructeur*, une fonction de calcul de surface, une fonction de calcul du périmètre. Dans le programme principal, déclarer deux objets de la classe cercle puis afficher leurs surfaces et leurs périmètres respectifs. Le rayon de chaque objet sera saisi par l'utilisateur.

Rappel : surface du cercle =  $r^2 \cdot \pi$  ; périmètre du cercle (circonférence) = diamètre  $\cdot \pi$  ;  
On prendra  $\pi = 3.14$ .



**« Qui ne planifie pas, planifie son échec »**

<b>Epreuve : INITIATION AUX METHODES AGILES</b>	<b>IUA 2023-2024</b>
<b>Filière: L2 GI</b>	<b>Durée : 02 heures</b>

**Exercice 1**

- 1- Citer les éléments de contrainte de gestion d'un projet.
- 2- Citer trois outils et trois méthodes de gestion de projet.
- 3- Quelle différence faites-vous entre une mêlée et une revue de sprint ?
- 4- Quel est rôle fondamental que joue le SCRUM MASTER dans la gestion de projet ?
- 5- Le manifeste Agile respecte 4 valeurs fondamentales à observer scrupuleusement lors de la gestion d'un projet. Complétez ce tableau selon ces valeurs.

<b>Nous privilégions...✓</b>	<b>plutôt que...✗</b>
les individus et leurs interactions	
des logiciels opérationnels	
la collaboration avec les clients	
l'adaptation au changement	

**Exercice 2 : Associez uniquement la ou les bonne (s) réponse (s) au numéro de question.**

**1. Lequel de ces éléments n'est pas une caractéristique des équipes Scrum?**

- ☐ A Auto-organisation
- ☐ B Experts
- ☐ C Polyvalents
- ☐ D Méchant

**2. Qui peut être un utilisateur mandataire (User Proxy)?**

- ☐ A Scrum Master
- ☐ B Coach agile
- ☐ C Développeur
- ☐ D Product Owner

**3. Qui est responsable de la définition des fonctionnalités, de la gestion des contributions des utilisateurs finaux et de la gestion des parties prenantes ?**

- ☐ A Scrum Master
- ☐ B Product Owner
- ☐ C Équipe de développement
- ☐ D Toutes les réponses sont vraies

## « Qui ne planifie pas, planifie son échec »

**4. Comment le Backlog de produit est-il organisé ?**

- A Les éléments sont disposés de manière aléatoire
- B Les éléments les plus récentes sont placées en tête de liste, suivies des plus anciennes.
- C Les éléments articles en haut, les petits éléments en bas
- D Les éléments de haute priorité en haut, suivis par les éléments de basse priorité.

**5. Qui est chargé de veiller à ce que les valeurs et les piliers de la méthode Scrum soient respectés à tout moment ?**

- ☐ A Product Owner
- ☐ B Scrum Master
- ☐ C Équipe de développement
- ☐ D Chaque membre

**6. Quel backlog contient les user stories sur lesquelles une équipe peut travailler ?**

- ☐ A Le backlog de l'équipe
- ☐ B Le backlog du système
- ☐ C Le backlog du produit
- ☐ D Le backlog du programme

**7. Quels sont les différents événements Scrum ?**

- ☐ A Stand-up, codage en groupe, affinement du backlog, rétrospective du Sprint.
- ☐ B Scrum quotidienne, codage en groupe, affinement du backlog, rétrospective du Sprint.
- C Scrum quotidienne, revue du Sprint, affinement du backlog, rétrospective du Sprint.
- D Scrum quotidienne, revue du Sprint, planification du Sprint, rétrospective du Sprint.

**8 : Un projet c'est :**

- a) Une activité qui se situe dans le triangle qualité cout délai
- b) Penser avec d'exécuter
- c) Une activité sans fin déterminée à l'avance
- d) Un déroulement jalonné d'étapes

**9. Parmi les affirmations suivantes, donnez les deux qui se vérifient le plus souvent dans un projet**

- a) En début de projet, on connaît assez bien les besoins du client pour prendre les bonnes décisions
- b) En début de projet on peut tout décider mais on ne connaît pas grand-chose du projet
- c) En début de projet, on connaît le montant qui sera dépensé mais l'argent n'est pas encore disponible
- d) En fin de projet on connaît tout du projet, mais on n'a plus le pouvoir de décision

**10. Parmi les qualificatifs suivants cochez celui qui est incompatible avec la notion de projet :**

- a) Permanent
- b) Novateur
- c) Complexe
- d) Evolutif



**EXAMEN**

Classe : Licence 2 GI

Matière : Théories de la Programmation Orientée Objet

Durée: 1 H

**Documents et internet non autorisés**

**EXERCICE 1: (10 points)**

Soit la classe **Entreprise** et la classe **Personne** Structurée avec les propriétés nom et salaire. Ces deux variables doivent être déclarées private de façon à encapsuler leurs valeurs. Dotez la classe **Entreprise** d'une méthode qui permet d'augmenter le salaire d'une personne selon un certain coefficient lorsque son salaire est inférieur à 1 500 000 F CFA.

**EXERCICE 2: (10 points)**

Définissez une classe **Equation** dans le but de calculer les racines d'une équation du second degré quelconque. Vous devez donner la possibilité à l'utilisateur de choisir les valeurs des coefficients a, b et c de l'équation  $ax^2+bx+c=0$ .

Prévoir les méthodes **calcul\_racines()** et **donne\_resultats()**.



**Consignes :**

- Cette épreuve comporte 2 pages
- Durée : 02H00
- Documents non autorisés
- Ordinateurs et appareils connectés non autorisés

**Partie I : Vérification des connaissances (08 pts)**

1. Que signifie Java EE et dites en quelques mots à quoi sert Java EE (1.5 pts)
2. Citez deux grands types de pages Java EE qui permettent de réaliser des sites web dynamiques (1 pt)
3. De quoi avons-nous besoin exactement pour développer une application Java EE ? (1 pt)
4. Qu'est-ce qu'un bean entreprise ? (1.5 pts)
5. A la question, une servlet ou une jsp pour développer des pages web en java, que pouvez-vous répondre ? (1,5 pts)
6. Définir le service JAAS(1.5 pts).

**Partie II : Pratique du Java EE (12 pts)**

1. Comment appelle-t-on le fichier qu'il faut déployer sur un serveur d'application pour exécuter une application JEE ? (0.5 pts).
2. M. DONI et M. BANCE, tous deux étudiants en GIL2 à IUA, désirent mettre en place une application web en java. M. DONI est très pointu en développement java pendant que M. BANCE est très pointu en html. Pourquoi devriez-vous les conseiller d'utiliser maven et JPQL pour leur application ? (1 pt)
3. La figure ci-dessous est une page développée par Mlle. KOKORA. Après avoir donné la nature de la page correspondante à la figure, relevez les différentes technologies (**parties grisées**) et donnez leur rôle (suivre l'exemple ci-dessous) (4 pts).

C: une directive taglib, permet d'ajouter des actions dans la page jsp

```

A. <%@page import="java.util.Date"%>
B. <%@include file="Bienvenue.jsp"%>
C. <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<!DOCTYPE html>
<html>

<body>
D.     <h1>Page d'authentification : Nous sommes aujourd'hui le</h1><h2><%=new Date()%></h2>
E.     
F.     <% String nom = "M1 MIAGE" ; %>
    <form action='Login' method='POST'>
G.         <c:out value="Bienvenue"></c:out>
        <div class="inputsDiv">
H.             <p><label>login:</label><input type='text' name='login' value='${login}' /><br/></p>
                <p><label>motpasse:</label><input type='password' name='password' ...
        </div>
    </form>
</body>

```

4. Le code ci-dessous écrit par M. NIAMIEN se trouve dans une page nommée **Login.java** ;  
est ce une servlet ou une jsp ? justifiez votre réponse. (0,5 pts)

```

1. String login=request.getParameter("login");
2. String mdp=request.getParameter("password");
3.     if (login == null)login="";
4.     if (mdp==null)mdp="";
5.     HttpSession session=request.getSession(true);
6.     session.setAttribute("login", login);
7.     session.setAttribute("mdp", mdp);
8.     if(login.equals("Boni")&& mdp.equals("Bogui")){
9.         request.getRequestDispatcher("/Accueil.jsp").forward(request, response);
10.    }else{
11.        request.getRequestDispatcher("/index.jsp").forward(request, response);
12.    JOptionPane.showMessageDialog(null,"erreur: login et / ou mot de passe incorrect");
    }

```

5. Expliquez pour chaque ligne de code, ce que M. NIAMIEN a écrit (0.5 pts pour chaque ligne).



EXAMEN

Classe : **L2** Matière : **SQL Server**

Durée : **1H30**
Documents, ordinateurs, téléphones non autorisés
**TRAVAUX PRATIQUES SOUS SQL SERVER :**

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d'étudiants :

- ETUDIANT (numetu, nom, prenom, datenaiss, rue, cp, ville)
- MATIERE (codemat, libelle, coef)
- EPREUVE (numepreuve, datepreuve, lieu, codemat#)
- NOTATION (numetu#, numepreuve#, note)

Remarque : les clés primaires sont soulignées et les clés étrangères sont marquées par #

Ecrire la commande SQL sous SQL SERVER permettant de :

- 1- Créer une base de donnée nommée **baseecole** et les tables **etudiant**, **matiere**, **epreuve**, **notation**.
- 2- Insérer dix lignes d'enregistrements dans chaque table.
- 3- Afficher les moyennes des notes pour les matières (indiquer le libellé) comportant plus d'une épreuve.
- 4- Afficher les moyennes des notes obtenues aux épreuves (indiquer le numéro d'épreuve) où moins de 6 étudiants ont été notés.
- 5- Créer **une vue** qui affiche les moyennes de chaque étudiant dans chacune des matières, la date des sessions 2 prévues dans une semaine.
- 6- Créer **une vue** permettant de d'afficher la moyenne générale de chaque étudiant (Possibilité d'utiliser d'une vue existante).
- 7- Créer **une vue** permettant de d'afficher la moyenne générale de la promotion, la moyenne la plus élevée, la moyenne générale la plus faible . (Possibilité d'utiliser d'une vue existante).
- 8- Créer **une procédure stockée** de mise à jour à **une valeur donnée** toutes les notes comprises entre **deux seuils donnés**.
- 9- Créer **une procédure stockée** permettant l'insertion d'un étudiant.
- 10- Créer **un déclencheur (trigger)** qui affiche le message « Matière(s) supprimées » après suppression d'une matière.

Année scolaire : 2023-2024

Etablissement : IUA

Classe : G1 L2

Enseignant : Dr. Victoire KIE

Date : 11.05.24

Heure : 09H00 -11H00

Durée : 02h00

**Consignes :**

- Documents non autorisés ;
- Ordinateurs non autorisés ;
- Téléphones non autorisés ;

1. Dans quel répertoire se trouve le fichier web.xml et quel est son rôle.
2. Représentez et décrivez un schéma de l'architecture client-serveur, comprenant un client, le service http et le serveur.
3. Quelles sont les technologies principales de JEE ?
4. Pourquoi doit on utiliser maven ?
5. Qu'est-ce qu'un JPA ? donnez ses avantages.
6. Soit le code ci-dessous :

```
@WebServlet("/login")
public class LoginServlet extends HttpServlet {
```

```
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws IOException, ServletException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");
```

```
        // Authentification de l'utilisateur
        if (authentifier(username, password)) {
            // Redirection vers la page d'accueil
            request.getRequestDispatcher("/home").forward(request, response);
        } else {
            // Afficher un message d'erreur
            request.setAttribute("errorMessage", "Mauvais nom d'utilisateur ou mot de
            passe");
            request.getRequestDispatcher("/login.jsp").forward(request, response);
        }
    }
}
```

```
private boolean authentifier(String username, String password) {
```

```
// Code d'authentification de l'utilisateur  
return true; // Remplacer par la logique d'authentification réelle  
}  
}
```

2

- a. Quel est le rôle de cette servlet ?
- b. Quelle méthode HTTP est utilisée par cette servlet ?
- c. Que fait la servlet si l'authentification de l'utilisateur est réussie ?
- d. Que fait la servlet si l'authentification de l'utilisateur échoue ?



**SUJET : PHP**

**Année universitaire : 2023-2024 Session 1- Semestre 4**

**Durée : 2h**

**Exercice 1** (4points)

Mettre en place un système d'authentification permettant de restreindre l'accès d'un site Web à un ensemble d'utilisateurs identifiés et appartenant à une liste établie. La liste des utilisateurs est stockée dans une table (**Membres**) d'une base de données **repertoire**. Cette table possède les attributs suivants : id int (auto\_increment clé primaire) , pseudo varchar(20) , motpasse varchar(20)

- 1 - Ecrire en HTML le formulaire authentication.html, permettant à l'utilisateur de donner son pseudo et son mot de passe.
- 2 - Le programme authentication.php utilise une fonction verif() qui vérifie dans la base de donnée si le pseudo et le mot de passe de l'utilisateur existent. Elle renvoie True ou False selon le résultat et à définir dans un fichier externe.
- 3- autorisation.php Pour vérifier si l'utilisateur est déjà authentifié, nous allons lire son pseudo dans la session. En d'autres termes, si l'utilisateur est identifié, la variable de session \$\_SESSION['pseudo'] existe

**Exercice 2** (2points) Créez une page « calculatrice.php ».

**Exercice 3** (14 points)

Soit le formulaire « formulaire.php » ci -dessous :

- 1) Afficher le récapitulatif des saisies en-dessus du formulaire (sur la même page).
- 2) Développer le code permettant l'insertion des saisies dans la table « **annuaire** » de la base de données « **repertoire** ». Chaque validation du formulaire doit ajouter une nouvelle ligne d'enregistrement dans la table « **annuaire** ».
- 3) Créez une page « **affichage\_annuaire.php** » qui permettra de récupérer les données et ainsi afficher le nom des champs suivi des informations contenues à l'intérieur de la table « **annuaire** ».
- 4) Sur la page « **affichage\_annuaire.php** », préciser :
  - Le nombre d'hommes
  - Le nombre de femmes
  - Le nombre de lignes
- 5) a- Donnez la possibilité de modifier les enregistrements (ouvrant un formulaire pour effectuer les modifications)
- 5) b- Donnez la possibilité de supprimer les enregistrements (avec un message demandant une confirmation).

Ces deux actions doivent être possibles directement via la page « **affichage\_annuaire.php** »



**Questions de cours** (07points)

- 1) Comment installe-t-on laravel 11.x sur un ordinateur ?
- 2) Quel est le rôle des commandes suivantes ?
  - a - php artisan make :controller GestionContactController --resource
  - d - php artisan make :migration create\_annuaire\_table
- 3) comment valider une opération de migration ?
- 4) Que permet la commande php artisan serve?
- 5) Que permet la ligne suivante :Route::get('/', [GestionContactController::class,'index']);.
- 6) Expliquer 3 directives balde de votre choix.

**Exercice 2**(13 points)

Soit la structure de la table **annuaire** de la base **repertoire** ci-dessous :

- |                                 |                                |
|---------------------------------|--------------------------------|
| - id_annuaire (INT, 3, AI - PK) | -codepostal (INT, 5, zerofill) |
| - nom (VARCHAR, 30)             | - adresse (VARCHAR, 30)        |
| - prenom (VARCHAR, 30)          | - date_de_naissance (DATE)     |
| +telephone (INT, 10, zerofill)  | +sexe (ENUM, 'm','f')          |
| - profession (VARCHAR, 30)      | - description (TEXT)           |
| - ville (VARCHAR, 30)           |                                |

- 1) A l'aide de la description de la table ci-dessus

a) compléter le schéma ci-dessous

```
public function up() {
```

```
    Schema::create('annuaire', function (Blueprint $table) {
```

```
        $table->increments('id_news');
```

```
        $table->string('titre',255);
```

```
        $table->timestamps();
```

```
    }); }
```

b) Précisez la commande ayant permis de créer ce schéma.

- 2) Ecrire le script permettant d'enregistrer les informations saisies dans la table **annuaire** à l'aide d'une vue AjoutContact.blade.php

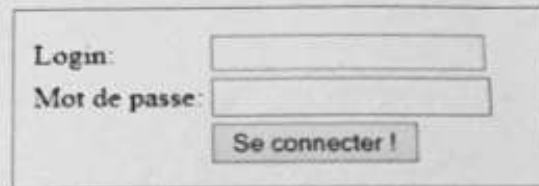
- 3) Afficher la liste de tous les contacts de la table **annuaire** à l'aide d'une balise <table> (ListeContact.blade.php)

Prévoir les options détail et supprimer pour chaque ligne éventuellement retournée.

- 4) Proposer un scripte qui permettra de modifier une information pour laquelle le détail est affiché (DetailContact.blade.php).
- 5) Proposer un scripte permettant de supprimer une ligne (avec un message demandant une confirmation) dans la liste affichée (ListeContact.blade.php).

**Exercice 1**

Ecrire un code php demandant à l'utilisateur de saisir vingt valeurs numériques puis de saisir une valeur t. Il affichera ensuite le nombre de valeurs strictement supérieures à t.

**Exercice 2** : Soit le formulaire login.php

```

    Login: 
    Mot de passe: 
    

```

La page validation.php est la page appelée lors du traitement du formulaire par login.php son rôle est de vérifier que les logins et mots de passe sont corrects. Pour cela, vous ferez appel à un fichier **config.php** à l'aide de l'instruction require. Le fichier config.php contient les instructions suivantes :

```
<?php define('USERLOGIN', 'IUAGIMIAGE') ; define('USERPASS', '2024-php') ; ?>
```

- Si la page validation.php détecte que le login ou le mot de passe est vide, elle renvoie vers la page login.php avec un code erreur 1
- Si la page validation détecte une erreur du login ou du mot de passe, elle renvoie vers la page login.php avec un autre code erreur 2
- Si la page validation constate que le login et le mot de passe sont bons, elle redirige vers la page accueil.php

La page login.php affichera donc un message adapté en fonction de l'erreur appelée

- Erreur 1 : Veuillez saisir un login et un mot de passe
- Erreur 2 : Erreur de login/mot de passe
- La page accueil.php affiche juste le texte « Hello »

**Exercice 3** : Soit le modèle logique de la base de données **Parcauto**:

**vehicule** (immatriculation, chassis, marque, type, annee, #id\_modele, #num\_cni\_pro);

**modele** (id\_modele, nom\_modele, date\_lancement);

**proprietaire** (num\_cni\_pro, nom\_prenom\_pro, date\_naiss\_pro, tel\_pro, adresse\_pro);.

1- Créez un script permettant d'afficher le contenu de la table propriétaire dans un tableau HTML. Les résultats doivent être triés par ordre alphabétique.

2- Ecrivez le script nécessaire permettant de modifier les informations relatives à un propriétaire.

3- Créez un formulaire permettant l'insertion de nouvelles données dans la table propriétaire.

4- Créez un formulaire de recherche permettant de retrouver tous les propriétaires d'un type de véhicule de marque et de modèle donnés. Affichez les résultats sous forme de tableau HTML.



Soit le modèle logique de la base de données **Parcauto**:

```
vehicule (immatriculation, chassis, marque, type, annee, #id_modele, #num_cni_pro);
modele (id_modele, nom_modele, date_lancement);
proprietaire (num_cni_pro, nom_prenom_pro, date_naiss_pro, tel_pro, adresse_pro);
```

1) Ci-dessous la fonction up() après la création du schéma de la relation propriétaire:

```
public function up()
{
    Schema::create(proprietaire, function (Blueprint $table) {
        $table->id();
        $table->timestamps();
    });
}
```

a) donnez la commande ayant permis de créer ce schéma.

b) Compléter ce schéma de sorte à prendre en compte les champs de la table propriétaire.

2) On suppose la vue **Ajoutproprietaire.blade.php** Créée.

a) Quelles raisons justifient l'utilisation de **.blade** dans le nom de cette vue.

b) Compléter la méthode suivante de **GestParcautoController** pour que la vue **Ajoutproprietaire.blade.php** soit la page d'accueil

```
public function create(){
    return view('...');
}
```

c) Complète la ligne ci-dessous du fichier routes/web.php pour préciser la route de la page d'accueil

```
Route::get('...', [.....::class, '.....']);
```

3) On souhaite enregistrer les informations saisies à partir de la vue (**Ajoutproprietaire.blade.php**) dans la table **proprietaire** d'une base de données **Parcauto**.

Créer la méthode store() de **GestParcautoController** et la route « post » qui permettront cette action

NB : - la classe **BD** permet l'utilisation des requêtes CRUD sur un SGBDR

- Utiliser **redirect** pour afficher la vue **Ajoutproprietaire.blade.php** après insertion des informations dans la table **proprietaire**.

4) Afficher la liste des **proprietaires** enregistrées à partir de la vue **Ajoutproprietaire.blade.php** grâce à la vue **listeproprietaire.blade.php** (utiliser la méthode **index()**)

6) Ajouter les codes nécessaires qui permettront de supprimer un **proprietaire**

NB : - On supposera que la base de données **Parcauto** existent déjà.

- Il n'est pas nécessaire de réaliser entièrement les formulaires en HTML / CSS.