

|   |   |  |
|---|---|--|
|    | <b>Atelier développement web<br/>coté serveur</b>                                 | <b>Au : 2024-2025<br/>Classe : L2DSI<br/>Enseignante : Nidhal Cherif</b> |
|   |  | <b>Atelier N°3</b>   |
| <b>Gérer les formulaires et les liens</b>   |   |  |
| <b>Objectifs :</b> <ul style="list-style-type: none"> <li>➤ Gérer la transmission des données via un formulaire (POST)</li> <li>➤ Gérer la transmission des données via l'URL (GET)</li> <li>➤ Contrôler les données saisies via les REGEX</li> <li>➤ Sécuriser Les formulaires contre la faille XSS (Cross-Site Scripting) : htmlspecialchars(), strip_tags()</li> </ul> |   |  |

### Exercice N°1:

1. Ecrire code php permettant d'afficher le formulaire et l'alerte de la figure N°1, sachant que le message alerte ne sera affiché que si

- Les champs sont tous saisis
- L'Email contient le caractère @
- Le bouton Inscription est activé



*Figure N°1*

2. Le bouton **Afficher** permet de parcourir le tableau associatif \$\_POST et de l'afficher sous forme d'une table HTML
3. Sécuriser votre formulaire (htmlspecialchars /strip\_tags)
4. Afficher le tableau avec la fonction **var\_dump** et puis avec la fonction **print\_r**. que remarquez-vous ?

## Exercice N°2 :

1. Ecrire le code du fichier « **Formulaire\_testIP.php** » (voir figure 2) contenant le formulaire et le script permettant de tester la validation d'une adresse IPv4. Pour cela appelez une fonction nommée : **valideIP(\$ip)** déjà définie dans le fichier « **Fonctions.php** ».

La fonction **valideIP(\$ip)** reçoit un seul paramètre, c'est l'adresse IP (\$ip : de type chaîne de caractères) à valider en argument et retourne **true** si celle-ci est valide sinon **false**

Vérification d'adresse IPv4

|     |   |     |   |    |   |   |
|-----|---|-----|---|----|---|---|
| 192 | . | 168 | . | 52 | . | 2 |
|-----|---|-----|---|----|---|---|

Vérifier

192.168.52.2 est une adresse IPv4 valide.

Vérification d'adresse IPv4

|     |   |     |   |    |   |     |
|-----|---|-----|---|----|---|-----|
| 192 | . | 168 | . | 52 | . | 256 |
|-----|---|-----|---|----|---|-----|

Vérifier

192.168.52.256 n'est pas une adresse IPv4 valide !

*Figure N° 2 : Formulaire\_testIP.php (sur le navigateur)*

2. Ecrire le code de La fonction **valideIP(\$ip)** du fichier « **Fonctions.php** », en utilisant la fonction prédéfinie `filter_var(chaîne_à_tester, FILTER_VALIDATE_IP)`.

👉 **filter\_var(chaîne\_à\_tester, FILTER\_VALIDATE\_IP)** : retourne la chaîne de caractères passée en argument si elle représente une IP valide, rien sinon.(false)  
Avec `FILTER_VALIDATE_IP` est un ID (prédéfini) du filtre qui valide une adresse IP

## Exercice N°3 :

Ecrire le code PHP qui permet de:

1. Définir un tableau associatif **TMOTS**, contenant les mots de passe d'un ensemble d'utilisateurs et dont les noms constituent les indices de ce tableau.
2. Écrire le code PHP qui donne à **sélectionner** (à partir d'une liste de choix) le nom d'un des utilisateur et affiche son mot de passe.
3. Écrire le code PHP qui donne à **sélectionner** (à partir d'une liste de choix) le mots de passe d'un utilisateur et affiche ensuite le nom qui lui correspond.

#### Exercice N°4 : Exemple de passage de paramètres via l'URL:

1. Nous voulons faire un lien de **Accueil.php** qui mène à **Bienvenue.php** et qui lui transmet des informations ( prénom et répétition) dans l'URL. Comme le schématise la figure suivante.

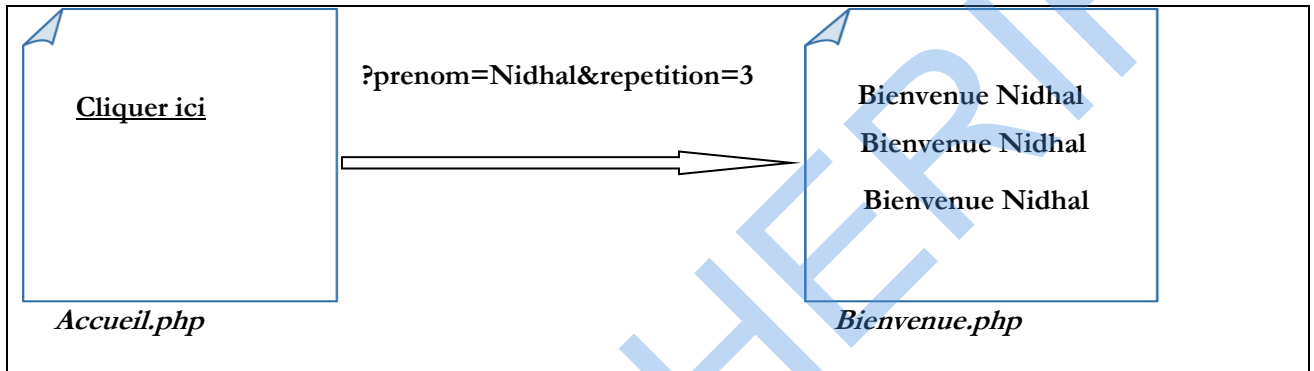


Figure N°3

2. Tester la présence des paramètres transmis avant de les afficher. Et afficher un message en cas d'inexistences des paramètres
3. Ajouter les contrôles suivants
  - La valeur du paramètre `repetition` doit être  $<10$
  - Le type de répétition doit être entier

#### Exercice N°5 :

Dans la perspective de création d'un site d'agence immobilière, créer un formulaire comprenant trois boutons submit nommés « Vendre », « Acheter » et « Louer ». En fonction du choix effectué par le visiteur, le rediriger vers une page spécialisée (`Vente.php` ou `Achat.php` ou `Location.php`) dont le contenu répond au critère choisi ( voir figure N°4)

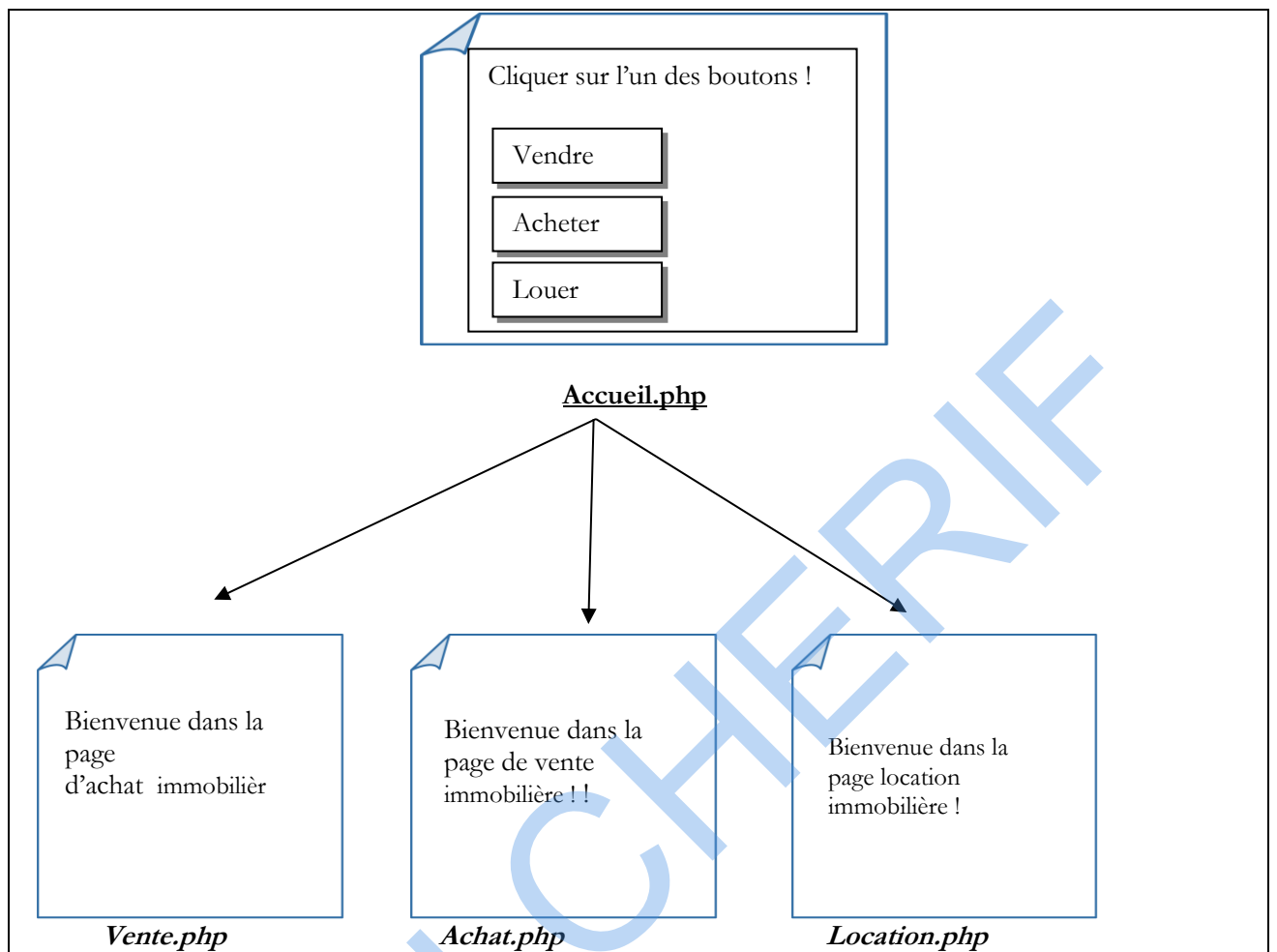


Figure N°4