

Webmastering II

Partie I : Les Éléments de Base du Langage PHP

Sommaire

1. Variables et Constantes
2. Opérateurs
3. Structures de Contrôle
 - o Structures Alternatives
 - if..else
 - switch()
 - les boucles
4. Tableaux
 - o Numériques
 - o Associatifs
 - o Multi-dimensionnels
5. Chaines et Fonctions de Manipulation
6. Super Globaux
 - o Formulaires
 - o SESSION et Cookies
 - o Variables d'URL
 - o Variables Serveur
 - o Variables Globales
7. Fonctions
8. Dates
9. Inclusion des Pages
10. Fichier
11. Mini Projet

source code : https://github.com/douvewane85/cours_webmasteringII_2021

Chapitre I

Variables ,Constantes et Opérateurs

1) Creation et Execution d'un Script PHP

un fichier source PHP peut contenir du :

- PHP entouré des balises `<?php ?>`
- HTML est le langage par défaut
- CSS entouré des balises `<style></style>`
- JAVASCRIPT entouré des balises `<script> </script>`

2) Instructions d'affichage

- `echo()` : affiche une variable de type primitifs ou une constante
- `print()` : affiche une variable de type primitifs ou une constante
- `var_dump()` : affiche une variable ou une constante ainsi que le type

Exemple :

```
<?php  
  
    echo "Cours de WebMastering Semestre 2"; echo"<br>";  
    echo 'Cours de WebMastering Semestre 2';  echo"<br>";  
    print 'Cours de WebMastering Semestre 2';echo"<br>";  
    print "Cours de WebMastering Semestre 2";echo"<br>";  
    var_dump("Cours de WebMastering Semestre 2") ;  
  
?>
```

NB: La fonction `var_dump()` fonction est généralement utilisé pour déboguer une variable

3) **Commentaire:** permet de documenter le code ,on les commentaires:

- Uniligne noté `//`
- Multiligne noté `/**/`

Exemple :

`/*`

NB: Une balise HTML qui est affichée dans un bloc PHP doit être précédé par la fonction `echo` par exemple :
`echo"
" ;`

`*/`

- **Types**
 - **Elémentaire :**
 - `int`
 - `float`
 - `bool`
 - `string`
 - `null`
 - **Composé**
 - `Tableau ou Array`
 - `Objet`
 - `Ressource`
 - `File`
 - `Traits`

NB : Pour connaître le type d'une variable , on peut utiliser les fonctions `is_type()` ou `gettype()`

4) Convention de nommage

- **Variable:** camelCase

Exemple: `$nomVar`

- **Fichier:** Kebab Case

Exemple: `mon.fichier.php`

- **Fonctions:** Snake Case

Exemple: `nom_fonction()`

- **Constantes:** Snake Case Majuscule

Exemple: `MA_CONSTANTE`

5) Variables : \$nomVar

Exemple : Affichage , Variables et Type

```
<?php

//Variables

    $entier=2;//Entier ou int

    $reel=2.5; //réel ou float

    $chaine1="Bonjour"; //chaine ou string

    $chaine2='Bonjour'; //chaine ou string

    $boolean=true; // boolean

// Affichage avec echo

    echo 'La valeur entière est  '.$entier;echo"<br>";

    echo "La valeur reel  est  $reel";echo"<br>";

    echo "La valeur chaine  est      {$chaine1}";echo"<br>";

// Affichage avec print

    print("La valeur chaine  est  $chaine2");

//Affichage Valeurs et Types

    var_dump($entier);echo"<br>";

    var_dump($reel);echo"<br>";

    var_dump($chaine 1);echo"<br>";

    var_dump($chaine2);echo"<br>";

    var_dump($boolean);

?>
```

**NB: L'injection de variables
fonctionne uniquement avec**

les guillemets "" et pas avec les '';

6) Constantes

define("MA_CONSTANTE", valeur) ;

Exercice Application :

```
<?php

define("PI",3.14);

echo "La valeur est ".PI;

echo"<br>";

//Constantes Sytemes

echo 'Separateur de Dossier ' .DIRECTORY_SEPARATOR. '<br>';

//Quelques Constantes Magiques

echo 'Numéro de ligne : ' .__LINE__. '<br>';

//Affiche le chemin du fichier et son nom

echo 'Chemin complet du fichier : ' .__FILE__. '<br>';

//Affiche le nom du dossier qui contient le fichier

echo 'Dossier contenant le fichier : ' .__DIR__. '<br>';

//Affiche à nouveau la ligne où la constante a été appelée

echo 'Numéro de ligne : ' .__LINE__. '<br>';
```


7) Opérateurs

a) Arithmétiques

`+, -, *, /, %`

b) Relationnels

`==, ===, !=, <=, >=, <, >`

c) Logiques

`&& ou and, || ou or, !`

d) Affectation (**`=`**)

e) Affectation Composé

`+=, -=, *=, /=, %=`

f) Incrémentation (**`++`**)

g) Décrémentation (**`--`**)

Remarques

- **Affichage des variables serveur en utilisant la fonction `phpinfo()`**
- **En PHP on peut l'affichage des Erreurs sur `php.ini` en changeant les valeurs des ces deux modules comme suit**
 - **`error_reporting=E_ALL`**
 - **`display_errors = on`**

APPLICATIONS

Exercice I:

Générer deux nombres puis calculer et afficher:

- Somme
- Exponentiel
- Différence
- Produit
- Modulo
- Division
- Carré

TAF ETUDIANT :

Exercice II: Générer deux nombres puis faire leur permutation. On affichera les deux nombres avant et après permutations.

Exercice III : Point: Générer deux points puis calculer et afficher la distance entre les deux points. Un point est caractérisé par son abscisse et son Ordonnée.

Chapitre II

Structures de contrôle

8) Structure conditionnelle

- a) `if(condition) ..else`
- b) `if(condition) ..elseif (condition)`
- c) `switch($var) .. cas`

APPLICATIONS

Exercice : Générer un nombre puis déterminer et afficher son signe

Exercice : Générer la valeur d'un mois(entier) puis afficher sa correspondance en chaîne.

TAF ETUDIANT :

Exercice : Générer deux nombres entiers puis les afficher dans l'ordre croissant et dans l'ordre décroissant

Exercice : Générer trois nombres entiers puis les afficher dans l'ordre croissant et dans l'ordre décroissant

Exercice : soit l'équation $ax+b=0$, générer la valeur de a et de b puis donner les solution de l'équation ci dessus.

Exercice : soit l'équation $ax^2+bx+c=0$, générer la valeur de a , de b et de c puis donner les solution de l'équation ci dessus.

9) Structure itérative

- a) for..
- b) while
- c) do..while

APPLICATIONS :

Exercice 1 ;

Écrire un script qui génère un nombre supérieur à 10 000 puis affiche dans:

- une table HTML ,les nombres premiers sont compris entre 1 et la valeur entrée.

TAF ETUDIANT :

Écrire un script qui génère un nombre supérieur à 10 000 puis affiche dans:

- une table HTML , les valeurs sont inférieures à la moyenne du tableau
- une table HTML ,les valeurs supérieures à la moyenne.

NB: on utilisera la fonction `set_time_limit()` qui permet d'augmenter la durée d'exécution du script.

Chapitre III

Les Tableaux et Chaînes

A) Les Tableaux

1) Tableau Numérique

- **Déclaration**
- **Initialisation**
- **Parcours avec la boucle foreach()**
- **Opérations**
 - **Ajout**
 - **Suppression**
 - **Recherche**
 - **Modification**
- **Quelques Fonctions utiles**
 - **cout()**
 - **array_push()**
 - **array_pop()**

- `array_shift()`
- `array_unshift()`
- `in_array()`
- `explode()`
- `implode()`
- etc.

Exercice Application :

```
<?php
//Déclaration
$array_num=[];
var_dump($array_num);
//Initialisation
$array_num=["Bonjour",'Au Revoir',1,2.5,true];
var_dump($array_num);
echo"Afficher un element du tableau";echo"<br>";
var_dump($array_num[0]); echo"<br>";
var_dump($array_num[4]); echo"<br>";
echo"Parcours du Tableau"; echo"<br>";
foreach ($array_num as $key => $value) {
    echo "{$key}=>{$value}"; echo"<br>";
}

echo"Ajout d'un elt en Fin  du Tableau"; echo"<br>";
array_push($array_num,"Nouvelle valeur 1");
var_dump($array_num);echo"<br>";
$array_num[]="Nouvelle valeur 2";
var_dump($array_num);echo"<br>";
echo"Ajout d'un elt en Debut  du Tableau";
echo"<br>";
array_unshift($array_num,"Nouvelle valeur 1 en debut");
var_dump($array_num);echo"<br>";
echo"Supression d'un elt en Fin  du Tableau";
echo"<br>";
$val=array_pop($array_num);
```

```
var_dump($array_num);echo"<br>";
echo"Supression d'un elt en Debut du Tableau";
echo"<br>";
$val=array_shift($array_num);
var_dump($array_num);echo"<br>";

echo"recherche d'une valeur dans le tableau Tableau";
echo"<br>";
$val=in_array(1,$array_num);
var_dump($val);echo"<br>";
echo"recherche d'une clé dans le tableau Tableau";
echo"<br>";
$val=key_exists(5,$array_num);
var_dump($val);echo"<br>";

echo"Recuperation des clés dans le Tableau";
echo"<br>";
$val=array_keys($array_num);
var_dump($val);echo"<br>";
echo"Recuperation des valeurs du Tableau";
echo"<br>";
$val=array_values($array_num);
var_dump($val);echo"<br>";

echo"Conversion Tableau to String"; echo"<br>";
$val=implode(":",$array_num);
var_dump($val);echo"<br>";
//Tableau Multi dimentions
$arr_multi=[
    [1,2,7,8],
    [1.5,2.4,8.6,8],
];
var_dump($arr_multi);echo"<br>";
```

APPLICATIONS :

Exercice 1 ;

Écrire un script qui génère une valeur supérieure à 10 000 puis crée des tableaux numériques :

- **T1 est associée à l'ensemble des nombres premiers compris entre 1 et la valeur entrée.**
- **T2 à l'ensemble des valeurs qui sont inférieures à la moyenne du tableau**
- **T3 à l'ensemble des valeurs qui sont supérieures à la moyenne.**

Les tableaux T1 , T2 et T3 sont affichés à l'aide de trois Table HTML.

Règle de Gestion

- **RG 1 : On utilisera des tableaux à une dimension**
- **RG 2 :On utilisera des tableaux à deux dimensions**

2) Tableau Associatifs

- **Déclaration**
- **Initialisation**
- **Parcours avec la boucle foreach()**
- **Opérations**
 - **Ajout**
 - **Suppression**
 - **Recherche**
 - **Modification**
- **tri**
 - **bulle**
 - **insertion**
 - **fusion**
- **Quelques Fonctions utiles**
 - **count()**
 - **array_push()**
 - **array_pop()**
 - **array_shift()**
 - **array_unshift()**
 - **in_array()**
 - **explode()**
 - **implode()**
 - **etc.**

Exercice Application :

```
?php

//Declaration

$bien=[

    "id"=>1,

    "reference"=>"Ref001",

    "Description"=>"Lorem ipsum dolor sit amet
consectetur adipisicing elit. At commodi
quia quis. Odit itaque dicta distinctio cumque
placeat possimus iure, amet fugiat rem at voluptas
eligendi saepe accusantium neque doloribus?",

    "prix"=>500000,

    "zone"=>[

        "id"=> 1 ,

        "nom"=> "zone 1"

    ],

    "type"=>"chambre"

];

var_dump($bien);

echo "Affichage de la Reference";echo"<br>";

var_dump($bien['reference']);

echo "Affichage du nom de le zone";echo"<br>";
```

```
var_dump($bien['zone']['nom']);echo"<br>";

//Tableau multi dimensionnel

$arr_bien=[

    [

        "id">1,

        "reference">"Ref001",

        "description">"Lorem ipsum dolor sit amet consectetur
adipisicing elit. At commodi quia quis. Odit itaque dicta distinctio
cumque placeat possimus iure, amet fugiat rem at voluptas eligendi
saepe accusantium neque doloribus?",

        "prix">50000,

        "zone">[

            "id"> 1 ,

            "nom"> "zone 1"

        ],

        "type">"chambre"

    ],

    [

        "id">2,

        "reference">"Ref002",

        "description">"Lorem ipsum dolor sit amet consectetur
adipisicing elit. At commodi quia quis. Odit itaque dicta distinctio
cumque placeat possimus iure, amet fugiat rem at voluptas eligendi
saepe accusantium neque doloribus?",

        "prix">500000,

        "zone">[

            "id"> 1 ,

            "nom"> "zone 2"
```

Birane Baila Wane

Module : Webmastering 2

```
    ],  
    "type"=>"Appartement"  
]  
];  
  
foreach ($arr_bien as $key => $bien) {  
    echo "-----Bien {$key}-----<br>";  
    echo "ID: {$bien['id']} <br>";  
    echo "REFERENCE: {$bien['reference']} <br>";  
    echo "DESCRIPTION: {$bien['description']} <br>";  
    echo "PRIX: {$bien['prix']} <br>";  
    echo "ZONE: {$bien['zone']['nom']} <br>";  
}
```

B) Les Chaînes

- **Déclaration**
- **fonctions utiles**
 - **strlen()**
 - **substr()**
 - **str_replace()**
 - **strpos()**
 - **explode()**
 - **strip_tags()**
 - **stripslash()**
 - **str_word_count()**
 - **trim()**
 - **ltrim()**
 - **rtrim()**

Exercice Application :

```
<?php
    $chaine="Ceci est une chaine de Caractere";
    //Affichage du premier caractere
    echo "Affichage du premier caractere ". $chaine[0]."<br>" ;
    //Modification du premier caractere
    echo "Affichage Modification  ". $chaine."<br>" ;
    $chaine[0]="F";
    echo "Apres Modification  ". $chaine."<br>" ;
    //Quelques Fonctions utiles des Chaines de Caracteres
    echo "Quelques Fonctions utiles des Chaines de Caracteres <br>";
    //Longueur de Chaine
    echo "La longueur de la chaine ".strlen($chaine)."<br>";
    //Extraction d'une partie de la Chaine
    echo "Recuperation des 2 premiers caracteres
".substr($chaine,0,2)."<br>";
    //Remplacer des caracteres de la chaine
    echo "Remplacer tous les e par des a
".str_replace("e","a",$chaine)."<br>";
    //Enlever les espaces en debut et en fin de mot
    echo "Enlever les espaces en debut et en fin de mot
".trim($chaine)."<br>";
    echo "Recherche du mot caractere dans la chaine
".strpos($chaine,"Caractere");
    echo"<br>";
    //Convertir un chaine en tableau
    $arr=explode(" ",$chaine);
    var_dump($arr);
    //Supprimer les balises d'une chaine
    $text = '<p>Test paragraph.</p><!-- Comment --> <a
href="#fragment">Other text</a>';
    echo strip_tags($text);
    echo"<br>";
    //stripslashes - Supprime les antislashes d'une chaîne
    $str = "Avez-vous l\'oreille dure?";
    echo stripslashes($str);
    echo"<br>";
    //Recherche d'une sous chaine dans une chaine

    //Le nombre de Mot de la Chaine
```

```
echo "Le nombre de mots d'une chaine  
".str_word_count($chaine). "<br>" ;
```

APPLICATIONS :

Exercice 1 : Générer des numéros ayant le format suivant LLL-CCC-LCL.

NB : L signifie Lettre et C signifie chiffre.

TAF ETUDIANT

Exercice 2 :

Générer une phrase puis écrire un script qui enlève tous les espaces inutiles de la phrase.

Règles de Gestion

- Les espaces inutiles sont:
 - les espaces en début et fin de chaîne
 - les espace avant un point
 - les espaces successives
 - les espaces avant une apostrophe

Exercice 3:

Écrire un script qui permet de générer un tableau N de mots. Chaque mot ne devrait contenir que 20 caractères. Le script affiche tous les mots du tableau puis détermine et affiche ;

- le mot le plus long et le mot le plus court
- le nombre de mots contenant la lettre « M » (la casse n'est pas tenue en compte).
- le mot qui a le plus de voyelles
- le mot qui a moins de consonne

CHAPITRE IV

Tableaux Superglobaux

Un tableau 'superglobaux', ou variable globale automatique. Cela signifie simplement que cette variable est disponible dans tous les contextes du script. Il n'est pas nécessaire de faire **global \$variable**; pour y accéder dans les fonctions ou les méthodes.

On distingue les tableaux superglobaux suivants:

- **\$_POST**
- **\$_GET**
- **\$_REQUEST**
- **\$_SESSION**
- **\$_SERVER**
- **\$_GLOBAL**

A) Gestion des Formulaires avec \$_POST

● \$_POST

- Formulaire
- Fonctions de validation
 - empty()
 - isset()
 - is_type()
 - filter_var()

Application I:

Réaliser un script qui effectue des opérations arithmétiques(+,-,/,%,*) à partir de deux nombres entrés à partir d'un formulaire.

Exercice d'application : calculatrice.php

Code HTML

```
<!doctype html>
<html lang="en">
  <head>
    <title>Title</title>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">
  </head>
  <body>
```

Birane Baila Wane
Module : Webmastering 2

```
<div class="container mb-5">
    <h2 class="text-info">Calculatrice</h2>
    <form>
        <div class="form-group row">
            <label for="inputName" class="col-sm-2
col-form-label"> Nombre 1</label>
            <div class="col-sm-6 ml-2">
                <input type="text" class="form-control"
name="nbre1" id="inputName" placeholder="">
            </div>
        </div>
        <div class="form-group row">
            <label for="inputName" class="col-sm-2
col-form-label"> Nombre 2</label>
            <div class="col-sm-6 ml-2">
                <input type="text" class="form-control"
name="nbre2" id="inputName" placeholder="">
            </div>
        </div>
        <div class="form-group row">
            <label for="" class="col-sm-2
col-form-label"> Operateur</label>
            <select class="form-control col-sm-6 ml-3" name="op"
id="">
                <option>Addition </option>
                <option>Soustraction</option>
                <option>Division</option>
            </select>
        </div>
        <div class="form-group row">
            <div class="offset-7 col-sm-4 ">
                <button type="submit" class="btn btn-primary"
name="btn_submit" value="btn_egal">=</button>
            </div>
        </div>
    </form>
</div>
<p class="mt-2">
    <?php echo "<strong>Resultat</strong> :{$result}"; ?>
</p>

<!-- Optional JavaScript -->
```

Birane Baila Wane

Module : Webmastering 2

```
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"
integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8ab
tTE1Pi6jizo" crossorigin="anonymous"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper
.min.js"
integrity="sha384-UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF
86dIHNDz0W1" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.mi
n.js"
integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x
0xIM+B07jRM" crossorigin="anonymous"></script>
</body>
</html>
```

Code PHP

```
<?php
    if (key_exists("btn_submit", $_POST)) {
        $arr_error = [];
        $nbre1 =
trim(stripslashes(strip_tags($_POST['nbre1'])));
        $nbre2 =
trim(stripslashes(strip_tags($_POST['nbre2'])));
        $op = trim(stripslashes(strip_tags($_POST['nbre2'])));
        $result = "";
        switch ($op) {
            case "+":
                $result = $nbre1 + $nbre2;
                break;
            case "-":
                $result = $nbre1 - $nbre2;
                break;
            case "*":
                $result = $nbre1 * $nbre2;
                break;
            case "/":
                if ($nbre2 != 0)
                    $result = $nbre1 / $nbre2;
                else
                    echo "Impossible";
                break;
        }
    }
```

```
        case "%":
            if ($nbre2!=0)
                $result=$nbre1%$nbre2;
            else
                echo "Impossible";
                break;
            break;
    }

}
```

?>

Règles de validation du formulaire:

- Les champs nbre 1 et nbre 2 sont Obligatoires
- Les champs nbre 1 et nbre 2 sont des chaîne

B) Gestion de la Connexion avec \$_SESSION

- \$_SESSION et \$_COOKIES
 - Notions de Session et de Cookies
 - Gestion des Erreurs dans un Formulaire

Application I:

Réaliser un système de connexion de déconnexion et d'enregistrement de nouveaux comptes .

Règles de Gestion:

RG1: Un compte de connexion est caractérisé par :

- login
- password
- role (Admin ou Visiteur)

RG2: Le login est un email

RG3: Les utilisateurs sont stockés dans la superglobale \$_GLOBALS

Règles de Validation:

- RV1: Tous les champs sont obligatoires
- RV2: Lors de l'inscription on doit confirmer le mot de passe

Structure du projet

- views
 - login.htm.php
 - register.html.php
 - accueil.visiteur.html.php
 - accueil.admin.html.php
- controllers
 - security.php

C) Gestion des variables d'url avec \$_GET

- **\$_GET**

- Variables d'url
- Inclusion de Pages
 - require et require_once
 - include et include_once
 - ob_start()\$var=ob_get_clean()
- Mise en Place d'un Menu

Application II:

Reprendre l'Application I et y insérer un menu, en utilisant :

- les inclusions de pages
- la notion de présentation ou layout

NB: La variable superglobale **\$_REQUEST** est un tableau associatif qui contient par défaut le contenu des variables **\$_GET**, **\$_POST** et **\$_COOKIE**.

CHAPITRE V

Fonctions et Fichiers

- **Fonctions**
 - **Définition**
 - **Appel**

Exercices Fonctions

Écrire les fonctions de validations suivantes:

- **bisextile():** verifie si une année est bisextile ou pas
- **nbre_jours_un_mois_dans_annee():**retourne le nombre de jours d'un mois dans une année .
- **date_valide():** vérifie si une date est valide ou pas
- **change_format():** formate une date en français ou en anglais
- **date_suivante() :** retourne la date après un nombre de jour
- **date_precedente():** retourne la date qu'il faisait avant ce nombre de jours .
- **est_vide() :** qui vérifie si une valeur est vide ou pas
- **est_entier():** qui vérifie si une valeur est un entier ou pas
- **est_reel():** qui vérifie si une valeur est un entier ou pas
- **est_email():** qui vérifie si une valeur est un emai ou pas
- **est_premier():** qui vérifie si un nombre est premier ou pas
- **est_une_phrase():** qui vérifie si une chaîne est une phrase ou pas
- **enleve_espace_initule():** enleve les espaces inutiles d'une phrase
- **est_numero_tel():**verifie si un numero est valide ou pas
- **genrate_matricule():** permet de générer le matricule suivant le format suivant LLL-LLL-LLL

CHAPITRE XI

Upload de Fichiers, Fichiers et Date

- Tableau SuperGlobal \$_FILE
 - upload de fichiers
- Date
 - `date("Y-m-d H:i:s");`
 -
- Fichiers Json
 - Format Json
 - Array to Json
 - `json_encode()`
 - Json to Array
 - `json_decode()`
 - Récupération des données d'un Fichier Json
 - `file_get_contents()`
 - `file_exists()`
 - Ajout des données dans un Fichier Json
 - `file_put_contents()`
 - Modification des données d'un Fichier Json
 - `file_put_contents()`

Application

I) Définition des Fichiers

- Définir un fichier Json de Bien. Un Bien est caractérisé par:
 - son id
 - son référence
 - description
 - son localité
 - prix
 - nbre de pièces
 - date d'enregistrement
 - son type(Chambre,Appartement,Immeuble , Studio et Villa)
 - ses Tags
 - ses photos
 - le propriétaire
- Définir un fichier json d'utilisation , chaque utilisateur est caractérisé par
 - Nom
 - son login(unique),
 - son password est codé dans le fichier Json
 - son avatar qui stocke le chemin de l'image
 - son rôle avec comme valeurs :
 - Admin
 - Visiteur
 - Client

II) Fonctionnalités

A) Users Stories

- US 1 : Un utilisateur doit pouvoir se connecter avec son login et son mot de passe
- US2 : utilisateur doit pouvoir se déconnecter

- **US3 : Un Admin a la possibilité d'ajouter ou de modifier ou supprimer les données d'un bien**
- **US4: Un Admin peut lister les Biens et les filtrer par type**
- **US5: Un Admin peut publier les biens**
- **US 6 : Un Admin peut lister tous les clients**
- **US 7 : Un Visiteur peut lister le catalogue de biens**
- **US 8 : Un Visiteur peut voir les détails d'un bien**
- **US 9: Un Visiteur peut faire une demande de réservation**
- **US 11: Un Client valide la demande de réservation**
- **US 12: Un Client peut créer un compte**
- **US 13: Un Client peut voir la liste de ses réservations et a la possibilité d'annuler ou de filtrer par état.**
- **US 14: Un Admin peut valider la réservation d'un client**

B) Règles de Gestion

- **RG 1: Un Client peut faire plusieurs réservations**
- **RG 2: Une Réservation est faite par un client**
- **RG3 : Un Bien peut faire l'objet de plusieurs reservation**
- **RG4: Un Réservation est faite sur un Bien**
- **RG5: Lors de la validation d'une réservation d'un client ,toutes les autres réservations faites sur ce même bien seront annulées.**
- **RG6: Lorsqu' une réservation est validée, le bien n'est plus disponible dans le catalogue .**
- **RG7: Annulation d'une réservation n'est possible que si elle est en cours de traitement.**
- **RG8:Le catalogue affiche seul les biens publiés**

C) Règles de Validation

- **RV1: Tous les champs sont obligatoire sauf le champ parcours**
- **RV2 : Le login d'user est unique**
- **RV3: La référence est Généré suivant le format REF-LLL-LLL exemple : RÉF-STU-001**
- **Le mot de passe doit être codé dans le fichier**

- Les images téléchargées doivent être enregistrées dans le dossier upload et l'url dans le champ avatar de l'étudiant ou de l'utilisateur.

III) Tâches de l'application

a) Sprint 1 :

- **Type:Initialisation**
- **Réalisation : Développeur**
- **US01: Création du projet structuré comme suit :**
 - **index.php**
 - **lib**
 - **validation.utils.php**
 - **public**
 - **images**
 - **uploads**
 - **img**
 - **css**
 - **style.css**
 - **views**
 - **front**
 - **layout**
 - **front.htm.php**
 - **catalogue.html.php**
 - **detail.html.php**
 - **admin**
 - **client**
 - **réservation**
 - **layout**
 - **admin.htm.php**
 - **security**
 - **connexion.html.php**
 - **register.html.php**
 - **bien**
 - **add.html.php**
 - **show.html.php**
 - **edit.html.php**
 - **delete.htm.php**
 - **data**
 - **user.json**
 - **bien.json**
 - **src**
 - **models**
 - **user.model.php**

- bien.model.php
- controllers
 - security.controller.php
 - bien.controller.php
- **US02:**
 - Initialiser le fichier user.json avec 1 admins et deux clients.
 - Initialiser le fichier bien.json à 5 biens
- **US03:** Créer le repository Github du projet
- **US04 :** Integration des differentes pages

Source :

Template Front:

Template Admin:

b) Sprint 2 : Gestion de la sécurité

• US1,US2

- Type : Sécurité
- Acteur: Client et Admin
- Réalisation: Développeur

• **US 23:** gérer les autorisations accès sur les pages

c'est-à-dire à part la page de connexion toutes les pages sont accessibles après la connexion.

- Type : Sécurité
- Réalisation: Développeur

Sprint 3 : Gestion des biens

• US3 à US6

- Type : Sécurité
- Acteur: Admin
- Réalisation: Développeur

• US7 à US8

- Type : Sécurité
- Acteur: SuperAdmin
- Réalisation: Développeur

Sprint 4 : Gestion des Réservations

Resources

Trello:

template :

Jours

XVII,XVII

Correction de l'Application