terminale	Thème : BDD

TD 5 : Statistiques joueurs en php et SQL

1 Principe du php : Ecrire du html

1.1 Syntaxe :

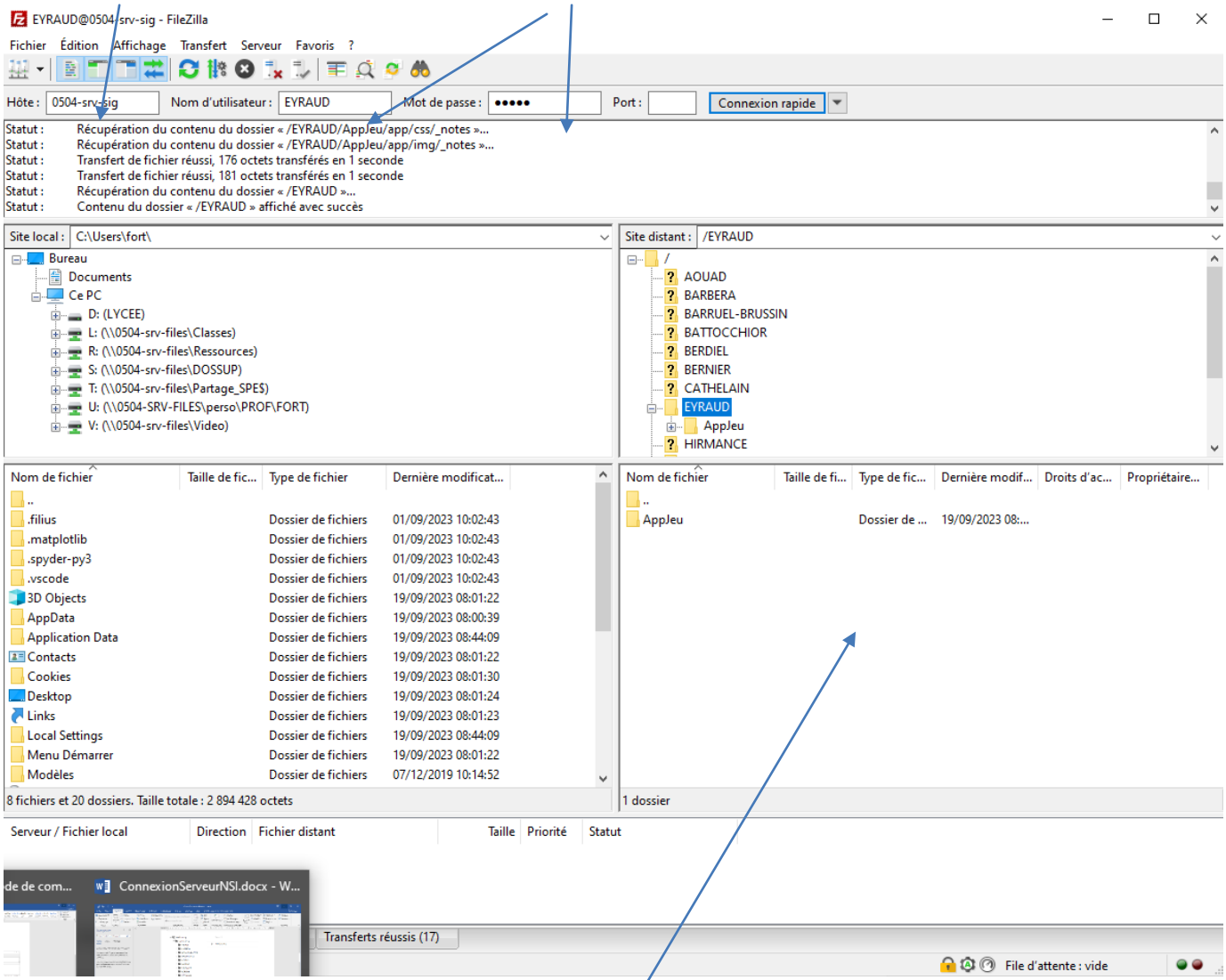
• Le code PHP commence par < ?php (sans espace) et finit par ?>. On va parler de « balise PHP ».
• On peut placer une balise PHP n'importe où dans le code HTML (pas dans du code CSS !)
• Dans une balise PHP on peut mettre autant d'instructions et de lignes PHP qu'on veut.
• On doit mettre un ; à la fin de chaque instruction PHP.
• On peut mettre des commentaires entre /* et */ dans le code PHP. Ces commentaires peuvent s'étaler sur plusieurs lignes. Ces commentaires disparaîtront dans le code HTML.
• On peut mettre des commentaires de fin de ligne avec

1.2 Exemple de l'instruction echo (ou print) : Afficher la date

- Lancer Filezilla ftp client
- Connectez vous au serveur 0504-srv-sig avec vos identifiants de session :

Adresse du serveur

Vos identifiants de session iaca

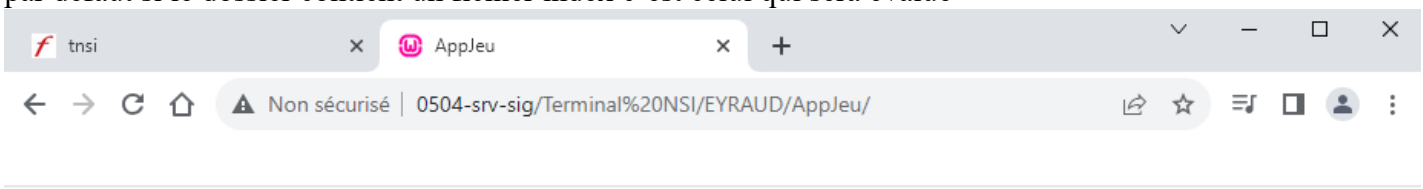


Se connecter puis glisser déposer vos fichiers (dossier AppJeu pour l'exemple) ici

Evaluation d'un fichier php :

Sur le navigateur cliquez sur le dossier AppJeu ou tapez dans la barre d'adresse
http://0504-srv-sig/Terminal%20NSI/NOM/AppJeu/

par défaut si le dossier contient un fichier index c'est celui qui sera évalué

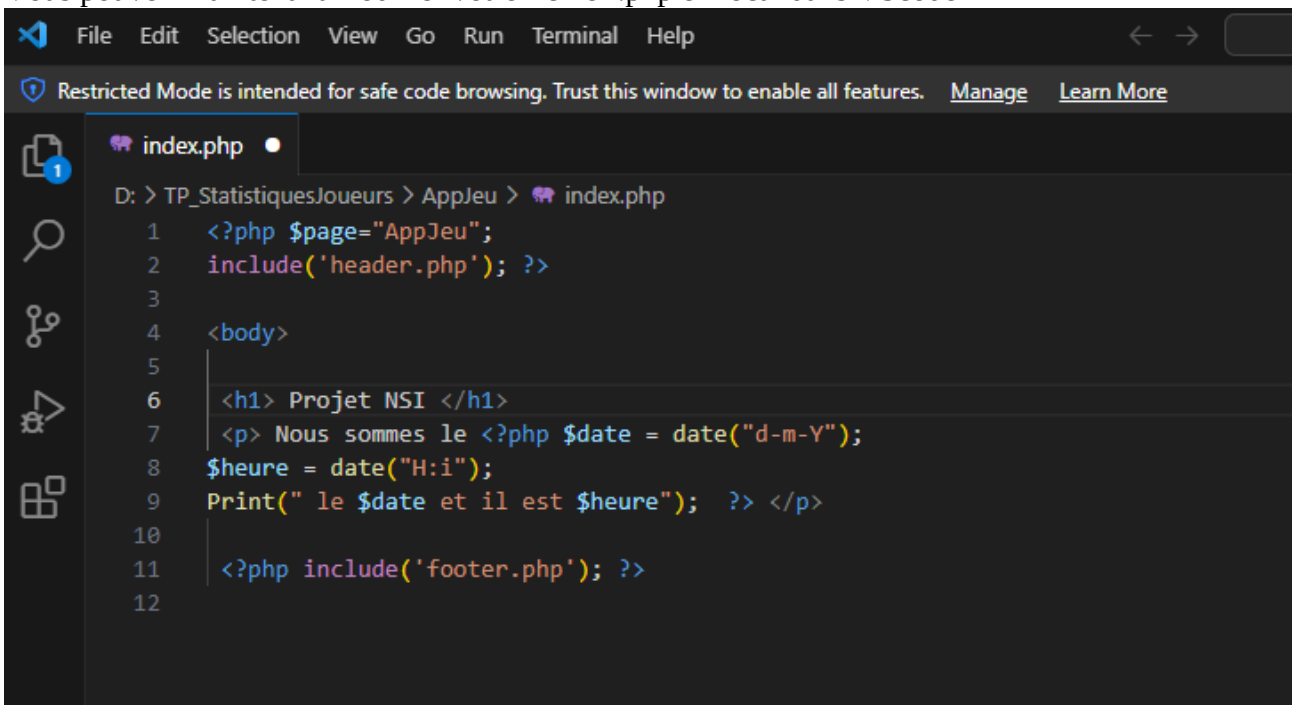


Gros titre

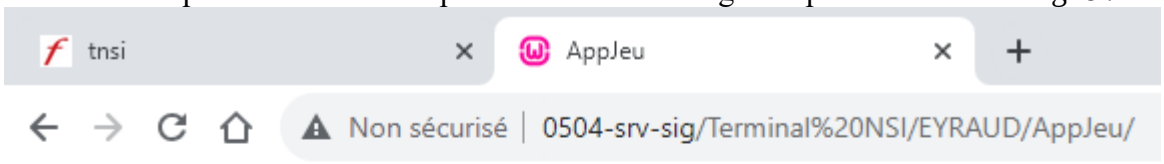
Nous sommes le le 19-09-2023 et il est 06:50

Nous sommes le le 19-09-2023 et il est 06:50

Vous pouvez Maintenant modifier votre fichier .php en local dans VScode



Il faudra le déposer sur le serveur puis actualiser le navigateur pour l'évaluer en ligne :



Projet NSI

Nous sommes le le 19-09-2023 et il est 07:00

Nous sommes le le 19-09-2023 et il est 07:00

Pour évaluer un autre fichier :
http://0504-srv-sig/Terminal NSI/www/AppJeu/app/players.php

1.3 Création d'un header :

Créer un fichier header.php
Recopier le début du
Ajouter dans index à la place du début

```
<?php $page="AppJeu";
include('header.php'); ?>
```

1.4 Création d'un footer :

Créer un fichier header.php
Recopier la fin du code
Ajouter dans index à la place de la fin

```
<?php include('footer.php'); ?>
```

2 BDD : MySQL

Tout d'abord, il nous faut une basse de données. téléchargez **AppJeu.sql**
(http://physalp.free.fr/TNSI/fev_pacques/tpWeb/AppJeu.sql)

- Lancer phpMyAdmin.
- Sélectionnez la base de données,
- Importez la base de données **AppJeu.sql**;

Q1. En cliquant sur les tables et en vous servant de l'onglet structure (et du lien « vue relationnelle » de cet onglet), déterminez le MCD et le schéma relationnel de cette mini base de données.
Q2. Dans l'éditeur SQL, rédigez les requêtes permettant : o d'afficher les parties du joueur « Joseph FOURIER » ;
Q3. d'afficher la liste des parties en cours où le joueur 1 est en attente d'adversaire ;
Q4. d'afficher le nombre de parties non commencées ;
Q5. d'afficher la durée moyenne d'une partie (en secondes) ;
Q6. d'afficher le nombre de parties jouées par « Joseph FOURIER » ;
Q7. d'afficher le nombre de parties remportées par « Joseph FOURIER ».

3 Votre mission

3.1 Analyse

- Copier le dossier app dans AppJeu
- Ouvrir le fichier **players.php** dans visual studiocode
- Il est accessible via http://0504-srv-sig/Terminal NSI/www/AppJeu/app/players.php

Q8. Expliquez à présent le contenu du fichier **players.php**.
Q9. En particulier, expliquez ce qui va se passe au clic sur l'icône « voir » sur chacune des lignes utilisateur.à l'aide des questions 3.2 et 3.3, complétez le fichier **player.php**, lequel permet d'affichez le détail des informations relatives à un utilisateur. On souhaite pouvoir consulter les informations suivantes :
Q10. le détail du joueur (déjà fait) ;
Q11. la liste des parties effectuées par celui-ci ;
Q12. son nombre de victoires et de défaites.
Q13. toute information qui vous semble à propos ;
Q14. toute mise en forme qui vous semble à propre (C'est secondaire !).

3.2 Evolutions

Quoique, en pratique, on se stockerait très probablement pas de telles informations en base de données, on souhaiterait désormais stocker :

Q15. le chemin parcouru par chacun des joueurs au cours d'une partie (évolution 1) ;
Q16. des informations suffisantes afin de pouvoir reproduire le labyrinthe sur lequel les joueurs se sont défiés (évolution 2).

Annexe : players.php (exemple avec l'utilisateur **nsi_eleve1**)

```
<?php
   $user = 'nsi_eleve1';
   $password = 'eleve1';
   $connection = new PDO('mysql:host=0504-srv-sig;dbname=nsi_eleve1', $user, $password);
   $players = $connection->query('SELECT * FROM Player');
?>
<div class="players">
   <table>
      <thead>
         <tr>
            <th rowspan="2">N°</th>
            <th rowspan="2">Prénom</th>
            <th rowspan="2">Nom</th>
            <th rowspan="2">Téléphone</th>
            <th rowspan="2">Mail</th>
            <th rowspan="2">Connecté ?</th>
            <th colspan="2">Dates</th>
            <th rowspan="2">Voir</th>
         </tr>
         <tr>
            <th>Inscription</th>
            <th>Dernière connexion</th>
         </tr>
      </thead>
      <tbody>
      <?php
         foreach($players as $player) {
      ?>
         <tr>
            <td><?=$player['id']?></td>
            <td><?=$player['first_name']?></td>
            <td><?=$player['last_name']?></td>
            <td><?=$player['phone']?></td>
            <td><?=$player['mail']?></td>
            <td>
            <?php
               if($player['connected']){
                  echo 'oui';
               }else{
                  echo 'non';
               }
            ?>
            </td>
            <td><?=$player['subscription_date']?></td>
            <td><?=$player['connection_date']?></td>
            <td>
               <a href="player?id=<?=$player['id']?>">Voir</a>
            </td>
         </tr>
      <?php
         }
      ?>
      </tbody>
   </table>
</div>
<?php
   $connection = null;
?>
```