

DOCUMENTATION TECHNIQUE

GAMESOFT

Etienne COYEN – ECF Studi

Session juin-juillet 2024

SOMMAIRE :

- A. Réflexions initiales technologique sur le sujet
- B. Configuration de mon environnement de travail
- C. Diagramme de classe
- D. Diagramme d'utilisation
- E. Diagramme de séquence

A. Réflexions initiales technologique sur le sujet :

Langage Front-end :

- HTML : c'est la base d'un site web, le langage HTML5 est la dernière version en date.
- CSS : j'ai choisi de l'utiliser dans des pages à part pour séparer le HTML et le CSS et ainsi faire un code plus propre et ordonné afin d'éviter la surcharge de lignes de code dans les pages. Pour faire un site responsive il est impératif d'utiliser le CSS si nous voulons le faire nous-même. Il offre aussi beaucoup plus de possibilité en termes de style. J'ai utilisé CSS3.
- JAVASCRIPT : j'ai choisi de l'utiliser afin d'avoir un site web dynamique et pour ses fonctionnalités. En effet pour utiliser AJAX et JSON, qui permet d'avoir un rendu agréable pour le visiteur.

Langage back-end :

- PHP avec utilisation PDO : j'ai choisi d'utiliser PHP version 8.2 (dernière version en date) pour plusieurs motifs. PHP est un langage open source et facile d'interprétation et d'utilisation, ainsi si ce site doit être utilisé par un tiers à l'avenir il le comprendra assez facilement. Contrairement à d'autres langages, par exemple ASP.NET, il ne demande pas des frais d'hébergement. Enfin j'ai décidé de l'utiliser afin de mieux apprendre la sécurité informatique que je souhaite faire moi-même.

Base de données :

- MariaDB : j'ai choisi de l'utiliser car il est un prérequis obligatoire pour tout développeur web qui se respecte. MariaDB ainsi que MySQL est la base et la référence quand nous parlons de base de données. Je l'ai choisi aussi car c'est une base de données relationnelle et que PlanetHoster le propose dans son phpMyAdmin.

Déploiement :

- PlanetHoster : j'ai choisi cet hébergeur car il offre une possibilité d'héberger gratuitement notre site web et après quelques recherches, il est aussi un hébergeur écologique. Ainsi je réponds à la demande du client ainsi qu'à mes valeurs. De plus PlanetHoster permet de faire évoluer son hébergement à un plus haut niveau selon nos besoins, même si ce n'est plus gratuit.
- PhpMyAdmin : proposer par planetHoster en MariaDB (version 10.5.21 - Apache) avec un PHP en version 7.4.33. Le PhpMyAdmin lui-même est en version 5.2.1.

J'ai mis volontairement aucune Bootstrap, Framework ainsi que le minimum de bibliothèque. J'ai voulu réussir à tout faire par moi-même sans « aide extérieur ». Ainsi j'ai pu voir la difficulté de mettre en place un site web de A à Z, en faisant le responsive ainsi que la cybersécurité.

Sécurité :

J'ai utilisé la base de la sécurité c'est-à-dire :

- Des mots de passes hashés ;
- Une politique de mot de passe (pour les administrateurs la politique est différente), pris la norme directement sur le site du CNIL ;
- Utilisé des requêtes PDO préparé pour éviter des injection SQL ou des failles XSS (htmlspecialchars...)
- Mettre un login et mot de passe à sécurité forte sur la connexion à la base de données ;
- Utilisé des tokens lors de la connexion ;
- Utilisé les jetons CSRF lors des formulaires ;
- Dans ma base de données, j'ai spécifié à chaque fois si c'était une date, INT, décimal, text ou varchar en spécifiant au maximum le nombre de caractère permis ;
- Dans les balises input, j'ai précisé au maximum ce qu'on attendait (exemple : dans un input où on attend un email, type=email).

Exemple d'un code avec de la sécurité :

```
// Vérifier si le formulaire a été soumis
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Vérifier le jeton CSRF
    if(!isset($_POST['csrf_token']) || $_POST['csrf_token'] !== $_SESSION['csrf_token']) {
        die("Invalid CSRF token");
    }

    $email = $_POST['email'];
    $password = $_POST['password'];

    // Vérifier la complexité du mot de passe (20 caractères minimum, une majuscule, une minuscule, un caractère spécial)
    if (strlen($password) < 20 || !preg_match('/[A-Z]/', $password) || !preg_match('/[a-z]/', $password) || !preg_match('/[^A-Za-z0-9]/', $password)) {
        echo "Le mot de passe doit contenir au moins 20 caractères, une majuscule, une minuscule et un caractère spécial.";
    } else {
        // Utilisation de requêtes préparées pour éviter les injections SQL
        $requete = $bdd->prepare("SELECT * FROM administrateurs WHERE email = :email");
        $requete->execute(array("email" => $email));
        $rep = $requete->fetch();

        // Verify the password.
        if ($rep['id'] != false) {
            //c'est ok
            setcookie("email", $email, time() + 3600);
            // Utilisation d'une requête préparée pour mettre à jour le token
            $requeteUpdateToken = $bdd->prepare("UPDATE administrateurs SET token = :token WHERE email = :email");
            $requeteUpdateToken->execute(array("token" => $token, "email" => $email));

            setcookie("token", $token, time() + 3600);

            // On redirige vers la page..
            header("Location: ../espace_admin/dashboard_admin.php");
            exit();
        } else {
            echo "email ou mdp incorrect !";
        }
    }
}
```

Je compte perfectionner au maximum la cybersécurité car c'est important pour l'utilisateur de savoir qu'il peut naviguer en toute sécurité. C'est ceci que je mettrai à jour ma base de donnée si une nouvelle version sort, j'ai aussi mis une alerte mail si on se connecte (réussi ou échec) à ma base de donnée ainsi qu'un nombre de tentative de connexion avant de bloquer l'accès.

J'ai fait le nécessaire pour que les images dans le site soient compressées afin de ne pas ralentir le flux. La sécurité est primordiale mais la vitesse de la connexion aussi. Pour ceci (ainsi que pour l'écologie demander par le client) j'ai choisi une palette de couleurs adaptée à l'écologie, une seule police simple sur tous le site.

Phpspreadsheet est la bibliothèque que j'utilise afin d'exporter des fichiers Excel.

JQuery est la bibliothèque que j'utilise pour Javascript (version 3.6.4).

B. Configuration de mon environnement de travail

Mon IDE :

J'ai choisi comme IDE, Visual Studio qui est pour moi le plus simple éditeur de code ainsi que le plus complet. En effet avec ces nombreuses extensions qui peuvent nous faciliter la vie et ainsi moins faire d'erreur, ce IDE n'est pas négligeable.

Extensions principalement utilisées :

- Better Comments (sert a mettre différents commentaire selon leur importance ou non) ;
- HTMLHint ;
- Indent-rainbow (sert à mieux visualiser ou on en est dans les if, for, while,etc...) ;
- Material icon theme (sert à mettre différents icones sur vos fichiers, dossiers) ;
- Microsoft Edge Developer Tools for VS code (permet de visualiser directement son site sur l'éditeur).

Mon OS :

J'ai utilisé mon ordinateur personnel pour réaliser ce projet, il est en Windows 11 Famille avec une version 23H2.

Mon environnement applicatif :

J'ai déployé mon site web grâce a PlanetHoster qui est simple d'utilisation.

J'ai testé mon site web grâce à XAMPP qui est un serveur local, j'ai pu ainsi tout écrire et tester localement.

J'ai aussi testé avec différents outils mon code :

- Snyk qui est un outil qui permet de voir nos failles de sécurité ;
- J'ai aussi regardé directement dans l'éditeur que propose PlanetHoster qui nous met des icones sur les morceaux de code qui n'ont pas d'utilité.

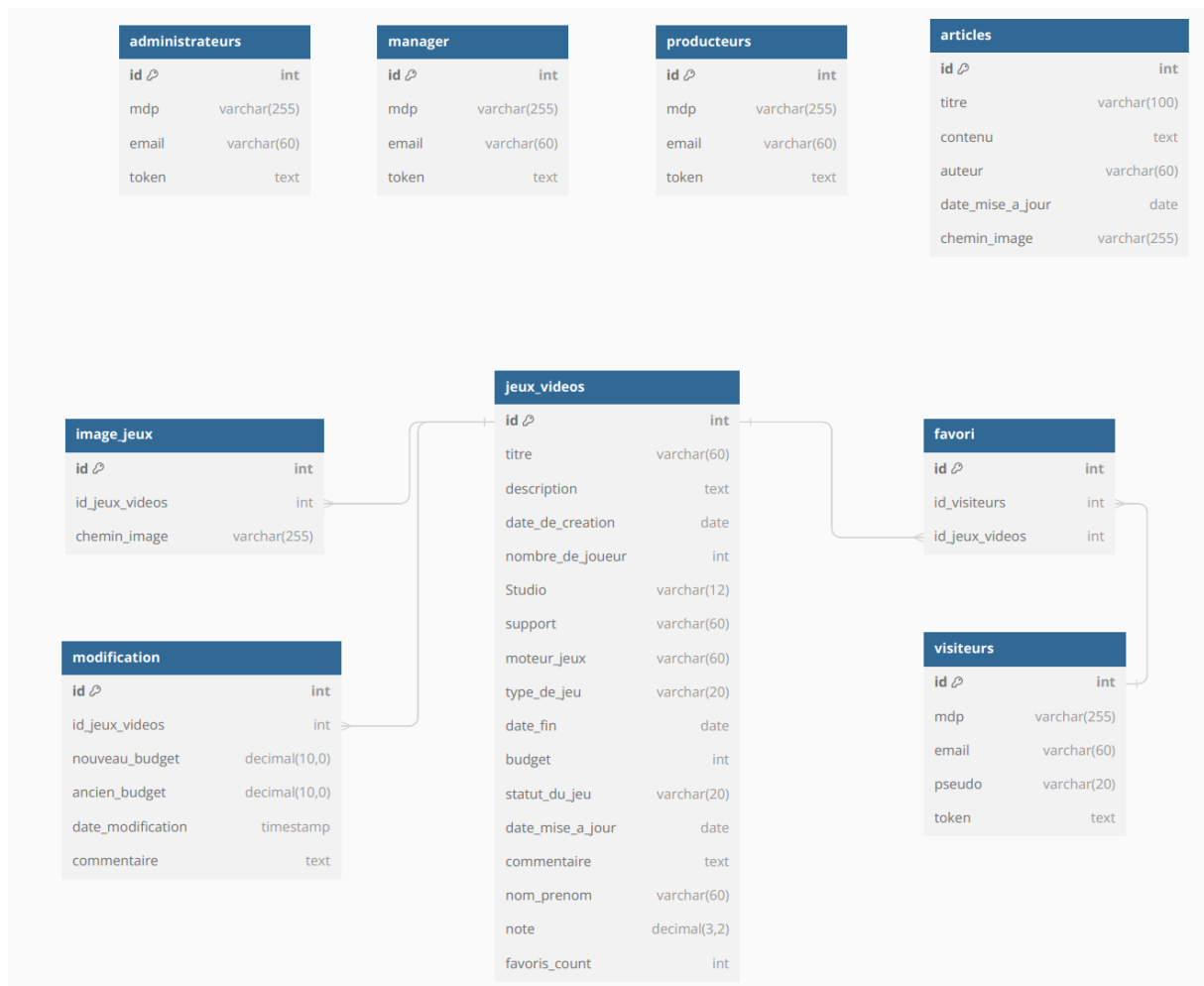
J'ai testé aussi l'écologie du site grâce au site suivant :

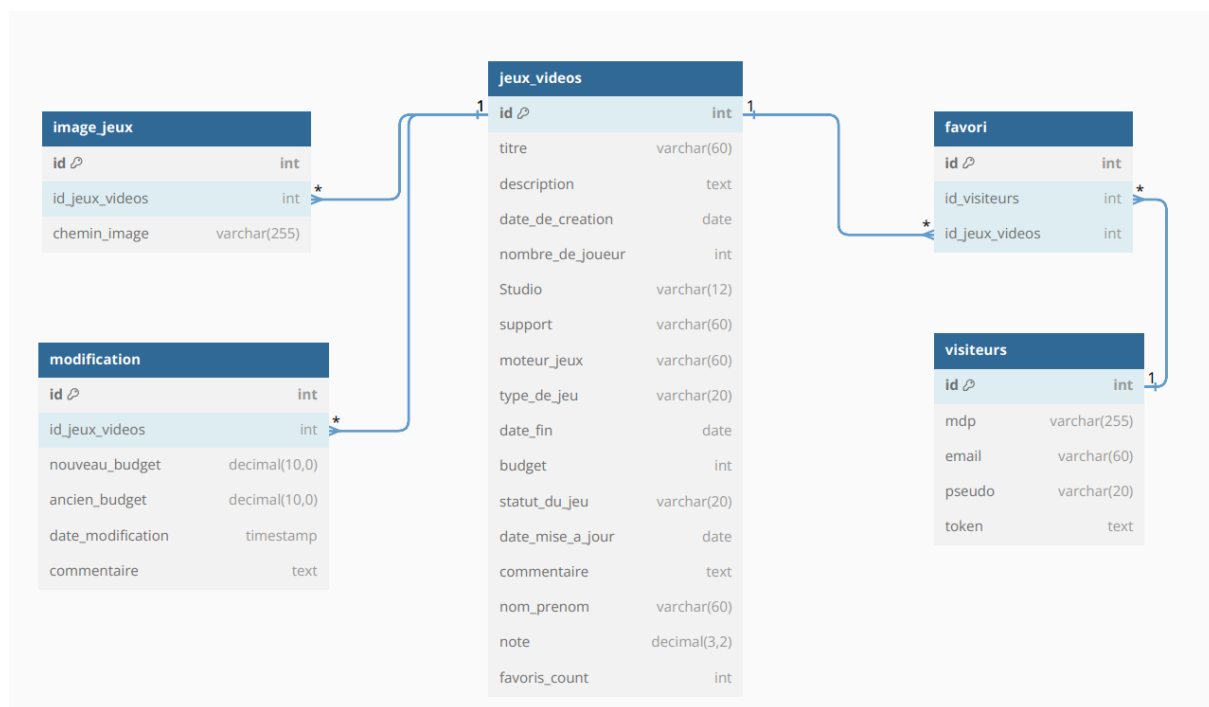
- <https://www.websitecarbon.com/>

Ainsi que sa vitesse grâce a ce site :

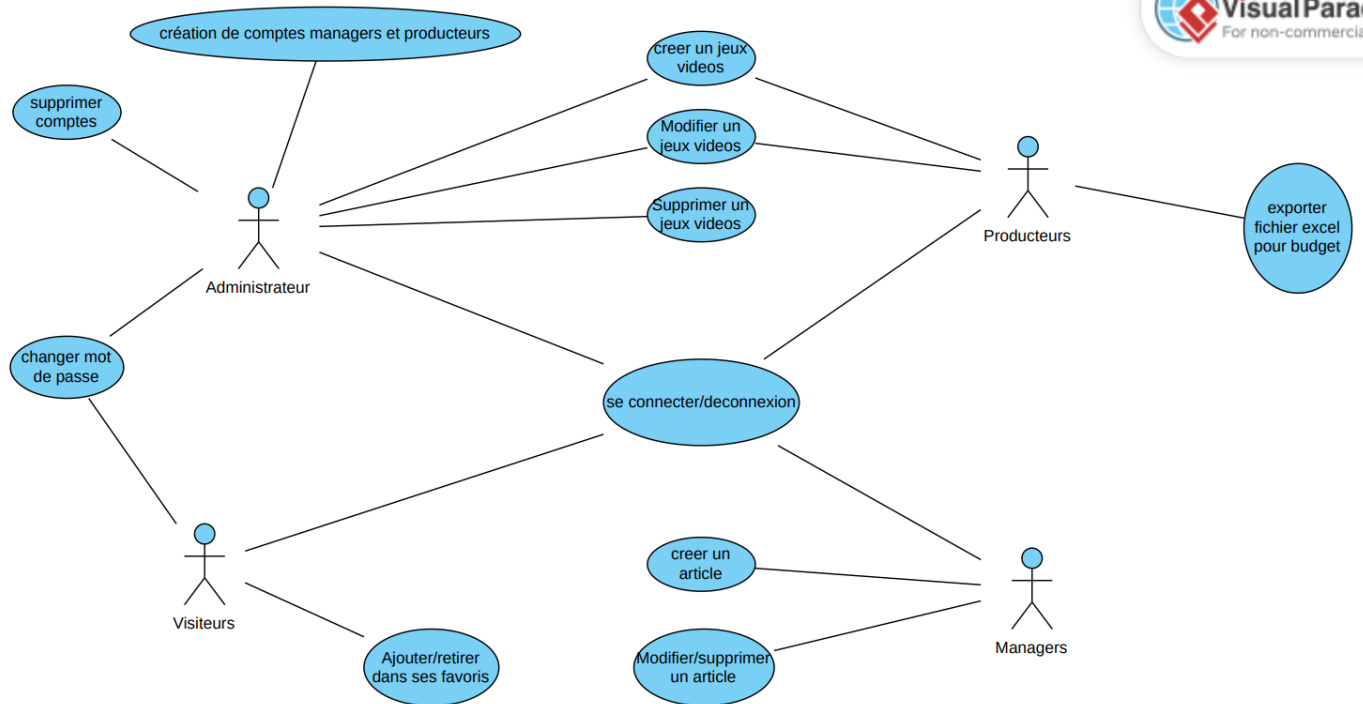
https://pagespeed.web.dev/?utm_source=psi&utm_medium=redirect&hl=fr

C. Diagramme de classe



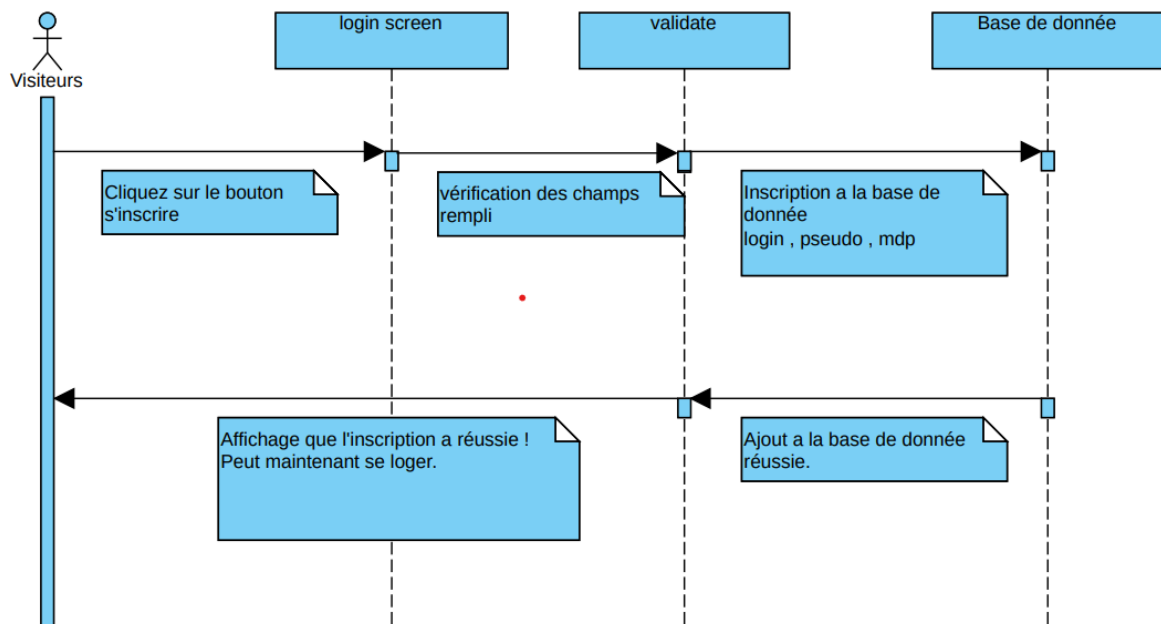


D. Diagramme d'utilisation



E. Diagramme de séquence

Pour une inscription :



Pour une vue d'un jeu en détails :

