Projet AP 3

Création d'une application de modération
User story : En tant qu'utilisateur, je souhaite, à partir du site restaurant avoir une interface de modération

Table des matières

Daily scrum debut (31/03/2025) :

GitLab : https://gitlab.com/Crouan/projet_java_moderation.git
GitLab API : https://gitlab.com/projet6274038/projet_java_moderationapi.git

Pendant la réunion :

➔ Attribution des tickets de l'IT4

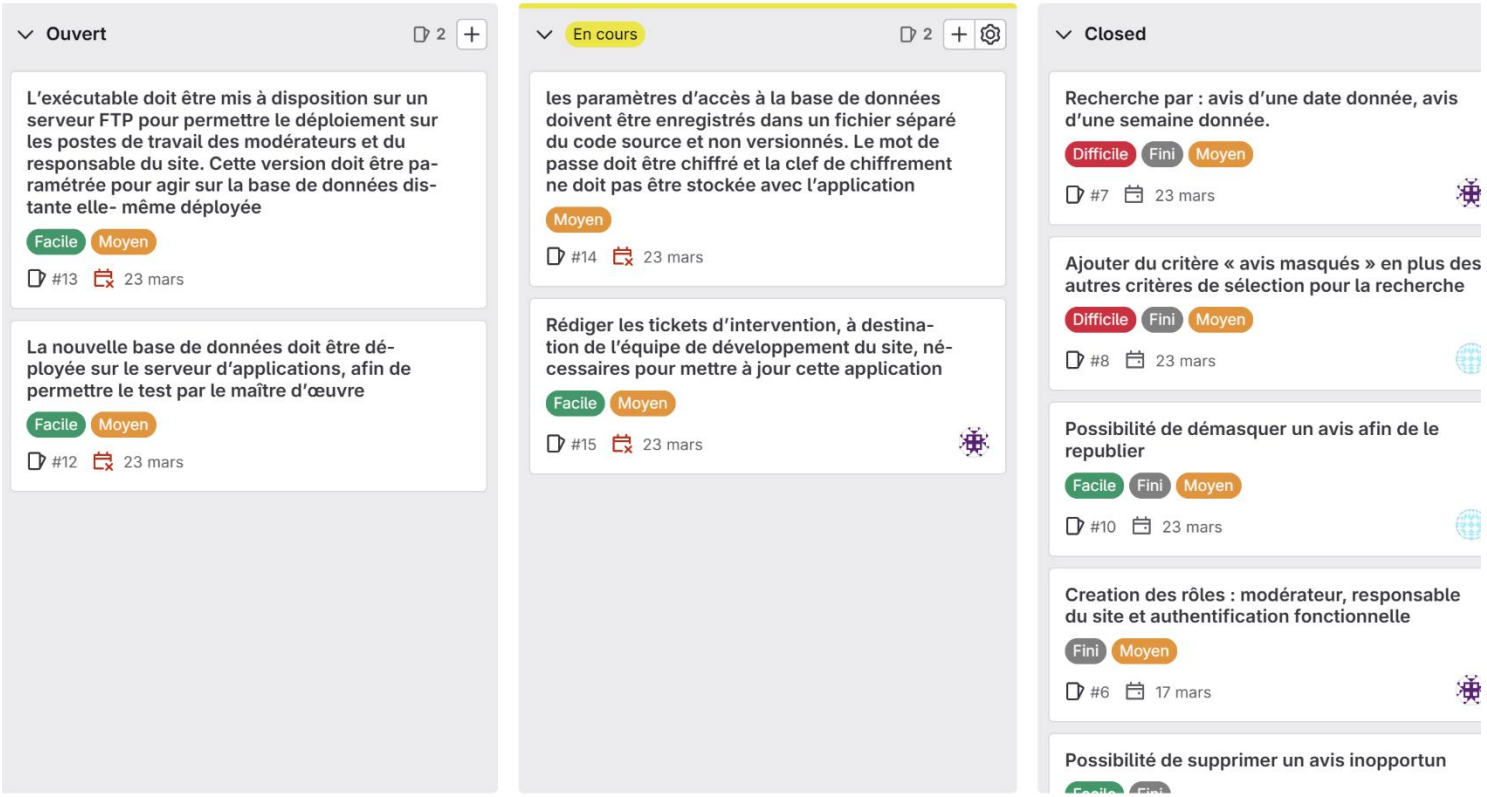
Planification de l'itération :

Dernière itération il nous manquant juste le chiffrement et le déploiement

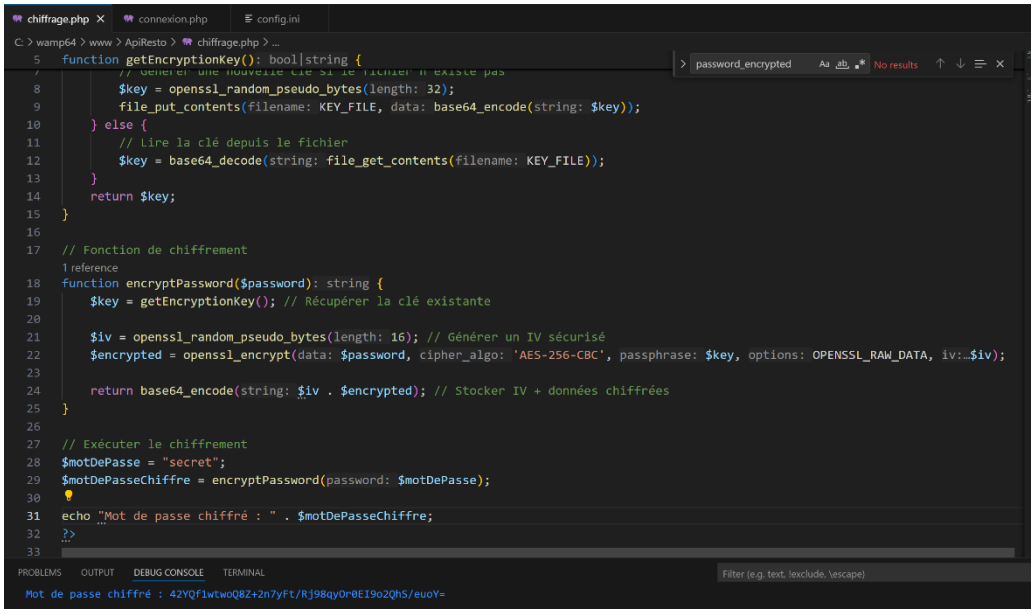
Répartition des tâches du début :

Pierre → Tâche 4

Tableau de Kanban du début :



Tache 4 : Chiffrement déchiffrement



J’ai commencé par créer un fichier chiffrage qui me permet de chiffrer mon mot de passe : secret dans un fichier encryption_KEY.php, comme on peut le voir sur le fichier.

Ensuite j’ai créé un fichier config.ini où je renseigne toutes les informations sur comment accéder à ma base de données ainsi qu’à la récupération du mot de passe chiffré et de l’utilisateur.

```php
<?php
// Essayer de récupérer la clé de chiffrement depuis une variable d'environnement
$key = getenv(name: 'ENCRYPTION_KEY');

// Si la clé n'est pas définie en tant que variable d'environnement, la charger depuis un fichier
if (!$key) {
    define(constant_name: 'KEY_FILE', value: 'encryption_key.txt'); // La clé de chiffrement est stockée séparément
    if (!file_exists(filename: KEY_FILE)) {
        die("Erreur : La clé de chiffrement est introuvable !");
    }
    $key = base64_decode(string: file_get_contents(filename: KEY_FILE));
} else {
    $key = base64_decode(string: $key);
}

// Charger la configuration
$config = parse_ini_file(filename: 'config.ini', process_sections: true);

$dbhost = $config['database']['host'];
$dbport = $config['database']['port'];
$dbname = $config['database']['dbname'];
$dbuser = $config['database']['user'];
$encryptedPassword = $config['database']['password_encrypted'];
```

Ensuite j'ai modifié le fichier connexion.php où j'ai récupéré ma variable d'environnement ENCRYPTION_KEY

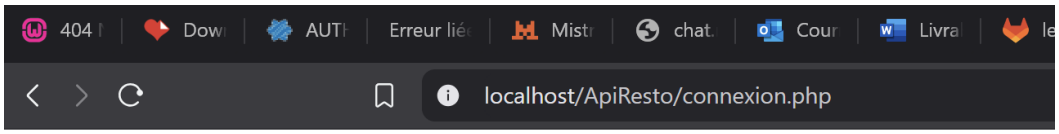
Et toutes mes informations de connexion à la BDD

```php
// Fonction de déchiffrement
1 reference
function decryptPassword($encryptedPassword, $key): bool|string {
    $data = base64_decode(string: $encryptedPassword);

    if (strlen(string: $data) < 16) {
        die("Erreur : Données chiffrées invalides !");
    }

    $iv = substr(string: $data, offset: 0, length: 16); // Extraire IV
    $encryptedText = substr(string: $data, offset: 16); // Extraire le texte chiffré

    return openssl_decrypt(data: $encryptedText, cipher_algo: 'AES-256-CBC', passphrase: $key, options: OPENSSL_RAW_DATA, iv:…$iv);
}

try {
    $dbpasswd = decryptPassword(encryptedPassword: $encryptedPassword, key: $key);

    if (!$dbpasswd) {
        throw new Exception(message: "Déchiffrement du mot de passe échoué !");
    }
```

Puis on déchiffre le mot de passe de l'utilisateur

```php
    // Connexion à MySQL
    $connexion = new PDO(dsn: "mysql:host=$dbhost;port=$dbport;dbname=$dbname", username: $dbuser, password: $dbpasswd);
    $connexion->setAttribute(attribute: PDO::ATTR_ERRMODE, value: PDO::ERRMODE_EXCEPTION);
    $connexion->exec(statement: "SET CHARACTER SET utf8");


} catch (PDOException $e) {
    die("Erreur de connexion : " . $e->getMessage());
} catch (Exception $e) {
    die("Erreur de sécurité : " . $e->getMessage());
}
?>
```

Ensuite, nous nous occupons de la connexion à MySQL



Connexion réussie à la base de données.

Et nous finissons par un test unitaire où l'on vérifie que la connexion à la base de données fonctionne bien

Daily scrum Fin (31/03/2025) :

Résultat :

Pierre s'est occupé du chiffrage et du déchiffrage du mot de passe de passe de l'utilisateur en créant une variable d'environnement. Il a rencontré quelques difficultés lors de celui-ci mais elles ont pu être réglées à l'aide de Mr Bourgeois.

Tableau de Kanban de fin :