Welcome Training
Extraits de code commentés

1. Création d'un utilisateur

1.1 Classe FetchAllClasses

```php
<?php
// Récupère toutes les classes de la base de données
class FetchAllClasses {
    private $pdo; // Instance PDO pour la connexion à la base de données

    public function __construct($pdo) {
        $this->pdo = $pdo; // Initialise l'instance PDO
    }

    // Récupère toutes les classes de la base de données
    public function fetchAllClasses() {
        // Prépare et exécute la requête SQL pour récupérer toutes les classes
        $sql = "SELECT * FROM `class`";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();

        return $stmt->fetchAll(PDO::FETCH_ASSOC); // Retourne les résultats sous forme de tableau associatif
    }

    // Récupère les ID et noms de toutes les classes
    public function fetchAllClassesID() {
        // Prépare et exécute la requête SQL pour récupérer les ID et noms des classes
        $sql = "SELECT id, name FROM class";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        return $stmt->fetchAll(PDO::FETCH_ASSOC); // Retourne les résultats sous forme de tableau associatif
    }
```

1.2 Classe UserForm

```php
<?php
// Gère le formulaire d'ajout d'un utilisateur
class UserForm {
    // Déclarattion des propriétés
    public $first_name;
    public $surname;
    public $email;
    public $password;
    public $role;
    public $class_id;

    public function __construct() {
        $this->fetchForm(); // Récupère les données du formulaire
    }

    // Associe les données du formulaire aux propriétés de la classe
    private function fetchForm() {
        $this->first_name = $_POST['first_name'];
        $this->surname = $_POST['surname'];
        $this->email = $_POST['email'];
        $this->password = $_POST['password'];
        $this->role = $_POST['role'];
        // Si la classe n'est pas définie, on la met à null (cas de l'admin et de l'enseignant))
        $this->class_id = !empty($_POST['class_id']) ? $_POST['class_id'] : null;
    }
```

1.3 Classe CreateUser

```php
// Gère la création d'un utilisateur
class CreateUser {
    private $pdo; // Instance de PDO pour la connexion à la base de données
    private $fetchClass; // Instance de FetchAllClasses pour récupérer les classes

    public function __construct($pdo) {
        $this->pdo = $pdo;
        $this->fetchClass = new FetchAllClasses($pdo); // Instance de FetchAllClasses
    }

    // Hash le mot de passe
    private function pwdHash($pwd) {
        $options = [
            'cost' => 12,
        ];
        return password_hash($pwd, PASSWORD_BCRYPT, $options);
    }

    // Récupère toutes les classes
    public function fetchClasses() {
        return $this->fetchClass->fetchAllClasses();
    }

    // Créé un nouvel utilisateur
    public function createUser() {
        $form = new UserForm; // Instance de UserForm, qui contient les données du formulaire

        $hashedPassword = $this->pwdHash($form->password); // Hash le mot de passe

        // Requête SQL pour insérer un nouvel utilisateur
        $sql = "INSERT INTO user (first_name, surname, email, password, class_id, role)
                VALUES (:first_name, :surname, :email, :password, :class, :role)";

        $stmt = $this->pdo->prepare($sql); // Prépare la requête SQL
        // Lier les paramètres à la requête
        $stmt->bindParam(':first_name', $form->first_name, PDO::PARAM_STR);
        $stmt->bindParam(':surname', $form->surname, PDO::PARAM_STR);
        $stmt->bindParam(':email', $form->email, PDO::PARAM_STR);
        $stmt->bindParam(':password', $hashedPassword, PDO::PARAM_STR);
        $stmt->bindParam(':class', $form->class_id, PDO::PARAM_INT);
        $stmt->bindParam(':role', $form->role, PDO::PARAM_STR);

        return $stmt->execute(); // Exécute la requête
    }
}
```

1.4 Traitement du formulaire de création d'utilisateur

```php
// Instance de CreateUser qui permet de créer un utilisateur depuis les données du formulaire
$user_form = new CreateUser($pdo);

// Récupération des classes pour le select
$classes = $user_form->fetchClasses();

// Si le formulaire est soumis, la méthode createUser est appelée pour créer l'utilisateur
if (count($_POST) > 0) {
    $user_form->createUser();
}
```

1.4 Formulaire de création d'utilisateur

```
<form action="create-user.php" method="POST">

    <!-- Prénom -->
    <div class="mb-3">...
    </div>

    <!-- Nom -->
    <div class="mb-3">...
    </div>

    <!-- Email -->
    <div class="mb-3">...
    </div>

    <!-- Mot de passe -->
    <div class="mb-3">...
    </div>

    <!-- Rôle -->
    <div class="mb-3">...
    </div>

    <!-- Classe -->
    <div class="mb-4">
        <label for="classe" class="form-label">Classe :</label>
        <select class="form-control custom-select form-select mb-4" id="class_id" name="class_id">
            <option value=""></option>
            <!-- Boucle pour afficher les classes dans le select, grâce à l'instanciation préalable de $classes' -->
            <?php foreach($classes as $class): ?>
                <option value=<?php echo($class["id"]); ?>>
                    <?php  echo($class["name"]); ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>

    <!-- Boutton -->
    <div class="d-flex justify-content-center">...
    </div>
</form>
```

2. Création d'un cours

2.1 Classe FetchAllSubjects

```php
<?php
// Récupère toutes les matières de la base de données
class FetchAllSubjects {
    private $pdo; // Instance PDO pour la connexion à la base de données

    public function __construct($pdo) {
        $this->pdo = $pdo; // Initialise la connexion PDO
    }

    //$ Récupère toutes les matières
    public function fetchAllSubjects() {
        // Requête SQL pour récupérer toutes les matières
        $sql = "SELECT * FROM `subject`";
        $stmt = $this->pdo->prepare($sql); // Prépare la requête
        $stmt->execute(); // Exécute la requête

        // Retourne les résultats sous forme de tableau associatif
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }

    // Récupère les ID et noms de toutes les matières
    public function fetchAllSubjectsID() {
        // Requête SQL pour récupérer les ID et noms de toutes les matières
        $sql = "SELECT id FROM subject";
        $stmt = $this->pdo->prepare($sql); // Prépare la requête
        $stmt->execute(); // Exécute la requête

        // Retourne les résultats sous forme de tableau associatif
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
?>
```

2.2 Classe FetchAllTeachers

```php
<?php
// Récupère tous les enseignants de la base de données
class FetchAllTeachers {
    private $pdo; // Instance PDO pour la connexion à la base de données

    public function __construct($pdo) {
        $this->pdo = $pdo; // Initialise la connexion PDO
    }

    //  Récupère tous les enseignants
    public function fetchAllTeachers() {
        // Requête SQL pour récupérer tous les utilisateurs ayant le rôle "enseignant"
        $sql = "SELECT id, first_name, surname, email FROM user WHERE role = 'enseignant'";
        $stmt = $this->pdo->prepare($sql); // Prépare la requête
        $stmt->execute(); // Exécute la requête

        // Retourne les résultats sous forme de tableau associatif
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

2.3 Classe ScheduleForm

```php
<?php
// Gère le formulaire d'ajout d'un cours
class ScheduleForm {
    // Déclaration des propriétés
    public $subject_id;
    public $class_id;
    public $teacher_id;
    public $start_datetime_str;
    public $end_datetime_str;

    private $date;
    private $time;
    private $length;

    public function __construct() {
        $this->fetchForm(); // Récupère les données du formulaire
        $this->setLength(); // Définit la durée
        $this->dateHandler(); // Gère la date et l'heure
    }

    // Associe les données du formulaire aux propriétés de la classe
    private function fetchForm() {
        $this->subject_id = $_POST['subject_id'];
        $this->class_id = $_POST['class_id'];
        // Vérifie si l'enseignant est sélectionné, sinon le définit sur NULL
        $this->teacher_id = empty($_POST['enseignant']) ? NULL : $_POST['enseignant'];
        $this->date = $_POST['date'];
        $this->time = $_POST['time'];
    }

    // Vérifie si la durée est définie, sinon la définit sur 0
    private function setLength() {
        $this->length = isset($_POST['duree']) ? (int)$_POST['duree'] : 0;
    }

    // Gère la date et l'heure de fin
    private function dateHandler() {
        // Associe la date et l'heure à un objet DateTime
        $start_datetime = new DateTime("$this->date $this->time");

        // Initialise l'heure de fin sur l'heure de début
        $end_datetime = clone $start_datetime;
        // Ajoute la durée du cours à l'heure de fin
        $end_datetime->modify("+$this->length hours");

        // Convertit les date et heure de DateTime en chaînes de caractères
        // pour les stocker dans la base de données
        $this->start_datetime_str = $start_datetime->format('Y-m-d H:i:s');
        $this->end_datetime_str = $end_datetime->format('Y-m-d H:i:s');
    }
```

2.4 CreateSchedule : classe qui créé un cours en base de données

```php
// Gère la création d'un cours'
class CreateSchedule {

    private $pdo; // Instance de PDO pour la connexion à la base de données
    private $fetchSubjects; // Instance de FetchAllSubjects pour récupérer les matières
    private $fetchTeachers; // Instance de FetchAllTeachers pour récupérer les enseignants
    private $fetchClasses; // Instance de FetchAllClasses pour récupérer les classes

    public function __construct($pdo) {
        $this->pdo = $pdo;
        $this->fetchSubjects = new FetchAllSubjects($pdo); // Instanciation de FetchAllSubjects
        $this->fetchTeachers = new FetchAllTeachers($pdo); // Instanciation de FetchAllTeachers
        $this->fetchClasses = new FetchAllClasses($pdo); // Instanciation de FetchAllClasses

    }

    public function fetchSubjects() {
        return $this->fetchSubjects->fetchAllSubjects(); // Récupère toutes les matières
    }

    public function fetchTeachers() {
        return $this->fetchTeachers->fetchAllTeachers(); // Récupère tous les enseignants
    }

    public function fetchClasses() {
        return $this->fetchClasses->fetchAllClasses(); // Récupère toutes les classes
    }

    public function createSchedule() {
        $form = new ScheduleForm(); // Instanciation de la classe ScheduleForm qui contient les données du formulaire

        // Requête préparée pour insérer un nouvel emploi du temps
        $sql_schedule = "INSERT INTO schedule (subject_id, class_id, teacher_id, start_datetime, end_datetime)
                        VALUES (:subject_id, :class_id, :teacher_id, :start_datetime, :end_datetime)";

        $stmt_schedule = $this->pdo->prepare($sql_schedule); // Préparation de la requête

        // Liaison des paramètres à la requête
        $stmt_schedule->bindParam(':subject_id', $form->subject_id, PDO::PARAM_INT);
        $stmt_schedule->bindParam(':class_id', $form->class_id, PDO::PARAM_INT);
        $stmt_schedule->bindParam(':teacher_id', $form->teacher_id, PDO::PARAM_INT);
        $stmt_schedule->bindParam(':start_datetime', $form->start_datetime_str, PDO::PARAM_STR);
        $stmt_schedule->bindParam(':end_datetime', $form->end_datetime_str, PDO::PARAM_STR);

        return $stmt_schedule->execute(); // Exécution de la requête
    }
}
```

2.5 Traitement du formulaire de création de cours

```php
// Instance de CreateSchedule qui permet de créer un cours depuis les données du formulaire
$schedule_form = new CreateSchedule($pdo);

$subjects = $schedule_form->fetchSubjects(); // Récupère les matières
$teachers = $schedule_form->fetchTeachers(); // Récupère les enseignants
$classes = $schedule_form->fetchClasses(); // Récupère les classes

// Si le formulaire est soumis, la méthode createSchdule est apprlée pour créer le cours
if(count($_POST) > 0) {
    $schedule_form->createSchedule();
}
```

2.6 Formulaire de création de cours

```php
<form action="create-schedule.php" method="POST">

    <!-- Matière -->
    <div class="mb-3">
        <label for="matiere" class="form-label">Matière :</label>
        <select class="form-control custom-select form-select mb-4" id="subject_id" name="subject_id" required>
            <option value=""></option>
            <!-- Boucle pour afficher les matières dans le select, grâce à l'instanciation préalable de $subjects' -->
            <?php foreach($subjects as $s): ?>
                <option value=<?php echo($s["id"]); ?>>
                    <?php echo($s["name"]); ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>

    <!-- Date -->
    <div class="mb-3">...
    </div>

    <!-- Horaire -->
    <div class="mb-3">...
    </div>

    <!-- Durée -->
    <div class="mb-3">...
    </div>

    <!-- Enseignant -->
    <div class="mb-3">
        <label for="enseignant" class="form-label">Enseignant :</label>
        <select class="form-control custom-select form-select" id="enseignant" name="enseignant">
            <option value=""></option>
            <!-- Boucle pour afficher les enseignants dans le select, grâce à l'instanciation préalable de $teachers' -->
            <?php foreach($teachers as $t): ?>
                <option value = <?php echo($t["id"]); ?>>
                    <?php echo($t["first_name"] . " ". $t["surname"]); ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>

    <!-- Classe -->
    <div class="mb-4">
        <label for="classe" class="form-label">Classe :</label>
        <select class="form-control custom-select form-select mb-4" id="class_id" name="class_id" required>
            <option value=""></option>
            <!-- Boucle pour afficher lesclasses dans le select, grâce à l'instanciation préalable de $classes' -->
            <?php foreach($classes as $class): ?>
                <option value = <?php echo($class["id"]); ?>>
                        <?php  echo($class["name"]); ?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>

    <!-- Bouton -->
    <div class="d-flex justify-content-center">...
    </div>

</form>
```