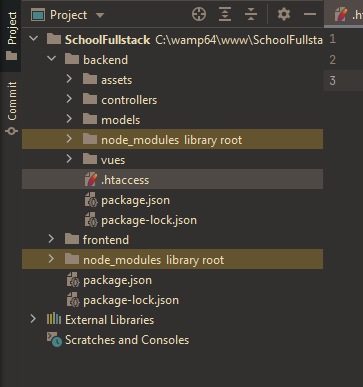
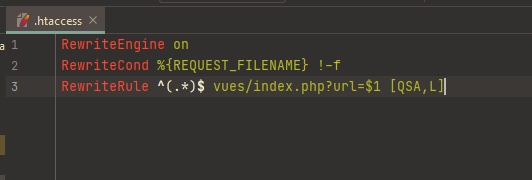
# **PROJET 13 PHP CRUD Backend + React Frontend**

## PARTIE A : LA CONNEXION

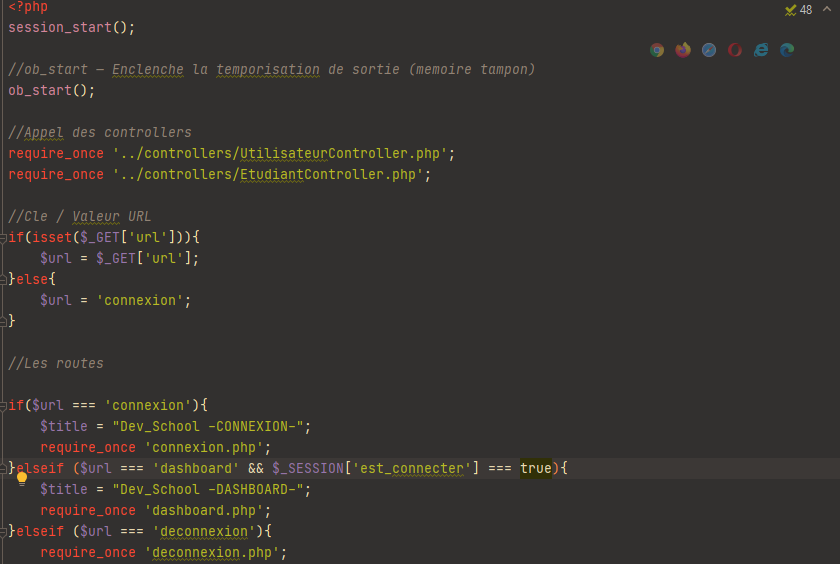
1. Créer un projet PHP (démarrer wamp ou xamp (PHP + apache + MySQL)
2. Créer la structure de votre projet avec le patron de conception (Design Pattern) MVC (Model Vue Controller)



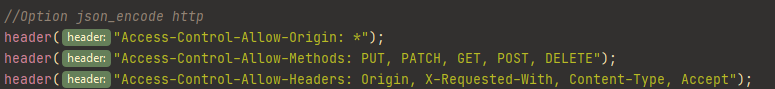
1. Créer un router et votre point entrée apache (.htaccess). Exemple : ici apache pointe vers le dossier vues + index.php ?=quelquechose



1. Votre router doit pointer vers un fichier de connexion, à l’aide de session\_start() et $\_SESSION interdisez l’accès au tableau de bord si le visiteur n’est pas connecté.



1. Créer un fichier template.php qui contient gabarit html utilisable sut toutes les pages à l’aide de la mémoire tampon (ob\_start() et ob\_get\_clean())
2. Créer une base de donnée et une table utilisateurs (insérer un ou plusieurs administrateur)
3. Créer une classe de connexion à MySQL avec l’objet PDO (privilégié le design pattern Singleton pour éviter les multiples instances de connexion)
4. Ajouter les options de vos futures requêtes http :



9) Créer une classe (modèle) de connexion des utilisateurs qui hérite de la classe mère PDO de connexion à MySQL.

10) Créer une fonction publique de connexion à l’aide email et mot de passe, dans une condition lorsque la connexion matches, créer vos variables de session avec $\_SESSION[‘estConnecter’] = true et une redirection vers votre tableau de bord (sinon afficher les erreurs)

11) Créer un fichier connexion\_controller.php qui appelle son modèle et une fonction de connexion qui stock l’instance du modèle et appel la méthode de connexion, la fonction retourne une variable.

12) Votre router (index.php) appel la vue connexion.php, cette dernière affiche un formulaire de connexion avec la méthode POST. Le bouton de soumission appel la fonction connexion de votre Controller. Les erreurs sont donc gérées depuis le model et affichée dans votre vue en cas de mauvais email et mot de passe.

13) Le fichier dashboard.php (tableau bord) est inaccessible si le visiteur n’est pas connecté :

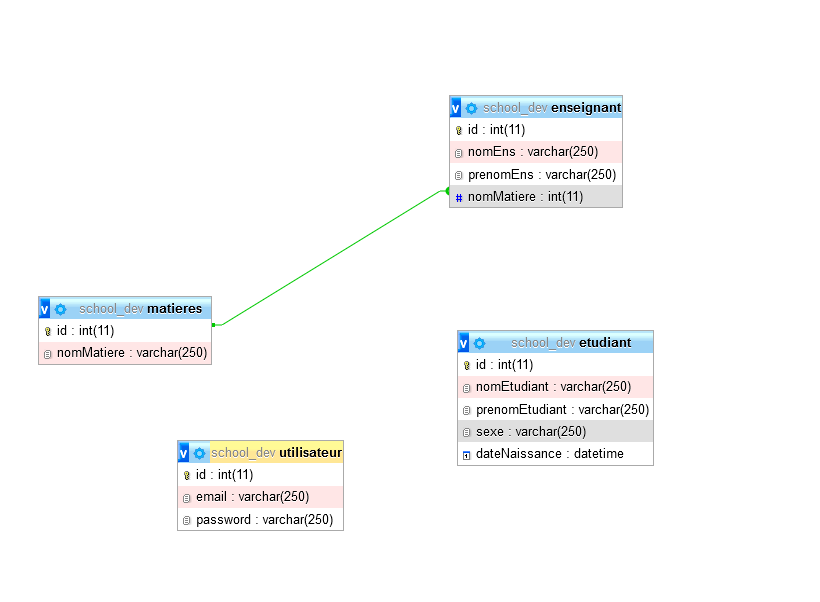


14) Dans votre router (index.php) si aucune route ne matche : exécuter une redirection :



## PARTIE B : LE CRUD ECOLE

1. Exemple de base du MCD (Modèle Conceptuel de Données)
2. UTILISATEUR (id (int primary key, email (varchar 250), password(varcher 250))
3. ETUDIANT (id (int primary key), nomEtudiant (varchar 250), prenomEtudiant (varchar 250), sexe (varchar 250), dateNaissance (DATETIME))
4. ENSEIGNANT (id (int primaty key), nomEns (varchar 250) prenomEns(varchar 250), #nomMatiere (int foreign key INDEX))
5. MATIERE (id (int primary key), nomMatiere (varchar 250))



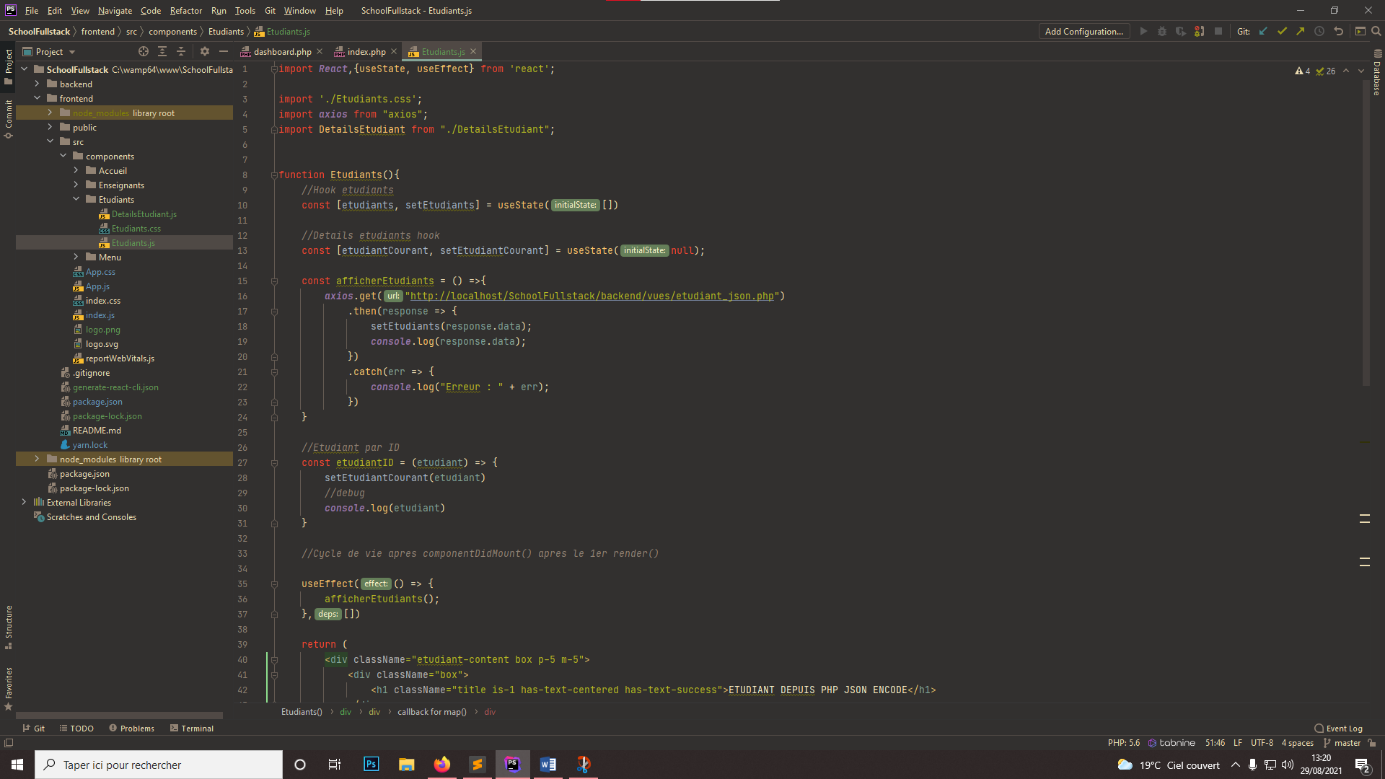
1. Dans un premier temps effectuer les opérations de CRUD dans votre etudiant\_modele.php (les fonctions de votre Controller seront appelées dans votre router) de la table étudiant qui n’a aucune dépendance fonctionnelle (pas de clé étrangère)
2. Transformer vos données SQL au format json à l’aide de la fonction PHP json\_encode, les données seront affichées dans un fichier PHP => etudiant\_json.php



1. Créer un dossier Frontend et générer une CRA (create react app)



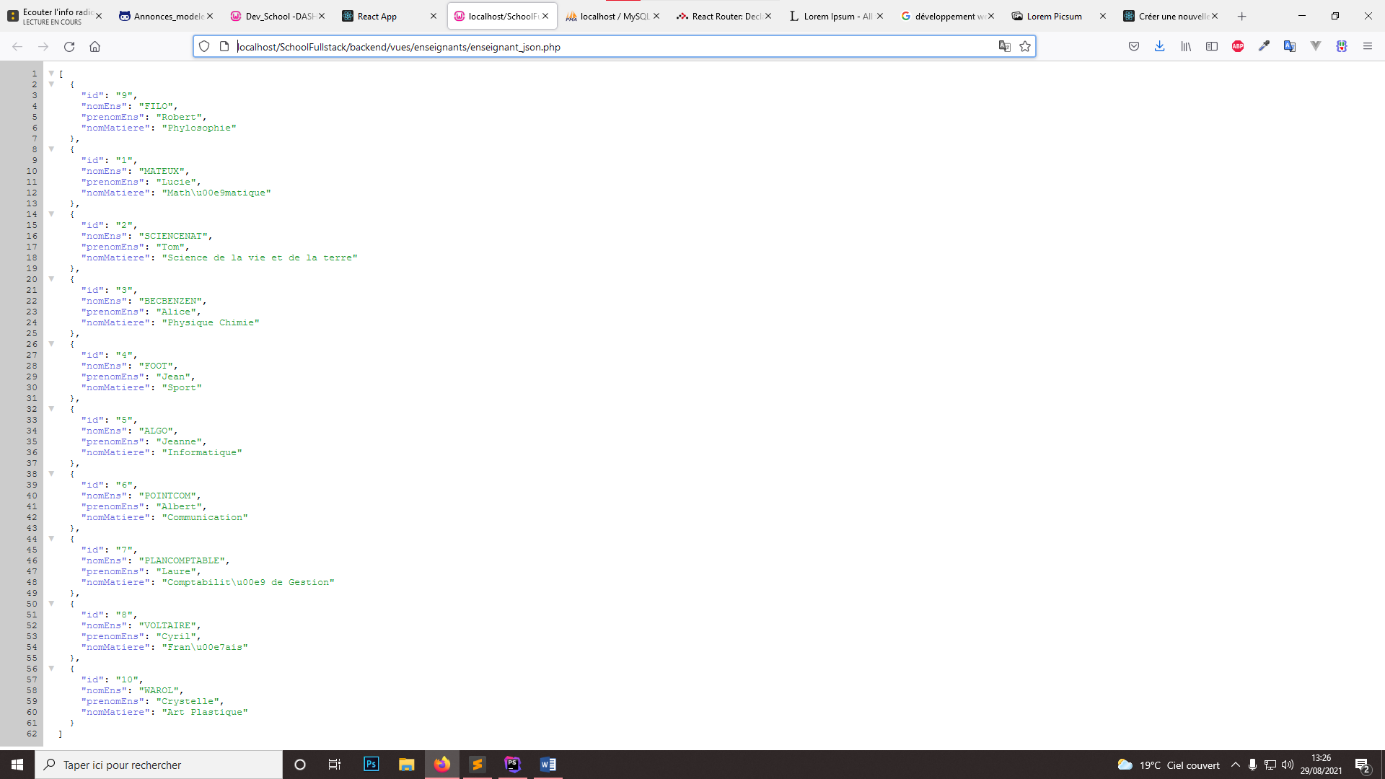
1. Créer un composant étudiant et à l’aide des hooks et axios (Requête http) => afficher tous les étudiants de votre backend :



6)Afficher les détails de chaque étudiant à l’aide de la programmation modulaire et les passages des propriétés (props) React (composant réutilisable) dans un composant <DetailsEtudiant props={valeur ID}/>

## PARTIE C : Les clés étrangères et json ?

1. Créer 2 tables (enseignant et matière) et une dépendance fonctionnelle entre les enseignants et le nom de la matière enseignée.
2. Effectuer les opérations de CRUD pour ces 2 tables.
3. Exporter les données SQL des enseignants vers une URL au format json à l’aide de PHP json\_encode



1. Répéter les opérations d’affichage dans la partie frontend avec react et url des enseignants.
2. Ajouter des filtres de tri dans React (ex : barre de recherche, check box, liste de choix, etc…) à l’aide des fonctions filter() et includes() JavaScript et des conditions à l’aide des ternaires :
3. RAPPEL {condition1 ? ‘’OUI ‘’ : ‘’NON ‘’}
4. OU {condition1 ? (Block JSX) : (Sinon autre block JSX)}

# BRAVO VOUS ETES UN DEVELOPPEUR FULLSTACK PHP + JAVASCRIPT