

# Cahier des charges: Gestion d'une école privée

Développer une application en se basant sur HTML, CSS, Bootstrap et JavaScript, TypeScript, Angular, Node, ExpressJS, MongoDB, API ...

Chercher et intégrer un template sous le thème de Education, School, Management ....

L'application présente 3 acteurs:

- Admin: c'est l'acteur qui peut faire le tout: consulter et supprimer tous les utilisateurs (students et teachers), tous les cours ...  
L'admin peut affecter des **students** à un **teacher** dans un **cours** bien défini.  
Il peut aussi valider ou non les Teachers qui ont fait des comptes.
- Teacher : Le teacher peut ajouter, modifier ou supprimer un cours. Il peut aussi consulter ses cours.  
Il consulte en plus le cours avec les students affectés par l'admin.  
Le teacher peut affecter une note et une évaluation à un student pour un cours bien défini.
- Student : Le student peut consulter les cours affectés par l'admin ainsi que la note donnée par le teacher. Il peut aussi voir l'évaluation de son teacher.
- Parent: Le parent peut consulter uniquement les notes et les évaluations sur les cours de son child.

Les fonctionnalités de base:

- Une page d'accueil qui présente quelques cours, quelques teachers, un header, un footer avec des informations ...
- Une page de connexion : Tel et mot de passe ( la connexion ne se fait qu'après le remplissage des deux inputs, si non, afficher des messages d'erreur)
- Une page d'inscription d'un teacher: FirstName, LastName, email, téléphone, adresse, mot de passe, spécialité, CV format pdf ( développer les contrôles de saisie nécessaires)
- Une page d'inscription d'un student: FirstName, LastName, email, téléphone, adresse, mot de passe, photo de profil format image ( développer les contrôles de saisie nécessaires )
- Une page d'inscription d'un parent: FirstName, LastName, email, téléphone, adresse, mot de passe, téléphone de son enfant (l'inscription ne se fait qu'après vérification que le numéro de son fils existe dans la liste des students) ( développer les contrôles de saisie nécessaires )
- Après la connexion d'un teacher (ce qui n'est possible qu'après sa validation par l'admin), il se trouve dans une page qui présente son dashboard où il peut ajouter, consulter, supprimer ou modifier un cours. Il peut consulter sous chaque cours la liste des students affectés par l'admin. Le teacher peut donner une note et une évaluation pour chaque student.
- Une page d'ajout d'un cours ( nom, description, durée ). N'oubliez pas d'affecter ce cours à un teacher.
- Une page de recherche des teachers par spécialité (Utiliser un TDF avec un champs obligatoire)
- Après la connexion d'un student, il se trouve dans son dashboard . Il peut voir la liste des cours affectés par l'admin. A chaque selection d'un cours, il peut consulter la note et l'évaluation de son teacher.

- Après la connexion d'un parent, il se trouve dans la page home où il peut voir la liste des cours et des teachers dans la plateforme. Le parent peut insérer le numéro de téléphone de son fils afin de chercher les cours dont il est inscrit ainsi que la note et l'évaluation du teacher.

NB:

- Utiliser @Input, les directives structurelles ainsi que les directives d'attributs.