

	<p>Année Universitaire : 2025-2026</p> <p><b>Applications Web Avancées</b></p>
<p><b>Atelier n°2 :</b></p> <p><b>Vue d'ensemble</b></p>	

## Objectifs

L'objectif de cet atelier est de permettre aux étudiants de mobiliser les notions suivantes du Framework Laravel : routes, contrôleurs, middleware, vues Blade, fournisseurs de services et façades. Il s'agit de construire une mini-application simulant un conseiller nommé Advisor.

### I- Prérequis

- Projet Laravel 12

### II- Travail demandé

#### Exercice 1 :

L'utilisateur va accéder à une page : `/advisor?age=22&interet=developpement`. L'application filtre les utilisateurs mineurs, et selon l'âge et l'intérêt renseigné, affiche un message personnalisé de conseil.

1. Créer une route **/advisor** dans le fichier qui accepte deux paramètres dans l'URL : age (obligatoire) et intérêt (optionnel).
2. Générer le contrôleur **AdvisorController** et ajoutez une méthode **show()** qui récupère les paramètres de la requête et les envoie à la vue **show.blade.php**
3. Créer un middleware **CheckAge** qui vérifie si l'utilisateur a plus de 18 ans. Sinon, redirigez-le vers une route **/acces-refuse**.
4. Enregistrer le middleware et appliquer-le uniquement à la route **/advisor**.
5. Créer le code qui permet d'afficher le message personnalisé dans **show.blade.php**
6. Créer une route **/acces-refuse** qui renvoie une vue simple informant l'utilisateur qu'il est trop jeune pour accéder au service.

## Exercice 2 :

1. Créer un contrôleur **ArticleController** qui envoie une liste d'articles (tableau associatif) à la vue principale.
  2. Créer un composant Blade nommé **ArticleCard** qui reçoit : un titre, un auteur, et un slot.
  3. Mettre en forme le composant ArticleCard CSS).
  4. Créer un autre composant Blade **Alert** qui reçoit un type (success, error, info) et un message (slot). Appliquer un style pour chaque type de message
  5. Afficher tous les articles dans la page `articles.blade.php` en utilisant **ArticleCard**
  6. En haut de la page, afficher les différents types de message avec le composant **Alert**.
- NB : n'oubliez pas de lancer la commande `npm run dev` pour compiler et charger les fichiers css .