



# مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Direction Régionale du Marrakech - Safi

EVALUATION DE FIN DE MODULE			
Établissement :		Durée : 02H00	
Filière :	Développement Digital Option Web Full Stack	Année : 2022/2023	1A <input type="checkbox"/> 2A <input checked="" type="checkbox"/>
Groupe :		Date :	
Module :	Développement Back-end		
Epreuve/Barème :	40 /40	VARIANTE : V1	

## Partie théorique : 10 points

Reproduire le tableau suivant sur votre feuille de rédaction et utiliser-le pour répondre au QCM :

Question 1	Question 2	Question 3	Question 4	Question 5

N.B: Une question peut avoir plusieurs réponses justes.

### Question 1 : Quel est l'objectif principal de la migration sous Laravel

	A. Importer des données dans une base de données.		B. Exporter des données dans une base de données.
X	C. Créer et modifier des tables dans la base de données.		D. Effectuer des requêtes SQL complexes sur une base de données.

### Question 2 : Où est ce qu'on ajoute un middleware global pour informer une application Laravel de son existence :

	A. Dans l'objet \$routeMiddleware du fichier App/HTTP/kernel.php		B. Dans l'objet \$middleware du fichier App/env
X	C. Dans l'objet \$middleware du fichier App/HTTP/kernel.php		D. Dans l'objet \$routeMiddleware du fichier App/env

### Question 3 : Quelle est la commande permettant de migrer les derniers fichiers créés pour créer ou modifier des tables de bases de données sous Laravel :

	A. php artisan make :migration	X	B. php artisan migrate
	C. php artisan migrate :refresh		D. php artisan migrate :rollback

### Question 4 : Quelle est la définition de route à utiliser avec un contrôleur à action unique

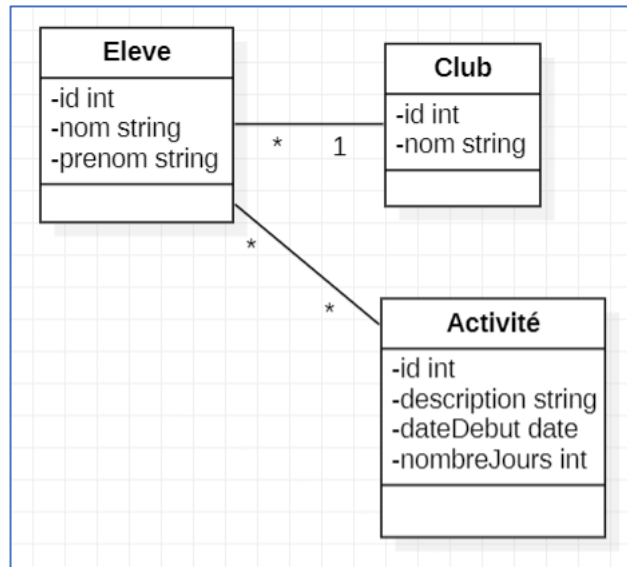
	A. Route ::resource('/chemin', NomContrôleur ::class) ;	X	B. Route ::get('/chemin', NomContrôleur ::class) ;
	C. Route ::invoke('/chemin', NomContrôleur ::class) ;		D. Route ::middleware('/chemin', NomContrôleur ::class) ;

### Question 5 : Dans une relation de type un à plusieurs (One to Many), quelle est la méthode employée pour récupérer les enregistrements du model enfant :

X	A. hasMany()		B. hasOne()
	C. belongsTo()		D. hasOneThrough()

## Partie pratique : 30 points

L'école AL-ANOUAR désire informatiser ses clubs d'activités parascolaires, elle vous a chargé de développer une application Web en utilisant le framework php Laravel et le système de gestion de base de données MySQL. Le diagramme de classe sur lequel vous vous êtes basé est le suivant :



Le tableau suivant illustre les URLs de l'application ainsi que les routes associées :

Méthode HTTP	URL	Action	Nom de la route
GET	/eleves	index	eleves.index
GET	/eleves/{id}	show	eleves.show
GET	/eleves/create	create	eleves.create
POST	/eleves/store	store	eleves.store
GET	/eleves/{id}/edit	edit	eleves.edit
PUT/PATCH	/eleves/{id}	update	eleves.update
DELETE	/eleves/{id}	destroy	eleves.destroy
GET	/clubs	index	clubs.index
GET	/clubs/{id}	show	clubs.show
GET	/clubs/create	create	clubs.create
POST	/clubs/store	store	clubs.store
GET	/clubs/{id}/edit	edit	clubs.edit
PUT/PATCH	/clubs/{id}	update	clubs.update
DELETE	/clubs/{id}	destroy	clubs.destroy
GET	/activités	index	activités.index
GET	/activités/{id}	show	activités.show
GET	/activités/create	create	activités.create
POST	/activités/store	store	activités.store
GET	/activités/{id}/edit	edit	activités.edit
PUT/PATCH	/activités/{id}	update	activités.update
DELETE	/activités/{id}	destroy	activités.destroy

On dispose des vues suivantes :

La vue **eleves.index** :

Gestion des élèves					
Liste des élèves					
ID élève	Nom	Prénom	ID club		
1	BADAoui	Imade	3	Modifier	Supprimer >>
2	MELLALI	Mourad	2	Modifier	Supprimer >>
3	SELLAMI	Amal	3	Modifier	Supprimer >>

Qui permet de lister les élèves afin de les modifier ou les supprimer, le lien « >> » permet d'afficher la vue **eleves.show** suivante :

## Gestion des élèves

ID Elève:

3

Nom:

SELLAMI

Prénom:

Amal

ID Club:

3

### Liste d'activités auxquelles l'élève a participé

ID activité	Description	Date début	Nombre de jours
4	Tournoi de football	2022-11-03	5
5	Tournoi de volley	2022-09-10	3
7	Tournoi de basketball	2022-10-07	2

Nombre total des jours: **10**

Qui permet de lister les activités auxquelles a participé un élève.

La vue « **eleves.create** » suivante :

## Gestion des élèves

### Ajouter un élève

ID élève:

Entrez l'ID de l'élève

Nom:

Entrez le nom de l'élève

Prénom:

Entrez le prénom de l'élèv

ID club:

1 Club des compétitions culturelles ▾

Ajouter

Annuler

Qui permet d'ajouter un élève, le bouton « **Annuler** » permet de revenir à l'index des élèves sans valider l'opération.

La vue « **eleves.edit** » suivante :

## Gestion des élèves

### Modifier un élève

ID élève:

3

Nom:

SELLAMI

Prénom:

Amal

ID club:

2 Club des compétitions sportives ▾

Modifier

Annuler

Qui permet d'inviter l'utilisateur à modifier un élève, le bouton « **Annuler** » permet de revenir à l'index des élèves sans valider l'opération.

Dans les deux vues « **eleves.create** » et « **eleves.edit** », la liste déroulante ID Club est remplie par les identifiants et les noms des clubs.

**Travail demandé :**

- 1) Donner les commandes permettant de créer les modèles **Eleve**, **Club** et **Activité** et de générer les migrations pour créer les tables **eleves**, **clubs** et **activites**. (1.5 pts)

```
php artisan make :model Eleve -m
```

```
php artisan make :model Club -m
```

```
php artisan make :model Activite -m
```

- 2) Editer les fonctions **up()** de chaque fichier de migration sachant que la table **eleves** doit contenir une clé étrangère qui fait référence à la table **clubs**. (3 pts)

```
public function up(){
    Schema ::create('eleves', function(Blueprint $table){
        $table->id() ;
        $table->string('nom') ;
        $table->string('prenom') ;
        $table->foreignId('club_id')->constrained() ;
    }
}

public function up(){
    Schema ::create('clubs', function(Blueprint $table){
        $table->id() ;
        $table->string('nom') ;
    }
}

public function up(){
    Schema ::create('activites', function(Blueprint $table){
        $table->id() ;
        $table->string('description') ;
        $table->date('dateDebut') ;
        $table->integer('nombreJours');
    }
}
```

- 3) Editer les modèles **Eleve**, **Club** et **Activité** en ajoutant les caractéristiques suivantes :

- a. L'accès à la liste des élèves d'un club dans le modèle **Club**. (1.5 pts)

```
public function eleves() : HasMany
{
    return $this->hasMany(Eleve ::class) ;
}
```

- b. L'accès au club de l'élève dans le modèle **Elève** (1.5 pts)

```
public function club() : BelongsTo
{
    return $this->belongsTo(Club ::class) ;
}
```

- c. Matérialiser la relation plusieurs à plusieurs entre les modèles **Elève** et **Activité** en ajoutant et éditant les migrations nécessaires de ou des tables pivots. (3 pts)

```
php artisan make :migration create_table_pivot_eleve_activite
```

```
public function up(){
    Schema ::create('eleve_activite', function(Blueprint $table){
        $table->id() ;
        $table->foreignId('eleve_id')->constrained() ;
        $table->foreignId('club_id')->constrained() ;
    }
}

public function eleves() : BelongsToMany
{
    return $this->belongsToMany(Activite ::class) ;
}
```

```

public function activites() : BelongsToMany
{
    return $this->belongsToMany(Club::class);
}

```

- 4) Donner les commandes permettant de créer puis éditer des seeders pour peupler les tables de la base de données. (1.5 pts)

```

php artisan make :seeder ElevesTablesSeeder
php artisan make :seeder ClubsTablesSeeder
php artisan make :seeder ActivitesTablesSeeder
public function run(){
    Eleve::create([
        'nom'=>'MAJDI',
        'prenom'=>'Farid',
        'club_id'=>2
    ]);
}

```

- 5) Donner les commandes permettant de lancer les migrations, et d'exécuter les seeders. (1 pt)

```

php artisan migrate
php artisan db :seed

```

- 6) Donner les commandes nécessaires à la créations des contrôleurs **EleveController**, **ClubController** et **ActiviteController** avec toutes les méthodes mentionnées dans le tableau ci-haut. (1.5 pts)

```

php artisan make :controller EleveController -r
php artisan make :controller ClubController -r
php artisan make :controller ActiviteController -r

```

- 7) Donner le code de la méthode index du contrôleur **EleveController** permettant d'afficher la vue **eleves.index**. (1.5 pts)

```

public function index(){
    $eleves=Eleve::all();
    return view('eleves.index', compact('eleves'));
}

```

- 8) Donner le code **html/blade** de la vue **eleves.edit** (utiliser l'html basique sans styler les éléments) : (3 pts)

- L'entrée '**ID élève**' est en lecture seule.
- La liste déroulante '**ID club**' affiche l'**ID** concaténé avec le nom du club.
- Le bouton « **Modifier** » valide la modification, le bouton « **Annuler** » revient vers l'index sans valider la modification.

```

<body>
<h1>Modifier un élève</h1>
<form action={{route('eleves.update',$eleve->id)}} method='post'>
    @csrf
    @method('put')
    ID élève :<input type='text' name='id' value='{{$eleve->id}}' readonly/><br/>
    Nom :<input type='text' name='nom' value='{{$eleve->nom}}'/><br/>
    Prénom :<input type='text' name='prenom' value='{{$eleve->prenom}}'/><br/>
    ID Club :<select name='club_id'>
        @foreach($clubs as $club)
            <option value='{{$club->id}}'>{{$club->id}} {{$club->nom}}</option>
        @endforeach
    </select><br/>
    <input type='submit' value='Modifier'/>
    <a href={{eleves.index}}>Annuler</a>
</form>
</body>

```

- 9) Donner le code des méthodes **edit** et **update** du contrôleur **EleveController** permettant d'afficher la vue **eleves.edit** et de valider la modification puis revenir à la vue **eleves.index**. (2 pts)

```

public function edit($id){
    $eleve=Eleve::find($id);
    $clubs=Club::all();
    return view('eleves.edit',[
        'eleve'=>$eleve,

```

```

        'clubs'=>$clubs
    ] ;
}
public function update(Request $request, $id){
    $eleve=Eleve ::find($id) ;
    $eleve->update([
        'nom'=>$request->nom,
        'prenom'=>$request->prenom,
        'club_id'=>$request->club_id
    ]) ;
    return redirect()->route('eleves.index') ;
}

```

- 10) Donner le code des méthodes **create** et **store** du contrôleur **EleveController** permettant d'afficher la vue **eleves.create** et de valider l'insertion puis revenir à la vue **eleves.index**. (2 pts)

```

public function create(){
    $clubs=Club ::all() ;
    return view('eleves.create',[
        'clubs'=>$clubs
    ]) ;
}
public function store(Request $request){
    $eleve=Eleve ::create([
        'id'=>$request->id,
        'nom'=>$request->nom,
        'prenom'=>$request->prenom,
        'club_id'=>$request->club_id
    ]) ;
    return redirect()->route('eleves.index') ;
}

```

- 11) Donner le code de la méthode **show** du contrôleur **EleveController** permettant d'afficher la vue « **eleves.show** ». (1.5 pts)

```

public function show($id){
    $eleve=Eleve ::find($id) ;
    return view('eleves.show', compact('eleve')) ;
}

```

- 12) Donner le code **html/blade** de la vue « **eleves.show** » (utiliser l'html basique sans styler les éléments). (4 pts)

```

<body>
    <h1>Gestion des élèves</h1>
    ID Elève : {{$eleve->id}}<br/>
    Nom : {{$eleve->nom}}<br/>
    Prénom : {{$eleve->prenom}}<br/>
    ID Club : {{$eleve->club_id}}<br/>
    <h2>Liste d'activités auxquelles l'élève a participé</h2>
    <table>
        <tr>
            <th>ID activité</th>
            <th>Description</th>
            <th>Date début</th>
            <th>Nombre de jours</th>
        </tr>
        @php($tot=0)
        @foreach($eleve->activites as $activite)
        <tr>
            <td>{{$activite->id}}</td>
            <td>{{$activite->description}}</td>
            <td>{{$activite->dateDebut}}</td>
            <td>{{$activite->nombreJours}}</td>
        </tr>
        @php($tot+=$activite->nombreJours)
    </table>
</body>

```

```
        @endforeach
    <table>
    Nombre total des jours :{{ $tot }}
</body>
```

- 13) En supposant que toutes les méthodes des contrôleurs **EleveController**, **ClubController** et **ActiviteConctroller** ont été définies, donner les routes nécessaires pour accéder aux URLs mentionnés dans le tableau ci-haut. (1.5 pts)

```
Route ::ressources([
    'eleves'=>EleveController ::class,
    'clubs'=>ClubController ::class,
    'activites'=>ActiviteController ::class
]);
```