

Semestre : 1 ☒ 2 ☐

Session : Principale ☒ Rattrapage ☐

Module: Application Côte Client 1

Enseignant(s): UP web

Classes : 4TWIN

Documents autorisés : OUI ☒ NON ☐

Calculatrice autorisée : OUI ☐ NON ☒

Date : 07/01/2025

Heure: 9h

Nombre de pages : 4

Internet autorisée : OUI ☐ NON ☒

Durée : 1h30

**A noter avant de démarrer :**

- 1- Seulement l'exécution du code est notée.
- 2- Les questions sont notées séparément. Si jamais vous voulez dépasser une question, vous pouvez faire ce qui vous convient pour pouvoir réaliser les questions d'après.
- 3- Le package « json-server » représente la partie backend de notre application et le fichier « db.json » représente la base de données.

**Enoncé :**

Nous voulons créer la partie frontend d'une application qui permet à un administrateur la gestion des matchs et des équipes avec le framework Angular version 16.

**Travail demandé :**

- 1- Créer un composant appelé **NavBarComponent** contenant trois liens comme indiqué dans la **Figure 1** et insérer le dans le composant AppComponent. (1 pt)

Liste des équipes   Liste des matchs   Ajouter match

Figure 1

- 2- Créer les composants et écrire le code nécessaire pour que :
  - En cliquant sur le lien « Liste des équipes », le composant **ListeEquipesComponent** est chargé au-dessous des liens. (0.75 pt)
  - En cliquant sur le lien « Liste des matchs », le composant **ListeMatchesComponent** est chargé au-dessous des liens. (0.75 pt)
  - En cliquant sur le lien « Ajouter match », le composant **MatchComponent** est chargé au-dessous des liens. (0.75 pt)
  - Au chargement de l'application, un composant **HomeComponent** que vous devez créer est chargé affichant le contenu de la **Figure 2** (0.25 pt)

Liste des équipes   Liste des matchs   Ajouter match

home works!

Figure 2



- 3- Créer un composant **FooterComponent** et insérer le dans le composant **AppComponent** affichant le message « **Match 2025** » (0.5 pt)

Soit les modèles de données « Equipe », « Match » et « Score » suivants :

Equipe	Match	Score
<b>id!</b> : string <b>nom!</b> : string <b>natif!</b> : string	<b>id!</b> : string <b>date!</b> : string <b>heure!</b> : string <b>lieu!</b> : string <b>equipeA:</b> Equipe = new Equipe() <b>equipeB:</b> Equipe = new Equipe() <b>score:</b> Score = new Score()	<b>equipeA!</b> : number, <b>equipeB!</b> : number

Alimenter le fichier db.json par le contenu suivant :

```
{
  "equipes": [
    { "id": "1", "nom": "Equipe 1", "natif": "Senior" },
    { "id": "2", "nom": "Equipe 2", "natif": "Junior" },
    { "id": "3", "nom": "Equipe 3", "natif": "Senior" },
    { "id": "4", "nom": "Equipe 4", "natif": "Junior" } ],
  "matches": []
}
```

- 4- Au chargement du composant **ListeEquipesComponent**, la liste des équipes est *affichée comme* indiqué dans la **Figure 3**. (2 pts)

The screenshot shows a web browser at localhost:4200/listequipes. It features three tabs: 'Liste des équipes', 'Liste des matchs', and 'Ajouter match'. The 'Liste des équipes' tab is active, displaying a list of four teams: 'Equipe 1 - Senior', 'Equipe 2 - Junior', 'Equipe 3 - Senior', and 'Equipe 4 - Junior'. Below the list, the text 'Match 2025' is visible.

Figure 3

- 5- Au niveau du composant **MatchComponent** ajouter le formulaire de la **Figure 4** (1 pt)

Les champs « EquipeA » et « EquipeB » sont deux listes déroulantes dont les valeurs sont récupérées à partir de la liste des équipes contenues dans le fichier db.json (1 pt)

The screenshot shows a form for creating a match. It includes fields for 'Date' (with a calendar icon), 'Heure' (with a clock icon), and 'Lieu' (a text input). Below these are two dropdown menus labeled 'Equipe A' and 'Equipe B'. At the bottom is a 'Soumettre' button.

Figure 4



- 6- Sachant que tous les champs sont obligatoires, afficher un message d'erreur pour chaque champ indiquant que ce champ est obligatoire. (1 pt)
- 7- Le bouton est désactivé tant qu'il existe un champ non rempli. Une fois tous les champs sont remplis le bouton devient actif. (0.5 pt)
- 8- En cliquant sur le bouton « Soumettre » :
- Si la date saisie est inférieure à la date d'aujourd'hui un message « **La date doit être supérieure ou égale à la date d'aujourd'hui** » est affiché. (1 pt)
  - Si l'administrateur a choisi la même équipe dans equipeA et equipeB un message d'erreur « **Il faut choisir deux équipes différentes** » s'affiche. (0.5 pt)
  - Si equipeA et equipeB n'ont pas la même valeur de l'attribut « natif » alors le message « **Les équipes doivent être de même natif** » s'affiche. (1 pt)
- 9- Une fois toutes les contraintes précédentes sont respectées, le match est ajouté dans le fichier « db.json » avec un score égal à 0 pour equipeA et 0 pour equipeB et une redirection se fait vers le composant ListeMatchesComponent. (2 pts)

NB : Un match est bien enregistré s'il correspond au format json suivant :

```
"matches": [
  {
    "id": "1", "date": "2025-01-10", "heure": "17:30", "lieu": "Rades",
    "equipeA": { "id": "1", "nom": "Equipe 1", "natif": "Senior" },
    "equipeB": { "id": "3", "nom": "Equipe 3", "natif": "Senior" },
    "score": { "equipeA": 0, "equipeB": 0 }
  }
]
```

- 10- Au chargement du composant ListeMatchesComponent, la liste des matchs est affichée comme indiqué dans la Figure 5 (2 pts)

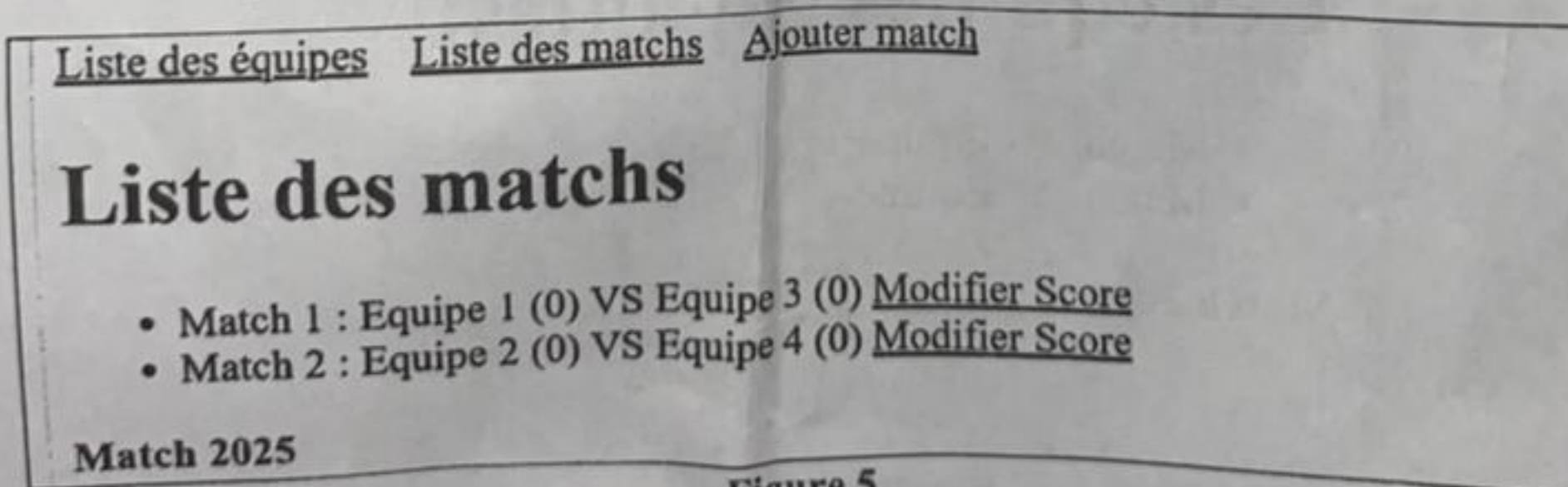


Figure 5

- 11- En cliquant sur le lien « Modifier Score » un nouveau composant à créer, appelé « UpdateScoreMatchComponent », est chargé en bas comme indiqué dans la Figure 6 affichant un formulaire contenant le score du match en question.
- NB : l'id du match est envoyé dans l'url comme mentionné dans la Figure 6. (2 pts)



← ↻ ⓘ localhost:4200/listematches/updatescore/1

[Liste des équipes](#) [Liste des matchs](#) [Ajouter match](#)

## Liste des matchs

- Match 1 : Equipe 1 (2) VS Equipe 3 (1) [Modifier Score](#)
- Match 2 : Equipe 2 (0) VS Equipe 4 (0) [Modifier Score](#)

Equipe A

Equipe B

Match 2025

Figure 6

- 12- En modifiant le score et en cliquant sur le bouton « **Modifier** », le score est modifié au niveau du fichier « db.json » et une redirection se fait vers la route liée au composant **ListMatchesComponent**. (1 pt)
- 13- Au niveau du composant **NavBarComponent**, ajouter un lien « **Résultat** », en cliquant dessus le module **ResultatModule** est lazy loaded affichant le contenu du composant **ResultatComponent** comme indiqué dans la Figure 7. (1 pt)

← ↻ ⓘ localhost:4200/listematches/updatescore/1

[Liste des équipes](#) [Liste des matchs](#) [Ajouter match](#) [Résultat](#)

## Les équipes gagnantes:

- Match: 1 - Equipe 1
- Match: 2 - Equipe 4

Match 2025

Figure 7