

EXAMEN

Semestre : 1 ☒ 2 ☐

Session : Principale ☒ Rattrapage ☐

Module : JavaScript Côté Serveur

Enseignants : Équipe Web

Classes : 4TWIN

Documents autorisés : OUI ☒ NON ☐

Calculatrice autorisée : OUI ☐ NON ☒

Date : 06/01/2025

Nombre de pages : 4

Internet autorisée : OUI ☐ NON ☒

Heure : 11h00

Durée : 01h30

Soit les deux entités « Bibliotheque » et « Livre » suivantes :

Bibliotheque
+ nom :string
+ nbr_livre : number
+ adresse :string

Livre
+ titre :string
+ auteur :string
+ etat : string
+ date_publication :Date
+ id_bibliotheque :string

Figure 1: Collections Bibliotheque et Livre

Vous êtes amené à créer une application web en utilisant les technologies :
Nodejs/Express MongoDB et mongoose qui permet la gestion des bibliothèques.

- 1) Créer la base de données "myexam2025" avec les deux entités Bibliotheque et Livre (voir figure n°1). (1pt)
- 2) Créer une fonction qui permet d'ajouter une bibliothèque avec la propriété nbr_livre initialisée à 0 (voir figure n°2). (1 pt)

POST ☒ http://localhost:3000/bibliotheque/create

```
{
  "nom": "Bibliotheque El Najeh",
  "nbr_livre": 0,
  "adresse": "Tunis",
  "_id": "67688ddb651d5d52719f67cb",
  "__v": 0
}
```

Figure 2: Ajouter une nouvelle bibliothèque

- 3) Créer une fonction (voir figure n°3) qui permet d'afficher la liste des bibliothèques. (1 pt)

GET ☒ http://localhost:3000/bibliotheque/list

Figure 3: Liste des bibliothèques

- 4) Créer une fonction (voir figure n°4) qui permet de récupérer une bibliothèque via son Id. (1 pt)

GET	✓	http://localhost:3000/bibliotheque/details/67688ddb651d5d52719f67cb
-----	---	---

Figure 4: Récupérer une bibliothèque via son Id

- 5) Créer une fonction "Supprimer" (voir figure n°5) qui permet de supprimer une bibliothèque via son Id. (1 pt)

DELETE	✓	http://localhost:3000/bibliotheque/delete/676882e64599dd3f12ba1a69
--------	---	--

Figure 5 : Supprimer une bibliothèque

- 6) Créer une fonction qui permet de créer un livre (voir figure n°6) et l'affecter à une bibliothèque via son « id », l'état du livre est initialisé à « disponible » puis incrémenter à 1 le nombre de livre de la bibliothèque auquel appartient ce livre. (2 pts)

POST	✓	http://localhost:3000/livre/create/67688ddb651d5d52719f67cb
------	---	---

Figure 6: Ajouter un livre

- 7) Créer un middleware qui contrôle les données pour l'ajout d'un Livre en utilisant la librairie "yup" (voir Figure 7), avec les spécifications suivantes :

- Le titre est obligatoire et doit être une chaîne de caractères d'une longueur d'au moins 3 caractères. (0.5 pt)
- L'auteur est obligatoire, doit être une chaîne de caractères d'une longueur d'au moins 3 caractères. (0.5 pt)
- La date est obligatoire et doit avoir un format d'une date et la date doit être supérieure à la date d'aujourd'hui. (1 pt)

```
{
  "error": [
    "La date de publication doit être ultérieure à aujourd'hui"
  ]
}
```

Figure 7: Erreur de validation 'yup' lors de l'ajout d'un livre

- 8) Élaborer une fonction nommée "location" qui prend en paramètre l'id du livre tels qu'illustrés dans la figure n°8. Cette fonction a pour but de mettre à jour l'état à « loué » puis décrétement à 1 le nombre de livre de la bibliothèque auquel appartient ce livre. (3 pts)

NB : Si l'état_construction du niveau est à « loué ». Il ne faut pas mettre à jour l'état du livre et afficher une erreur qui indique que le livre est loué.

Voici un exemple illustrant ces actions :

- Bibliothèque avant la location :	- Bibliothèque après la location :
<pre>{ _id : 67688ddb651d5d52719f67cb nom : 'Bibliotheque El Najeh' nbr_livre : 3 adresse : 'Tunis' }</pre>	<pre>{ _id : 67688ddb651d5d52719f67cb nom : 'Bibliotheque El Najeh' nbr_livre : 2 adresse : 'Tunis' }</pre>
- Livre avant la location :	- Livre après la location :
<pre>{ _id : 6768826a4599dd3f12ba1a5f titre : 'Harry Potter à l'école des sorciers' auteur : 'J. K. Rowling' etat : 'disponible' date_publication : 26-06-1997 id_bibliothèque : 67688ddb651d5d52719f67cb }</pre>	<pre>{ _id : 6768826a4599dd3f12ba1a5f titre : 'Harry Potter à l'école des sorciers' auteur : 'J. K. Rowling' etat : 'loué' date_publication : 26-06-1997 id_bibliothèque : 67688ddb651d5d52719f67cb }</pre>

PUT ✓ http://localhost:3000/livre/location/6768826a4599dd3f12ba1a5f

Figure 8 : Location d'un livre

- 9) Créer une fonction qui permet d'afficher la somme des livres loués et des livres non loués de chaque bibliothèque (voir figure n°9). (1 pt)

GET ✓ http://localhost:3000/livre/resume

```
[
  {
    "bibliothèque": "Bibliothèque El Amal",
    "loués": 2,
    "disponibles": 2
  },
  {
    "bibliothèque": "Bibliothèque El Zouhour",
    "loués": 1,
    "disponibles": 1
  },
  {
    "bibliothèque": "Bibliothèque El Najeh",
    "loués": 0,
    "disponibles": 1
  }
]
```

Figure 9: Interface de location d'un livre

- 10) Créer une nouvelle page web (voir figure n°10) où on y trouvera une partie consacrée à l'affichage des informations des livres puis afficher la liste des livres **non loué** (état : '**disponible**') (voir figure n°10) en utilisant Socket.io. (2 pts)
NB : Créer un champ qui permet de saisir un id d'un livre et un bouton « location » et l'affichage de la liste des livres disponible se fait automatiquement en ouvrant cette page web.

Location des livres

Id livre : Location

Id : 67688155a489239415d600ad | Titre : Harry Potter | Auteur : J. K. Rolling | Date de publication : 2024-12-23T00:00:00.000Z | État : disponible
Id : 6768826a4599dd3f12ba1a5f | Titre : Harry Potter 1 | Auteur : J. K. Rolling | Date de publication : 2024-12-25T00:00:00.000Z | État : disponible
Id : 676882794599dd3f12ba1a62 | Titre : Harry Potter 2 | Auteur : J. K. Rolling | Date de publication : 2025-12-25T00:00:00.000Z | État : disponible
Id : 67689161651d5d52719f6937 | Titre : Harry Potter 1 | Auteur : J. K. Rolling | Date de publication : 2025-12-25T00:00:00.000Z | État : disponible

Figure 10 : Interface de location d'un livre

- 11) Lorsque on remplit le champ par l'id d'un livre disponible et qu'on clique sur le bouton « location » la fonction location (question 8) va être exécutée puis on aura la nouvelle liste mise à jour des livres disponible (voir figure n°11) en utilisant Socket.io. (3 pts)

Location des livres

Id livre : Location

Id : 67688155a489239415d600ad | Titre : Harry Potter | Auteur : J. K. Rolling | Date de publication : 2024-12-23T00:00:00.000Z | État : disponible
Id : 6768826a4599dd3f12ba1a5f | Titre : Harry Potter 1 | Auteur : J. K. Rolling | Date de publication : 2024-12-25T00:00:00.000Z | État : disponible
Id : 676882794599dd3f12ba1a62 | Titre : Harry Potter 2 | Auteur : J. K. Rolling | Date de publication : 2025-12-25T00:00:00.000Z | État : disponible
Id : 67689161651d5d52719f6937 | Titre : Harry Potter 1 | Auteur : J. K. Rolling | Date de publication : 2025-12-25T00:00:00.000Z | État : disponible

Id : 6768826a4599dd3f12ba1a5f | Titre : Harry Potter 1 | Auteur : J. K. Rolling | Date de publication : 2024-12-25T00:00:00.000Z | État : disponible
Id : 676882794599dd3f12ba1a62 | Titre : Harry Potter 2 | Auteur : J. K. Rolling | Date de publication : 2025-12-25T00:00:00.000Z | État : disponible
Id : 67689161651d5d52719f6937 | Titre : Harry Potter 1 | Auteur : J. K. Rolling | Date de publication : 2025-12-25T00:00:00.000Z | État : disponible

Figure 11 : Affichage de la liste des livres disponible après location

- 12) Créer une fonction qui s'exécute périodiquement chaque 5 secondes. Cette fonction affiche une notification s'il y a eu une nouvelle location d'un livre (voir figure n°12). (2pts)

NB : La notification sera affichée si on loue un livre via l'**interface** et aussi si on loue un livre via **postman**

Location des livres

Id livre : Location



Figure 12 : Afficher une notification lors de la location d'un nouveau livre

Bon Travail 😊