

ING1 - Développement WEB

vendredi 29 Mai 2026

13h15 - 15h15

Contrôle continu

Jordan SABLON

EIL Côte d'Opale, Calais

Modalités de l'examen

- L'épreuve est à réaliser sur un ordinateur des salles informatiques de l'école
- L'accès à Internet n'est pas autorisé durant l'épreuve
- Seuls les documents fournis avec le sujet d'examen sont autorisés

Procédure pour rendre votre évaluation

Dans le dossier backend : supprimer le dossier « node_modules » s'il existe.

Dans le dossier frontend : supprimer les dossiers « node_modules » et « .next » s'ils existent.

Compressez votre dossier code dans une archive au format .zip ou .rar nommée « **web_NOM_Prenom** » et déposez-le à cet endroit sur votre machine : « V:/donnees/TESTEXAM ».

⚠ Le non respect du format de dépôt ou du nommage de l'archive entrainera une pénalité sur la note de l'examen

Pré-requis techniques

Afin de pouvoir réaliser cet examen, votre ordinateur doit disposer des logiciels suivants :

- NodeJS
- NPM
- VS Code (ou tout autre environnement de développement de votre choix)
- Un navigateur web

Barème de notation

Le barème de notation est donné à titre indicatif à la fin du sujet d'examen.

Précisions

- Vous disposez des supports et codes des TDs et du TP dans le dossier « ressources »
- **⚠ Attention à bien récupérer le contenu qui se trouve dans le dossier « V:/donnees/SUJET » et à bien travailler depuis votre session au risque de ne pas pouvoir sauvegarder votre travail !**

Configuration de l'environnement de développement

1. Récupération du code source

Récupérer le code source fourni avec le sujet d'examen et ouvrir le dossier « code » dans VS Code (ou l'IDE de votre choix)

2. Backend

Depuis le dossier « backend », exécutez dans un terminal la commande suivante pour démarrer le serveur : « **npm start** »

Une fois démarré, le serveur est accessible via l'adresse (<http://localhost:5000/>) et doit afficher « Hello ! »

3. Frontend

L'application React présente dans le dossier frontend a été fournie sans les dépendances. Il n'est pas nécessaire de démarrer l'application pour répondre aux questions concernant cette partie.

Présentation du sujet

Vous devez répondre aux questions ci-dessous. Chaque question est indépendante et ne nécessite pas que la précédente soit traitée pour pouvoir y répondre, vous pouvez donc traiter les questions dans l'ordre qui vous convient.

⚠ Comme vous ne disposez pas de l'extension REST Client sur VS Code, l'ensemble des routes peuvent être testées depuis le navigateur.

Question n°1 :

Implémentez une route **GET /books** qui permet de retourner l'ensemble des livres présents dans le jeu de données, sans les attributs pages et tags . Les livres retournés seront triés par ordre alphabétique.

Question n°2 :

La route **GET /books/:id** ne fonctionne pas. Trouvez la ou les raisons et apportez les modifications nécessaires pour que la route soit de nouveau fonctionnelle.

Question n°3 :

Une route **GET /books/instock** a été créée afin de récupérer tous les livres en stock. Le résultat est correct mais le code peut être amélioré. Apportez les modifications nécessaires afin de simplifier au maximum le code existant.

Question n°4 :

Créez une route **GET /books/format/:format** qui retourne uniquement les livres qui correspondent au format passé en paramètre.

Question n°5 :

Un utilisateur souhaite afficher la liste des livres disponibles quand il accède au composant `<Books />`. Il dispose également d'un bouton qui permet d'actualiser la liste des livres lorsque l'on clique dessus. Toutefois, l'utilisateur rencontre des problèmes pour récupérer les livres. Vous devez identifier les éventuels problèmes afin de les corriger et ajouter le code manquant si nécessaire.

Question n°6 :

L'utilisateur souhaite, toujours dans le composant `<Books />`, afficher en gras les livres dont la note est supérieure à 4.5/5. Implémentez le code nécessaire.

Pour afficher du texte en gras, on peut utiliser la balise HTML `<b></b>`, exemple :
`<b>Ceci est un texte en gras</b>`

Question n°7 :

Dans le composant `<Fibonacci />`, on dispose d'une fonction `fibonacciNumbers()` qui permet de déterminer les 10 premiers nombres de la suite de Fibonacci.

Lorsque l'on fait appel à cette fonction, ces nombres sont actuellement simplement affichés dans la console du navigateur.

On souhaite afficher en gras et en dessous les uns des autres ces nombres dans le composant `<Fibonacci />`. La fonction qui permet de déterminer les 10 premiers nombres de Fibonacci sera automatiquement appelée lorsque l'on accède au composant `<Fibonacci />`.

Implémentez le code nécessaire.

Indications techniques

Tri d'un tableau

Les données d'un tableau en Javascript peuvent être triées en utilisant la fonction `sort()`. Voici un exemple d'utilisation :

```
const months = ["March", "Jan", "Feb", "Dec"];
months.sort();
console.log(months);
// Résultat attendu : Array ["Dec", "Feb", "Jan", "March"]
```

```
const array = [1, 30, 4, 21, 100000];
array.sort();
console.log(array);
// Résultat attendu : Array [1, 100000, 21, 30, 4]
```

Tri d'un tableau d'objets selon un de ses attributs

Lorsque l'on souhaite trier un tableau contenant des objets selon un attribut de cet objet, on peut utiliser la fonction `localeCompare()` avec la fonction `sort()`. Voici un exemple d'utilisation où l'on souhaite récupérer les utilisateurs triés par age croissant :

```
type User = {
  name: string,
  age: number
}

const usersWithAge : User[] = [{name: 'John', age: 18}, {name: "Alan", age: 20}, {name: "Martin", age: 19}];

const usersByAge = usersWithAge.sort((a, b) => a.name.localeCompare(b.name))
```

Barème

Le barème donné ci-dessous est uniquement à titre indicatif et est susceptible d'être adapté lors de la correction du contrôle.

Partie n°1 - Backend NodeJS : 11 points

Question n°1	Implémentation de la route GET /books	4 points
Question n°2	Correction de la route GET /books/:id	2 points
Question n°3	Optimisation de la route GET /books/instock	2 points
Question n°4	Implémentation de la route GET /books/format/:format	3 points

Partie n°2 - Frontend React : 9 points

Question n°5	Correction des problèmes lors de l'affichage des livres	3 points
Question n°6	Affichage en gras des livres dont la note est supérieure à 4.5/5	2 points
Question n°7	Affichage des nombres de la suite de Fibonacci	4 points

Fin de l'énoncé