**Les documents sont interdits. Durée 90’.**

Répondre au **stylo bille foncé** en **noircissant complètement** la case correspondante sur **les feuilles de réponse fournies**. Pour les questions ouvertes, écrire la réponse dans le cadre et ne pas noircir les cases de ces questions (qui sont destinées au correcteur du contrôle).

Les questions à choix multiple **avec** le symbole ♣ peuvent présenter *zéro, une ou plusieurs bonnes réponses*. Ces questions sont notées sur 2 points avec -1 point par réponse fausse (0 minimum par question). Les questions à choix multiple **sans** le symbole ♣ ont une *unique bonne réponse*. Ces questions sont notées sur 2 points avec 2 points si la réponse est correcte, 0 s’il n’y a pas de réponse et -0.5 si la réponse est fausse. Le barème des questions ouvertes est précisé sur chacune.

Merci de remplir correctement les cases, et d’écrire de manière à ce que les correcteurs puissent lire et comprendre vos réponses. Un malus pourra être appliqué si la copie n’a pas été correctement remplie ou est trop difficile à comprendre.

1 DOM et CSS

Question 1 (/4) Écrire le code JavaScript qui utilise le template suivant pour afficher la liste de noms dans un élément `<div id="contacts">`.

```
1 <template id="name-template">
2   <div class="name-card">
3     <p>{{name}}</p>
4   </div>
5 </template>
6
7 <script>
8   const names = ["Alice", "Bob", "Charlie"]; // Liste de noms
9   // a completer
10 </script>
```

Question 2 (/2) Quel est l’ordre de spécificité des sélecteurs CSS ? On note $x > y$ pour indiquer que x est plus spécifique que y .

A .class > #id > element.

C element > .class > #id.

B #id > .class > element.

D #id > element > .class.

Question 3 (/4) On souhaite créer et insérer dans l’élément `` un bouton intitulé “Démarrer”. Écrire le code javascript permettant de manipuler le DOM à cette fin en suivant les bonnes pratiques vues en cours et TP.

Question 4 (/2) Avec Bulma, quelle classe est utilisée centrer un `<div>` avec une largeur maximale adaptée à la taille de l’écran ?

A center

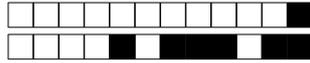
D box

B columns

E container

C section

F frame



2 Programmation JavaScript générale

Question 5 (/4) Expliquer à quoi sert l'opérateur spread ... en JavaScript et donnez un exemple d'utilisation.

Question 6 (/4) Écrire le code javascript pour doubler chaque nombre, puis ne garder que ceux supérieurs à 9 en utilisant des **Array methods** (méthodes de tableaux) et sans boucle.

Vous avez le tableau suivant `const numbers = [2, 5, 7, 10]` ;, et le résultat sera une constante nommée `result`.

Question 7 (/6) Le code ci-dessous utilise la méthode `.then()` pour gérer une promesse. Réécrivez-le en utilisant la syntaxe `async/await` :

```
1 fetch("https://api.example.com/data")
2   .then((response) => response.json())
3   .then((data) => {
4     console.log(data);
5   })
6   .catch((error) => {
7     console.error("Erreur :", error);
8   });
```

Question 8 (/2) Dans le code ci-dessus, que se passe-t-il si le serveur retourne une erreur HTTP (comme 404 ou 500) ?

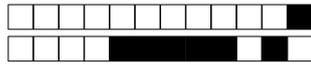
- A Le bloc `catch` est automatiquement exécuté
- B Le flot d'exécution sort au niveau du 1e `.then()` pour se poursuivre
- C Le programme s'interrompt avec une exception
- D La 2e méthode `.then()` est exécutée, mais `response.ok` vaut `false`

Question 9 (/2) En quoi consiste la stratégie "Fail First" dans le cadre des requêtes d'API côté client ?

- A à réessayer automatiquement la requête en cas d'échec jusqu'à ce qu'elle réussisse.
- B à envoyer plusieurs requêtes à la fois pour tester l'API sous haute charge.
- C à vérifier immédiatement les erreurs après chaque étape de la requête pour éviter de traiter des données incorrectes.
- D à envoyer plusieurs requêtes à la suite pour minimiser le temps de réponse.

Question 10 (/6) Indiquer quel(les) ligne(s) du code faudrait il modifier pour respecter la stratégie fail first, expliquer le code à insérer.

Question 11 (/4) L'API vous signale que vous l'avez trop demandé, quel code HTTP renvoie t'elle ? Quelle est la classe de réponse (la centaine du code), expliquer à quoi correspond la classe.



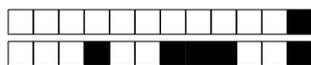
3 Programmation Web en Node.js

On rappelle que les principales méthodes HTTP, appelées aussi verbes HTTP, utilisées pour effectuer ces opérations sont POST, GET, PUT et DELETE. Dans cette partie, on s'intéresse à la programmation Web en Node.js et à l'architecture utilisée dans le TP5. On donne à cet effet un extrait du code du routeur pour l'API de raccourcissement d'URL ci-après.

```
1 import Joi from "joi";
2 import Boom from "@hapi/boom";
3
4 async function validateExpirationDate(request, h) {
5   const date = request.payload.expires;
6   if (date < Date.now()) {
7     throw // A compléter
8   }
9   return true;
10 }
11
12 export default {
13   register: async function (server, _options) {
14     const { db } = server.app;
15     /* (A) */
16     server.route({
17       method: "POST",
18       path: "/",
19       handler: async (request, h) => {
20         const { X, Y } = request.payload;
21         /* (B) */
22         const link = await db.createLink(uri, expires);
23         /* (C) */
24         return h
25           .response({
26             ...link,
27             uri: `${server.info.uri}${prefix}/${link.short}`,
28           })
29           .code(201);
30       },
31       options: {
32         validate: {
33           /* (D) */
34           payload: Joi.object({
35             uri: Joi.string().uri().example("http://perdu.com"),
36             expires: Joi.date().optional().label("Expiration date"),
37           }),
38         },
39       },
40     });
41
42     /* ensuite, les autres routes sont configurées */
43   },
44 };
```

Question 12 ♣ (/2) Quel est l'intérêt d'utiliser SQLite plutôt que PostgreSQL dans un projet Node.js ?

- A SQLite ne nécessite pas de serveur de base de données séparé.
- B SQLite propose une meilleure gestion native des dates que PostgreSQL.
- C SQLite permet une scalabilité horizontale plus performante que PostgreSQL.
- D SQLite convient bien pour les tests, les prototypes ou les applications locales.



Question 13 (/2) Quel est le principal avantage de Node.js en termes de montée en charge dans le contexte de la programmation Web côté serveur ?

- A La parallélisation implicite de toutes les fonctions JavaScript.
- B L'équilibrage automatique de la charge.
- C Les opérations I/O non bloquantes.
- D Le moteur JavaScript V8.

Question 14 ♣ (/2) Quelles commandes permettent de tester une route POST /login dans une API Node.js, avec les identifiants ?

- A `curl -X POST http://192.168.111.3/login -d "user=admin&pass=1234"`
- B `ssh admin@192.168.111.3 /login`
- C `http POST http://192.168.111.3/login user=admin pass=1234`
- D `scp admin@192.168.111.3:/login`
- E `ping 192.168.111.3/login`

Question 15 ♣ (/2) Dans le TP5c, un des compte fourni n'est pas `sudoer`, à quoi sert il ?

- A Faciliter le déploiement de l'application.
- B Exécuter l'application Node.
- C Configurer le serveur nginx
- D Redémarrer le serveur en cas de panne.

Question 16 (/4) Donner deux inconvénients de la stratégie de rendu-côté client (*Client-Side Rendering - CSR*) comparé celle côté serveur (*Server-Side Rendering - SSR*).

Question 17 (/4) Expliquez de façon concise le rôle d'un reverse proxy dans une architecture web.

Question 18 (/4) Dans la route POST / ligne 20, par quoi devrait être remplacé X et Y pour que le code soit fonctionnel ?

Question 19 (/2) A quel endroit devrait être appelé la fonction `validateExpirationDate` ? (voir commentaires dans le code pour les positions)

- A (B)
- B (A)
- C (D)
- D (C)

Question 20 (/4) Dans la route POST / compléter le code ligne 7, en respectant la logique de l'application de raccourcisseur d'URL du TP5.

Question 21 (/8) On considère l'API Web d'une application de partage de dépenses (type `tricont` ou `splitwise`) permettant aux utilisateurs de suivre et partager des dépenses entre amis. L'API doit permettre aux utilisateurs de créer, mettre-à-jour, consulter, et supprimer des dépenses au sein de leur groupe, ainsi que calculer combien chaque membre du groupe doit donner/recevoir. Proposer une API Web en utilisant les verbes HTTP appropriés et les chemins associés pour ces cinq routes.

Question 22 (/4) Dans l'application de partage de dépenses, on souhaite ajouter une fonctionnalité pour que seuls les utilisateurs qui ont ajouté une dépense puisse la supprimer. En vous inspirant de ce qui a été réalisé dans le TP5, expliquer comment cela pourrait être implémenté. En considérant les différences avec le raccourcisseur d'URL, expliquer en quoi l'implémentation pourrait être différente (en une phrase).