Examen technologies avancées du web - A.5.03 (30) TD.36.306 (1) - 17/05/2022
Partie Pierre Pompidor

1. Interrogation sous Mongo (1 point)

Dans la ligne suivante qui interroge une collection *Produits* sous Mongo

```
db.Produits.find(...)
```

remplacer les ... par le code sélectionnant les documents qui possèdent une propriété *prix* dont la valeur est comprise entre *10* et *100*.

2. Service web (2 points)

Soit une collection *clients* gérée par un serveur MongoDB, chaque objet de la collection comprenant entre autres les propriétés *mail*, *nom* et *prénom*. Ecrire le service web (d'un serveur Node.js) qui sur une route contenant en paramètre l'adresse email du client renvoie son nom et son prénom (la connexion préalable à la base MongoDB n'est pas à écrire).

3. Requêtes imbriquées (1.5 points)

Décrivez le problème qui peut survenir lors de requêtes imbriquées à un serveur Mongo et le moyen de le corriger (sans implémenter la solution). Voici un exemple générique qui peut poser problème (les ... désignant du code) :

```
db.collection(...).find(...).toArray((err, documents) => {
   let resultats = [];
   for (let doc of documents) {
      db.collection(...).find(...).toArray((err, documents) => {
         if (documents != undefined && documents.length == 1) resultats.push(doc);
      });
   }
   // renvoi du résultat
   ...
});
```

4. Fragment de template Angular (2 points)

Une liste d'objets nommée *articles* est déclarée dans une classe Angular, chaque objet ayant les propriétés *nom* et *stock*. Ecrire le fragment du template associé à cette classe qui affiche dans une liste HTML les noms d'articles qui ont un stock supérieur à 0.

5. Table de routage du router Angular (1,5 points)

Donnez le nom du fichier dans lequel se trouve la table de routage principale du router Angular.

Donnez le contenu de l'objet *routes* de telle sorte que le composant *ConnexionComponent* soit invoqué sur la route */user/connexion*.

6. Quel est le rôle de la balise <router-outlet> ? (1 point)

7. Instanciation d'objet (1 point)

Pourquoi un composant Angular va-t-il utiliser ainsi le service *HttpClient* :

```
...
export class monService {
  constructor(private http: HttpClient) { }
  ...
}
```

alors que la programmation objets "classique" nous pousserait à le mettre en oeuvre comme ceci :

```
...
export class monService {
  private http = new HttpClient();
  ...
}
```