

**Final Exam (Contrôle)**

Name: .....

Group: .....

**Exercise 1: Quiz (6 pts)****Part1:** Please check the answers you think are correct. / *Cocher les réponses que vous pensez être correctes.*

1- Which of the following is NOT a front-end framework?

*Lequel des éléments suivants n'est PAS un framework front-end ?*

- ☐ Node.js
- ☐ Angular
- ☐ React
- ☐ Ember

2- What is the main advantage of using a front-end framework?

*Quel est le principal avantage d'utiliser un framework front-end ?*

- ☐ Improved website security / *Sécurité améliorée du site Web*
- ☐ Better code organization and maintainability / *Meilleure organisation et maintenabilité du code*
- ☐ Enhanced search engine optimization / *Optimisation améliorée des moteurs de recherche*
- ☐ It makes the website look more attractive / *Rend le site Web plus attrayant*

3- Which of the following is a valid way to create a React component?

*Lequel des éléments suivants est une façon valide de créer un composant React ?*

- ☐ `class MyComponent extends Component`
- ☐ `const MyComponent = () => {}`
- ☐ `function MyComponent () {}`
- ☐ `<MyComponent />`

4- What is the role of virtual DOM in React?

*Quel est le rôle du DOM virtuel dans React ?*

- ☐ Optimize SQL queries / *Optimiser les requêtes SQL*
- ☐ Improve website security / *Améliorer la sécurité du site web*
- ☐ Improve performance by minimizing DOM manipulations / *Améliorer les performances en minimisant les manipulations du DOM*
- ☐ Manage server-side rendering / *Gérer le rendu côté serveur*

5- What tool does React use to compile JSX?

*Quel outil React utilise-t-il pour compiler JSX ?*

- ☐ Babel
- ☐ Jest
- ☐ ReactDOM
- ☐ JSX Compiler

**Part2:** Answer the following questions. / *Répondre aux questions suivantes*

1- Explain the concept of Components in React?

*Expliquer le concept des composants dans React?*

.....

.....

.....

2- What are the differences between State and props in React?

*Quelles sont les différences entre State et props dans React?*

.....

.....

.....

Exercise 2: (6 pts)

What is the output of the following code?

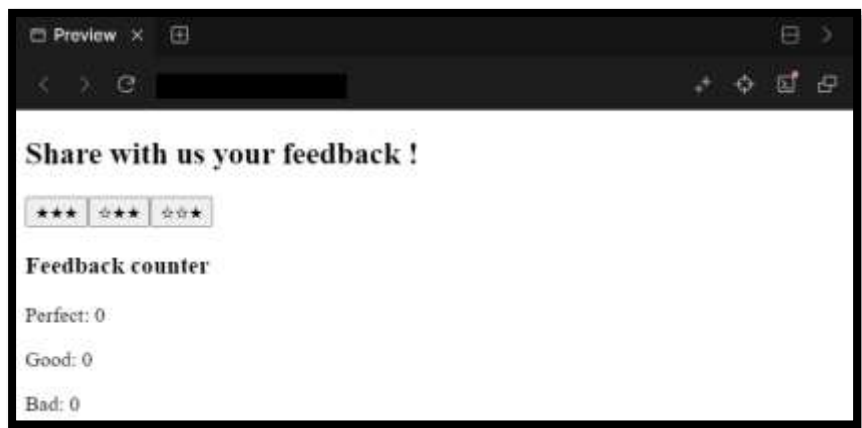
Qu'affichent les codes JavaScript suivants?

<pre>const a = [1, 2, 3]; const b = [4, 5, 6]; const c = [...a, ...b]; console.log(c);</pre>	
<pre>const make = 'Ford'; const model = 'Mustang'; const car = { make, model }; console.log(car);</pre>	
<pre>function Car() {   return &lt;h1&gt;Ford Mustang&lt;/h1&gt;; } const root = createRoot(document.getElementById('root')); root.render(&lt;Car /&gt;);</pre>	
<pre>var instructor = {   fullName: 'Abu Obaida',   saySalam: function(){     console.log("Salam " + this.fullName);   } } instructor.saySalam()</pre>	
<pre>let x = 10; function foo() {   let x = 20;   console.log(x); } foo(); console.log(x);</pre>	
<pre>function format(pattern, ...params) { return ({pattern, params}); } console.log(format(1, 2, 3)); console.log(format(4));</pre>	

Exercise 3: (8 pts)

A web application uses the following page (see display below) to obtain and display feedback from its clients (the figures displayed are examples):

Une application web utilise la page suivante (voir affichage ci-dessous) pour obtenir et afficher les feedbacks de ses clients (les chiffres affichés sont des exemples):



There are 3 possible choices for feedback in the form of 3 buttons: **3stars (Perfect)**, **2stars (Good)** and **1star (Bad)**. The application displays the number of clicks on each button.

- 🟢 1- Complete the code below to obtain this display by adding:
- a) the JSX code allowing the indicated display to be obtained, knowing that stars (★☆) are ascii code and titles are level-two and level-three headings respectively <h2> and <h3>.

b) the **handlePerfect**, **handleGood** and **handleBad** methods which allow you to process clicks on the 3 buttons.

Il existe 3 choix possibles de feedback sous la forme de 3 boutons : 3 étoiles (Parfait), 2 étoiles (Bon) et 1 étoile (Mauvais). L'application affiche le nombre de clics sur chaque bouton.

🟢 1- Complétez le code ci-dessous pour obtenir cet affichage en ajoutant :

- le code JSX permettant d'obtenir l'affichage indiqué, sachant que les étoiles (★) sont du code ascii et les titres sont en-têtes de type `<h2>` et `<h3>` respectivement.
- les méthodes `handlePerfect`, `handleGood` et `handleBad` qui permettent de traiter les clics sur les 3 boutons.

```
import React, { useState } from "react";

function FeedbackApp() {
  const [perfect, setPerfect] = useState(0);
  const [good, setGood] = useState(0);
  const [bad, setBad] = useState(0);

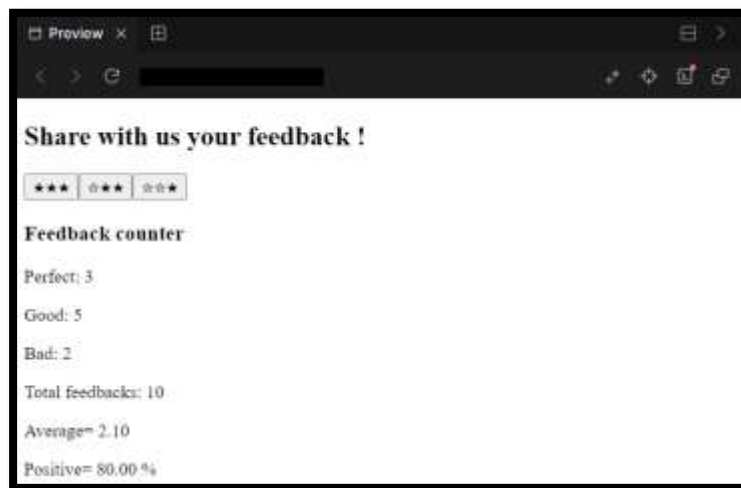
  // Here you add methods for processing clicks, answer to question 1.b)

  return (
    <div>
      // Here you add answer to question 1.a)
    </div>
  );
}

export default FeedbackApp;
```

🟢 2- Add to your application the code to display more information about the returned feedback as follows:

Ajouter à votre application le code permettant d'afficher plus d'informations sur le feedback retourné comme suit:



Knowing that / Sachant que:

- Total feedbacks** is number of all feedbacks obtained / est le nombre de tous les feedbacks obtenus.
- Average** is the average score (average stars obtained) / est la note moyenne (moyenne des étoiles obtenues)
- Positive** is the percentage of positive feedback / est le pourcentage de commentaires positifs,  

$$\text{positive} = ((\text{perfect} + \text{good}) / \text{all}) * 100$$

🟢 3- Modify your code to allow displaying the requested information in a separate component called **Statistics** with props, which allows displaying the information: **Perfect, Good, Bad, Total feedbacks, Average** and **Positive**. For that:

- Write the content of the **Statistics** component which displays the requested information

```
const Statistics = (props) => {
  const {perfect, good, bad} = props;
  // ...}
}
```

- Edit JSX code in **FeedbackApp** component now using **Statistics** component

🟢 3- Modifiez votre code pour permettre l'affichage des informations demandées dans un composant distinct appelé **Statistics** avec props, qui permet d'afficher les informations **Perfect, Good, Bad, Total feedbacks, Average** and **Positive**. Pour ça:

- Ecrire le contenu du composant **Statistics** qui affiche les informations demandées

```
const Statistics = (accessoires) => {
  const {perfect, good, bad } = props;
  //...}
}
```

- Modifiez maintenant le code JSX dans le composant **FeedbackApp** à l'aide du composant **Statistics**