

Exercice1 :

Terminer le code source :

1. Question1

```
import React from "react";

function Counter() {
  // Quel hook utiliser ici ?
  const [count, setCount] = .....(0);

  return (
    <div>
      <h2>Compteur : {count}</h2>
      <button onClick={() => .....(count + 1)}>Incrémenter</button>
    </div>
  );
}

export default Counter;
```

2. Question2

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

function AutoFocusInput() {
  const inputRef = .....(null);

  _____(() => {
    inputRef.current. -----;
  }, []);

  return <input type="text" placeholder="Votre nom" .....={inputRef} />;
}

export default AutoFocusInput;
```

3. Question3 : Afficher la liste des utilisateurs récupérés depuis une API (<https://jsonplaceholder.typicode.com/users>).

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

function UserList() {
  const [users, setUsers] = .....([]);

  .....(() => {
    fetch("https://jsonplaceholder.typicode.com/users")
      .then(res => res.json())
      .then(data => setUsers( ..... ))
  });
}
```

```

}, []);

return (
  <ul>
    {users..... (u => (
      <li .....={u.id}>{u.name}</li>
    ))}
  </ul>
);
}
..... UserList;

```

4. Question 4 : Thème global clair/sombre avec useContext

Fichier ThemeContext.js

```

import React, { ..... , ..... } from "react";

const ThemeContext = .....();
export function ThemeProvider({ children }) {
  const [theme, setTheme] = useState("light");

  const toggleTheme = () => setTheme(theme === "light" ? "dark" : "light");
  return (
    <..... value={{ theme, toggleTheme }}>
      {children}
    </.....>
  );
}

export default ThemeContext;

```

Fichier Navbar.js

```

import React, { ..... } from "react";
import ThemeContext from "../ThemeContext";
function Navbar() {
  const { theme, toggleTheme } = ..... (ThemeContext);

  return (
    <nav className={theme}>
      <h1>Mon Site</h1>
      <button onClick={..... }>Changer le thème</button>
    </nav>
  );
}

export default Navbar;

```

Question5 : Créer un chronomètre qui démarre automatiquement et s'incrémente chaque seconde.

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

function Timer() {
  const [seconds, setSeconds] = ..... (0);

  .....(() => {
    const interval = setInterval(() => {
      setSeconds(prev => prev + 1);
    }, 1000);

    .....() => clearInterval(interval);
  }, []);

  return <h2>Temps écoulé : {seconds}s</h2>;
}

export default Timer;
```

Exercice2 :

- On suppose que node js est déjà installé sur votre machine ,donner les lignes de commande pour créer une application react qui portera le nom suivant 'appreact'
 - Sans utiliser vite
 - En utilisant vite
- Donner les lignes commandes permettant de démarrer l'application dans les deux cas
- Donner la signification de chaque ordre suivant : `const [nb,setNb] =useState(0)`
 - `useEffect(() =>setNb(Stagiaires.length))`
 - `useEffect(()=>setNb(Stagiaires.length),[])`
 - `useEffect(()=>setNb(Stagiaires.length),[a,b])`

Exercice 3 :

On souhaite développer une application en ReactJS

On représente une classe de stagiaires sous forme d'un tableau ayant la structure suivante :

Stagiaires = [{id : 1, matricule : 1454 , nom : 'Alaoui' ,codepostal : 20400, ville : 'casa' ,moyenne : 12.56 }, {id : 2, matricule : 1455 , nom : 'Mansouri ,codepostal : 20400, ville : 'casa' ,moyenne : 14.67 }, , , {} , {} ,]

On souhaite créer un composant fonctionnel pour gérer toutes les fonctionnalités de l'interface ayant l'aperçu suivant :

1. Composant ListStagiaires.js

Filtrer
Rechercher

Id	Matricule	Nom	Ville	Code Postal	Moyenne	Actions
1	1454	Alaoui	Casa	20400	12.56	Supprimer Editer
2	1455	Mansouri	Casa	20400	14.67	Supprimer Editer

- La moyenne générale la plus élevée : 14.67
- La moyenne générale la moins élevée : 11.45
- La moyenne de la classe : 12.34

2. Composant FormStagiaire.js

Pour Ajouter et modifier un stagiaire après validation

3. Définir la liste des routes au niveau du composant app.js
4. Créer un composant Navbar.js
5. Tester votre application en utilisant Eslint
6. Mesurer la qualité de ton code en utilisant sonareqube