

Q1 - what is useCallback Hook, use Memo ??

1 - useCallback Hook .

■ **useCallback** هو ميزة مقدمة من React ، وهي مكتبة جافا سكريبت بنسخدمها عشان نبني واجهات المستخدم. بنستخدم ال الهوك لتحسين أداء المكونات الوظيفية عن طريق تخزين ال Function في الذاكرة المؤقتة . لما بننشئ fun جوه component بيتم إعادة إنشائها في كل عملية إعادة رسم، وده بيؤدي إلى إعادة رسم غير ضروري لل component الفرعية الي بتعتمد على ال function دي . باستخدام هوك **useCallback** ، بقينا بنخزن الدالة في الذاكرة المؤقتة بحيث تفضل ثابتة بين عمليات إعادة الرسم طول مافيش تغير في تبعياتها. وده بيساعد في تجنب إعادة الرسم غير الضرورية وتحسين أداء تطبيقك.

- مثال بسيط علي استخدام ال **useCallback**.

```
1 import React, { useCallback, useState } from 'react';
2
3 function MyComponent() {
4   const [count, setCount] = useState(0);
5
6   const handleClick = useCallback(() => {
7     setCount(count + 1);
8   }, [count]); // The array of dependencies triggers re-creation of the function if any of these dependencies change
9
10  return (
11    <div>
12      <p>Count: {count}</p>
13      <button onClick={handleClick}>Increment</button>
14    </div>
15  );
16 }
```

.useMemo - 2

■ useMemo هو عبارة عن هوك برضو بتقدمه React لتحسين الأداء. بنستخدمه لتخزين نتيجة حساب ما بحيث اننا مانعدش حسابه تاني مادام تبعياته ماتغيرتش. وده مفيد بشكل خاص لما بتعامل مع عمليات حسابية مكلفة أو حسابات ممكن تأثر على أداء ال component

- مثال بسيط علي استخدام ال useMemo.

```
1 import React, { useMemo } from 'react';
2
3 function MyComponent({ data }) {
4   const processedData = useMemo(() => {
5     // Some expensive data processing based on 'data'
6     return data.map(item => item * 2);
7   }, [data]); // Re-run the computation if 'data' changes
8
9   return (
10     <div>
11       <p>Processed Data: {processedData.join(', ')}</p>
12     </div>
13   );
```