**Front-end Questions**

1. **Explain the what's similar & difference between cookie / localStorage / sessionStorage.**

ทั้ง 3 อย่าง ใช้สำหรับจัดเก็บข้อมูลเหมือนกัน ใน Local Storage จะมีการเก็บข้อมูลไว้ได้นานและมากกว่า cookie หลายเท่า และข้อมูลที่เก็บจะยังคงอยู่ แม้ว่าจะปิด browser หรือลบ cookie ไปแล้วก็ตาม ส่วน session Storage เหมือนกัน Local Storage แทบทั้งหมด ต่างกันตรงที่ข้อมูลจะถูกลบทิ้งทุกครั้งที่เราปิด browser.

1. **Today React have hooks. Do we still need a class component? If your answer is yes then which case that we still need to use class component.**

ยังมีการใช้ class component อยู่ ใช้ตอนที่เราต้องการใช้คำสั่งเหล่านี้ในการวิเคราะห์ เพื่อให้ได้การ return ในสิ่งที่ต้องการ เช่น ComponentWillMount , ComponentWillUnMount

1. **Breifly describe how Virtual DOM works.**

เป็นการจำลอง DOM ขึ้นมาอีกชุดหนึ่ง โดยถ้าหากเราต้องการเปลี่ยนแปลงข้อมูลใน code ในตำแหน่งใดๆ ก็ตามบน DOM เราจะนำข้อมูลที่เปลี่ยนแปลงไปเทียบกับ Virtual DOM ก่อน หากเทียบกันแล้วข้อมูลไม่ต่างกัน แสดงว่าเราไม่จำเป็นต้องอัพเดท DOM ให้เสียเวลา แต่หากข้อมูลต่างกันจะมีการทำการเปลี่ยนแปลง DOM แบบเป็นชุด

1. **Consider this React's components tree**

Apps > ComponentA > ComponentB > ComponentC > ComponentD > ComponentE

If we have a state at Apps component, and ComponentE component want to access that state value. How do you implements this?

Import ไฟล์

1. **What different between using relative / absolute / fixed to position the element.**

**Relative :** ใช้กำหนดตำแหน่งที่อ้างอิงความสัมพันธ์กับตำแหน่งเดิมเช่น เรากำหนดให้ div1 เป็น left:20px; ผลลัพธ์คือ ตำแหน่ง div1 จะเลื่อนออกจากตำแหน่งเดิมทางซ้าย 20px

**Fixed :** ใช้กำหนดตำแหน่งของ element โดยอ้างอิงความสัมพันธ์กับตำแหน่งขอบด้านในของ browser และจะไม่เคลื่อนเมื่อมีการเลื่อน scrollbar เช่น เรากำหนดให้ div1 เป็น left:30px; ผลลัพธ์คือ ตำแหน่ง div1 จะเลื่อนออกจากด้านซ้ายของขอบด้านใน browser 30px

**Absolute :** เหมือนแบบ Fixed แต่ต่างตรงที่จะมีการเคลื่อนเมื่อมีการเลื่อน scrollbar

1. **Explain the different between using callback / Promise / async await. When to use and when not to**

**Callback :** callback คือ function ที่จะถูกเรียกหลัง function อื่นทำงานเสร็จ ใช้ในกรณีที่ไม่มีการเรียก callback ซ้อนกันหลายๆชั้น เนื่องจะทำให้เกิด callback hell

**Promise** : promise เป็น function ที่แก้ปัญหาในการเกิด callback hell โดยการทำงานจะมีทั้งหมด 3 สถานะ สถานะเริ่มแรกจะเป็น pending คือรอค่าส่งกลับมา (จาก background) เมื่อข้อมูลส่งกลับมา การทำงานอาจจะสำเร็จหรือไม่สำเร็จก็ได้ ซึ่งถ้าหากสำเร็จมันจะส่งกลับมาเป็น resolved ถ้าไม่สำเร็จจะส่งกลับมาในรูปแบบของ rejected

**Async await** : async เป็นการระบุว่า function นี้จะมีการทำงานแบบ asynchronous (ทำงานใน background) และคืนค่าในรูป promise await ใช้เพื่อบอกให้รอจนกว่าคำสั่งนั้นจะเสร็จ ถึงค่อยไปทำงานอันต่อไป โดย function ที่จะมี await อยู่ข้างในได้ต้องประกาศเป็น async เสมอ ควรใช้เมื่อมีการทำงานหลายๆงาน เพื่อ code จะได้สั้นและเป็นระเบียบ