**Frontend Question**

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

**ตอบ** Local Storage คือ การเก็บข้อมูลใส่ Local Storage แบบถาวร ข้อมูลจะหายไปก็ต่อเมื่อ clear Storage หรือกำหนดเวลาวันหมดอายุ

Session Storage คือ การเก็บข้อมูลใส่ Session Storage ซึ่งเมื่อปิดเว็บบราวเซอร์ข้อมูลก็จะหายไปหรือกำหนดระยะเวลาวันหมดอายุ

Local Storage เปรียบเทียบได้กับ Cookie ของบราวเซอร์ แต่ทำงานได้มีประสิทธิภาพสูงกว่ากันมาก

localStorage และ sessionStorage นั้นเหมือนกันแทบทุกประการ สิ่งเดียวที่ต่างกันก็คือ “ระยะเวลา” ที่จะเก็บข้อมูลเอาไว้

**2.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.**

**ตอบ** React Hooks ใช้ได้เฉพาะการเขียนในรูปแบบของ Function Components เท่านั้น (เรายังเขียนส่วนอื่นเป็นแบบ Class Components ได้อยู่ แค่จะใช้ React Hooks ไม่ได้ ถ้า Components ไหนจะใช้ ต้องเขียนในรูปแบบ Function Component เท่านั้น

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

**ตอบ** คือการจำหน้าจอไว้ในหน่วยความจำชั่วคราว (in-memory cache) เฃ่น พอ User กดปุ่ม หรือทำอะไรกับหน้าจอ ตัว React ก็จะเอาข้อมูลตัวนี้มาเทียบกับตัวที่เก็บไว้ในหน่วยความจำชั่วคราว แล้วเช็คดูว่า มีส่วนไหนที่ต้องวาดใหม่ไหม แล้วก็ทำการวาดใหม่แค่ “บางจุด” เท่านั้น

**4.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?

**ตอบ จ**ะสามารถส่งผ่าน props / state โดย props จะเป็นตัวส่ง ค่าต่างๆ รวมถึงฟังก์ชั่น ไปยัง Component อื่นๆ แต่ **state นั้นเสมือนเป็น data ที่มีการใช้แค่ภายใน Component นั้นๆ** ส่วนประกอบการทำงานภายในของ Components และก็เก็บข้อมูลและยังสามารเปลี่ยนแปลงระหว่างการทำงานได้อีกด้วย

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

**ตอบ position:relative** เป็นการกำหนดตำแหน่งที่อิงความสัมพันธ์กับตำแหน่งเดิม เช่น div1 กำหนด position เป็น relative และกำหนด left:20px; ผลลัพธ์คือ div1 จากเลื่อนออกจากตำแหน่งเดิมทางซ้าย 20 pixels  
ข้อมูลหรือเนื้อหาที่อยู่ใน element ที่กำหนด position เป็น relative สามารถที่จะเลื่อนและอยู่วางซ้อนเหนือ elements ตัวอื่นได้ แต่จะเกิดพื้นที่ว่างจากเนื้อหาหรือข้อมูลขึ้นในตำแหน่งเดิม **position:relative** จะใช้บ่อยสำหรับบรรจุบล็อกของ elements ที่กำหนด position เป็น absolute

**position:absolute**จะกำหนดตำแหน่งโดยอิงกับตำแหน่ง parent ตัวแรก ที่กำหนด position เป็นค่าอื่นที่ไม่ใช่ **static**

**position:absolute**สามารถที่จะซ้อนทับกันได้ โดยใช้ร่วมกับ z-index เพื่อกำหนดตำแหน่งบน ล่าง

**position:fixed** ใช้กำหนดตำแหน่งของ element โดยอิงความสัมพันธ์กับตำแหน่งขอบด้านในของบราวเซอร์ และจะไม่เคลื่อนเมื่อมีการเลื่อน scrollbar

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

**ตอบ**

การที่จะทำคำสั่ง Asynchronous ให้เสร็จก่อนแล้วค่อยทำคำสั่งถัดไป จะใช้ callback โดย callback จะถูกเรียกเมื่อคำสั่ง Asynchronous ทำงานเสร็จแล้ว โดยธรรมเนียมแล้วจะให้ตัวแปรแรกของ callback เป็น error object ถ้าเป็น null แปลว่าไม่มี error

การใช้ Callback ก่อให้เกิดปัญหาที่เรียกว่า Callback hell คือ callback ซ้อน callback เต็มไปหมด เลยเกิดวิธีแก้ปัญหาโดยการใช้ Promise

ตัวแปร Promise เป็นเหมือนกับคำสัญญาว่าจะทำคำสั่งนี้ให้ โดยจะมีอยู่ 3 สถานะก็คือ pending, resolved, rejected เมื่อเริ่มต้นทำคำสั่ง promise จะมีสถานะเป็น pending ถ้าทำเสร็จแล้วจะมีสถานะเป็น resolved โดยจะทำคำสั่งถัดไปที่อยู่ใน .then() และถ้าทำคำสั่งไม่สำเร็จ จะมีสถานะเป็น rejected และจะไม่ทำคำสั่งถัดไป แต่จะทำคำสั่งที่อยู่ใน .catch() แทน

**ข้อดีคือไม่มี callback hell แล้ว และเวลาจะ handle error เราก็ไม่ต้องทำที่ละจุดใน callback แล้ว**

แต่ Promise ก็ยังมีข้อเสียอยู่ตรงที่จะต้องเขียน .then() ทุกครั้งที่ต้องการทำคำสั่งถัดไป ดังนั้น async/await จึงเป็นภาคต่อของ Promise โดยการเขียน async ไว้หน้า function จะเป็นการระบุว่า function นั้นทำงานแบบ asynchronous

**โดยการเรียก await นั้นจะสามารถใช้กับ function ได้ 2 แบบก็คือ function ที่ return ค่า Promise object และอีกแบบหนึ่งคือ async function**

**สรุป async/await**

Code จะสะอาดกว่าใช้งาน Promise

การ Debug จะง่ายขึ้นเพราะจำนวน Callbacks จะลดลงไปเยอะมาก

การเปลี่ยนโค๊ดจาก Promise .then() หรือ .catch() ง่าย