Back-end Questions

1.

- สร้าง API Gateway ที่รวมข้อมูลจากทั้งสาม microservices โดยใช้ REST API ที่มีอยู่แล้ว เพื่อดึงข้อมูลมารวมและส่งต่อให้กับ client
- Caching ใช้ Redis เพื่อลดภาระในการดึงข้อมูลที่เปลี่ยนแปลงไม่บ่อย
- ใช้ WebSocket เพื่ออัปเดตข้อมูลให้ client แบบ push-based แทนการ polling ซ้ำ ๆ
- เพิ่มประสิทธิภาพสำหรับการดึงข้อมูลจาก RDBMS โดยใช้ index
- ใช้ Circuit Breaker เพื่อ ช่วยป้องกันระบบจากการเรียกใช้งานบริการที่ล้มเหลวซ้ำๆ และลดผลกระทบของความล้มเหลวที่อาจกระจายไปยังส่วนอื่นของระบบ

2.

- การกำหนดขอบเขตและเป้าหมาย
    - ความสามารถรองรับผู้ใช้สูงสุด
    - API ต้องตอบสนองภายในเวลาที่กำหนด , Timeout ไม่ควรเกินกี่วินาที
    - ระบบต้องรองรับ สามารถรับ transactions สูงสุดเท่าไหร่
    - Resource Usage กำหนด CPU Usage ไม่เกินเท่าไหร่ , Memory Usage ไม่เกินเท่าไหร่ , Disk I/O ไม่เกินเท่าไหร่
- ประเภทการทดสอบ
    - ทดสอบพฤติกรรมระบบภายใต้โหลดปกติ
    - ทดสอบขีดจำกัดของระบบ
    - ทดสอบเสถียรภาพระยะยาว
    - จำลองการเพิ่มขึ้นของโหลดอย่างฉับพลัน

- การเตรียมการทดสอบ
    - สภาพแวดล้อมการทดสอบ จำลองสภาพแวดล้อมให้ใกล้เคียง production
    - สร้างข้อมูลจำลองที่สมจริง ครอบคลุมทุกกรณีการใช้งาน
- การดำเนินการทดสอบ
    - Unit Performance Testing
    - Integration Performance Testing
    - End-to-End Performance Testing
    - บันทึก logs อย่างละเอียด
- การวิเคราะห์และรายงานผล
    - เปรียบเทียบผลกับเป้าหมาย
    - วิเคราะห์ระบบที่ต้องปรับปรุงและแนวทางการแก้ไขปัญหา
    - การรายงานผล
- แผนรับมือปัญหา
    - ทีมรับผิดชอบ
    - ขั้นตอนการแก้ไข
    - แผน Rollback

React Questions

1.

- ใช้สำหรับการสร้างฟังก์ชันที่มีการเก็บค่าไว้ในหน่วยความจำ โดยการ memoize ฟังก์ชัน เพื่อป้องกันการ re-render ที่ไม่จำเป็น