



ฝ่ายพัฒนาระบบสารสนเทศ
องค์การบริหารสโมสรนิสิตจุฬาลงกรณ์มหาวิทยาลัย (อบจ.)

แบบทดสอบทักษะพื้นฐาน สำหรับผู้สมัครตำแหน่ง

Frontend Developer

ฝ่ายพัฒนาระบบสารสนเทศ

องค์การบริหารสโมสรนิสิตจุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2565

คำชี้แจง

1. แบบทดสอบนี้ถูกออกแบบมาให้ทำเสร็จภายใน **120 ชั่วโมง (5 วัน)** อย่างไรก็ตามคณะกรรมการฯ จะให้เวลาในการทำทั้งหมด **168 ชั่วโมง (7 วัน)** หลังจากที่ผู้สมัครได้รับบททดสอบนี้ เนื่องจากผู้สมัครบางท่านอาจจะมีช่วงเวลาที่สะดวกแตกต่างกันไป
2. ให้ผู้สมัครทำการ fork repository [isd-sgcu/sgcu65-frontend-recruitment](https://github.com/isd-sgcu/sgcu65-frontend-recruitment) โดยใช้ account ของผู้สมัครเอง
3. กรุณาอ่านรายละเอียดโจทย์และทำความเข้าใจให้เรียบร้อยก่อนทำโจทย์ หากมีข้อสงสัยประการใดผู้สมัครสามารถส่งอีเมลไปที่ isd.team.sgcu@gmail.com เพื่อสอบถามได้ตลอดเวลา
4. พึงระลึกไว้เสมอว่าสิ่งเหล่านี้มีความสำคัญในการพัฒนาซอฟต์แวร์จริง ๆ ซึ่งมีผลต่อการพิจารณาผู้สมัคร
 - Code Readability – โค้ดอ่านรู้เรื่อง กระชับ เข้าใจง่าย
 - Commit Message Readability – ข้อความใน commit ต้องมีความหมาย
5. การส่ง assignment ขอให้ผู้สมัครอัปโหลด source code ทั้งหมดลง repository ดังกล่าว โดยหากต้องการอธิบายเกี่ยวกับโค้ดที่ส่งมาเพิ่มเติมสามารถเขียนลงในไฟล์ README.md ได้
6. หากมีไอเดียเพิ่มเติมจากที่โจทย์ระบุไว้ที่ดีกว่าจากที่โจทย์กำหนด ผู้สมัครสามารถระบุไอดีนั้นลงในไฟล์ README.md โดยเขียนด้วยรูปแบบดังนี้
 - # Additional Idea
 - <ไอดีเพิ่มเติมของผู้สมัคร>
7. หากมีสมมติฐานเพิ่มเติมที่โจทย์ไม่ได้ให้ข้อมูลไว้ ผู้สมัครสามารถระบุสมมติฐานนั้นลงในไฟล์ README.md โดยเขียนด้วยรูปแบบดังนี้
 - # Assumption
 - <สมมติฐานของผู้สมัคร>
8. การตรวจ assignment จะยึดจาก **commit สุดท้ายใน branch main** ก่อนเวลาสิ้นสุดแบบทดสอบที่ระบุไว้ในอีเมลที่ผู้สมัครได้รับเท่านั้น โปรดพึงระลึกไว้เสมอว่า การตรงต่อเวลา เป็นคุณสมบัติที่สำคัญอย่างหนึ่งต่อการเป็นวิศวกรซอฟต์แวร์ที่ดี
9. ขอให้ทยอยส่ง source code เมื่อทำเสร็จแต่ละ Feature เพื่อที่จะได้ไม่มีปัญหาการส่งเมื่อเวลาใกล้หมด **ไม่จำเป็นต้องเสร็จทั้งหมดแล้วค่อยส่ง**
10. ผู้สมัครสามารถค้นหาข้อมูลเพิ่มเติมจากช่องทางใดก็ได้ ในระหว่างการทำแบบทดสอบ
11. การตัดสินใจของคณะกรรมการคัดเลือกถือเป็นที่สุด

มหากภัยพิบัติวันสิ้นโลก

จุดประสงค์

- เพื่อทดสอบความรู้ HTML, CSS และ JavaScript เบื้องต้น
- เพื่อประเมินความสามารถในการแก้ปัญหาของผู้สมัคร
- เพื่อประเมินความคิดสร้างสรรค์ของผู้สมัคร

รายละเอียดโจทย์

วันหนึ่ง ดาวเคราะห์ดวงที่สามของระบบสุริยะทางช้างเผือกหรืออีกชื่อหนึ่ง "โลก" ได้ล่มสลายไปจากมหากภัยพิบัติทางธรรมชาติที่ใช้เวลาก่อร่างสร้างตัวกว่า 2.745 ล้านปีจนทำให้สิ่งมีชีวิตเกือบทั้งหมดในโลกมลายหายไปล่าสุด คุณและเหล่าผองเพื่อนเป็นผู้รอดจากมหากภัยพิบัตินั้น ซึ่งคุณต้องการที่จะกอบกู้โลกใบเดิมให้กลับมาให้เร็วที่สุด แต่ทั้งนี้ทั้งนั้นคุณได้พบว่าโลกใบเดิมของคุณมีระบบจัดการฐานข้อมูลที่ไม่ดีเท่าที่ควร ระบบราชการและสวัสดิการของบางประเทศมีความจำเป็นที่จะต้องกรอกข้อมูลซ้ำ ๆ ทั้ง ๆ ที่มนุษย์สมัยก่อนทุกคนมีสิ่งที่เอาไว้ยืนยันตัวตนเช่น บัตรประชาชน ที่ข้อมูลพื้นฐานทุกอย่างเช่น ชื่อ นามสกุล อยู่ในนั้นทั้งหมด

คุณผู้เป็นผู้รอดชีวิตจากภัยพิบัติและเห็นถึงปัญหานั้นจึงได้เริ่มทำ "ระบบลงทะเบียน" ที่จะให้มนุษย์ที่เหลือรอดทุกคนลงทะเบียนและมอบสวัสดิการที่ดีให้อย่างทั่วถึง ซึ่งคุณได้รวบรวมเหล่าเพื่อนพ้องที่ต้องการจะแก้ปัญหาขึ้นมาได้จำนวนหนึ่งแล้ว ซึ่งเพื่อน ๆ ของคุณก็ทำหน้าที่ต่าง ๆ กันไป เพื่อนคนแรกได้ออกแบบ [UX/UI มาให้คุณแล้วเรียบร้อยแล้ว](#) คุณสามารถเข้าไปดูและพัฒนาเว็บให้มีหน้าตาตามนั้นได้เลย เพื่อนคนที่สองได้ทำการพัฒนาระบบ [Backend มาให้คุณแล้วเรียบร้อยแล้ว](#) คุณสามารถเข้าไปดูวิธีการใช้งานและทดลองใช้ API ได้

หน้าที่ของคุณคือ "พัฒนาเว็บลงทะเบียน" ที่จะมอบประสบการณ์ในการใช้งานที่ดีให้กับผู้ใช้ มีความสวยงามรองรับได้ทั้งโทรศัพท์และบนคอมพิวเตอร์ และทำงานได้อย่างถูกต้องตามที่ควรจะเป็น โดยคุณต้องทำการพัฒนาส่วน Frontend ร่วมกับเพื่อนอีกคนหนึ่งแต่เพื่อนคนนั้นดันลาป่วยยาว คุณได้ตัดสินใจที่จะพัฒนาเว็บไซต์ให้ได้เยอะที่สุดเท่าที่จะทำได้ระหว่างที่เพื่อนของคุณพักฟื้น แต่เนื่องจากเพื่อนของคุณเก่งมาก feature ไหนที่คุณคิดว่าทำไม่ไหว (หรือขี้เกียจทำ) คุณก็สามารถเก็บไว้ให้เพื่อนคุณทำต่อได้ (mark feature ที่ทำแล้วใน README.md ด้วย!) อย่าลืมที่จะเขียน Code ให้อ่านง่าย และเขียน Commit Message ให้มีความหมาย คุณจะได้อะไรตอบแทนเพื่อนของคุณกลับมาทำงาน

บางครั้งงานออกแบบที่เพื่อนคุณทำนั้นก็ได้สมบูรณ์แบบเสมอไป คุณสามารถปรับแต่งอะไรได้นิดหน่อยตามความเหมาะสมตามที่เห็นสมควร แต่บอกเพื่อนของคุณด้วยว่าคุณแก้อะไรไปและบอกเหตุผลด้วยว่าทำไม (ในที่นี้คืออย่าลืมเขียน Additional Idea ด้วยยย)

Features - หน้าลงทะเบียน

1. เว็บไซต์มีหน้าตาเหมือนกับ UX/UI ที่กำหนดไว้
2. เว็บไซต์สามารถ Navigate ผู้ใช้ไปยัง Social Media อื่น ๆ ได้เช่น Facebook, Instagram และ Twitter
3. เว็บไซต์สามารถ Responsive ได้ (เว็บใน Mobile และ Desktop ต้องไม่ละ)

4. เว็บไซต์มีระบบ Form Validation
 - a. ทุก field ต้องไม่ใช่ string เปล่า
 - b. ชื่อผู้ใช้ต้องประกอบด้วยตัวอักษรภาษาอังกฤษหรือตัวเลขเท่านั้น
 - c. อีเมลต้องถูกรูปแบบ
 - d. รหัสผ่านกับยืนยันรหัสผ่านตรงกัน
5. เมื่อกดปุ่ม Submit แล้ว เว็บไซต์จะสามารถเรียก API เพื่อส่งข้อมูลของผู้ใช้ไปให้ Backend
6. เว็บไซต์สามารถแสดง error จาก API response ได้

Features - หน้าที่รายชื่อผู้ลงทะเบียน

1. เว็บไซต์มีหน้าตาเหมือนกับ UX/UI ที่กำหนดไว้
2. เว็บไซต์สามารถ Responsive ได้ (เว็บใน Mobile และ Desktop ต้องไม่ละ!)
3. เว็บไซต์สามารถดึงข้อมูลผู้ใช้จาก API มาแสดงได้
4. User สามารถพิมพ์ค้นหาผู้ใช้ได้จาก
 - a. ชื่อ
 - b. นามสกุล
 - c. ชื่อผู้ใช้

Tips

1. คุณสามารถค้นหาข้อมูลเพิ่มจากแหล่งความรู้ไหนก็ได้ไม่จำกัด ยกเว้นถามเพื่อน ๆ ทั้งนี้ เราคาดหวังว่าคุณจะสามารถแก้ไขปัญหาได้ด้วยตัวเอง
2. คุณไม่จำเป็นต้องพัฒนาเว็บไซต์ให้ครบทุก Feature โดยสามารถเลือกทำตามความสำคัญ ตามความยากง่าย หรือตามความถนัดของคุณก็ได้
3. คุณสามารถหาไฟล์รูปและ icon ต่าง ๆ ที่ถูกใช้ใน design ได้จาก folder assets
4. หากคุณใช้ VS Code ในการพัฒนา แนะนำว่าติดตั้ง [VS Code Extension: Live Server](#) เพื่อความง่ายในการพัฒนาเว็บไซต์
 - ก่อนอื่นทำการ Install Extension มาก่อนตามลิงก์ที่แปะไว้ข้างบนเลย
 - เมื่อ Install เสร็จแล้ว ให้คลิกขวาที่ไฟล์ `index.html` แล้วจะมี Menu หนึ่งมาว่า **Open with Live Server**
 - ซึ่ง Live Server จะรันอยู่ที่ <http://localhost:5500> ถ้าเปิดแล้วได้หน้าตาคล้าย ๆ folder ไม่ต้องตกใจให้คุณกดเลือก name-list หรือ register หน้าเว็บก็จะแสดงออกมา
 - ทุกครั้งที่คุณ save file, Live Server จะทำการ Rerender หน้าเว็บใหม่ให้ ดังนั้นคุณไม่ได้ต้องเปิด ๆ ปิด ๆ ระหว่างทำงานอีกต่อไป
5. คุณไม่จำเป็นต้องใช้ framework ใด ๆ ในการทำข้อนี้ แต่หากต้องการใช้ก็สามารถใช้ได้

6. ใน repository ที่คุณทำการ fork ไป ทั้งสองหน้าจะถูกแยกไว้คนละ folder กัน แต่คุณสามารถลบทั้งสอง folder ทั้งและเขียนรวมกันได้โดยใช้ routing ในการแสดงแต่ละหน้า
 - ให้แสดงหน้ารายชื่อผู้ลงทะเบียนที่ route “/”
 - ให้แสดงหน้าลงทะเบียนที่ route “/register”
7. หากคุณอยากเพิ่มอะไรมากกว่าที่กำหนดก็สามารถทำได้เลย!

การส่งงาน

1. ขอให้อัปโหลด source code ทั้งหมด (ยกเว้น dependencies ที่มีขนาดใหญ่ เช่น node_modules ถ้ามี) หากไม่ได้เขียนด้วย pure HTML, CSS, JavaScript ขอให้ระบุวิธีการเปิดเว็บขึ้นมาดูในไฟล์ README.md ด้วย โดยเขียนด้วยรูปแบบดังนี้

How To Run

<วิธีการเปิดเว็บ>