

## บทที่ 3

### เทคโนโลยีที่ใช้ในการศึกษา

ในการดำเนินโครงการ “การทดสอบระบบบริหารจัดการงานสนับสนุนภายในองค์กรด้วยการทดสอบอัตโนมัติ” ของบริษัท ลูคัส สแตรททิจี จำกัด นั้นเป็นการทดสอบระบบเว็บแอปพลิเคชัน ผู้จัดทำโครงการจำเป็นต้องศึกษาทำความเข้าใจและเลือกใช้เทคโนโลยีต่าง ๆ ที่เกี่ยวข้องกับการพัฒนาและการทดสอบระบบ ดังนี้

#### 3.1 การวิเคราะห์และออกแบบระบบ (System Analysis and Design)

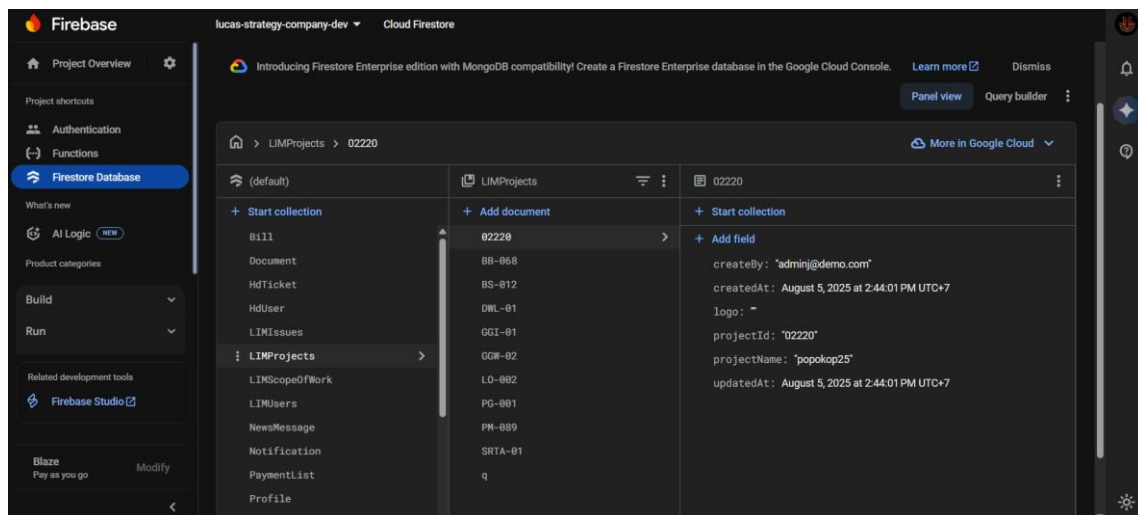
Figma เป็นเครื่องมือสำหรับออกแบบและสร้างแบบจำลองหน้าจอของเว็บแอปพลิเคชันและแอปพลิเคชันบนมือถือที่ได้รับความนิยมในกลุ่ม UX/UI Designer ทั่วโลก ที่นิยมในระดับสากล โดยสามารถใช้ออกแบบโครงสร้างหน้าจอของเว็บแอปพลิเคชันและแอปพลิเคชันบนอุปกรณ์เคลื่อนที่ได้ อย่างมีประสิทธิภาพ ในบทบาทของ QA/Tester การเข้าใจแบบจำลองที่ถูกออกแบบผ่าน Figma มีความสำคัญอย่างยิ่ง เนื่องจากช่วยให้ผู้ทดสอบสามารถวิเคราะห์โครงสร้างระบบ ฟังก์ชันการทำงาน และลำดับกระบวนการทำงานภายใน และความสัมพันธ์ของข้อมูลได้อย่างชัดเจน การมีความเข้าใจในภาพรวมเช่นนี้ทำให้ ผู้ทดสอบสามารถวางแผนการทดสอบออกแบบกรณีทดสอบ (Test Cases) รวมถึงเตรียมข้อมูลทดสอบที่เหมาะสมสำหรับการทดสอบในทุกกรณี และกำหนดข้อมูลทดสอบได้อย่างถูกต้องและครอบคลุม



รูปที่ 3.1 เครื่องมือออกแบบการแสดงผล Figma

### 3.2 ระบบจัดการฐานข้อมูล (Database System)

Firebase เป็นแพลตฟอร์มคลาวด์ที่มีความยืดหยุ่นสูงและรองรับการจัดการฐานข้อมูลที่หลากหลาย โดยเฉพาะฐานข้อมูลแบบ NoSQL ที่ใช้โครงสร้างข้อมูลแบบ Document-based ประกอบด้วย Collection และ Document ช่วยให้การจัดเก็บและเข้าถึงข้อมูลเป็นไปอย่างรวดเร็วและมีประสิทธิภาพ เหมาะสมกับระบบที่ต้องการความทันสมัยและการประมวลผลแบบเรียลไทม์ สำหรับ QA/Tester การเข้าใจโครงสร้างและรูปแบบข้อมูลใน Firebase มีความสำคัญ เนื่องจากต้องสามารถเข้าถึง ตรวจสอบ และควบคุมข้อมูลที่เกี่ยวข้องกับการทดสอบได้อย่างถูกต้องและแม่นยำ Tester สามารถใช้ Firebase Console หรือ API ในการเตรียมข้อมูลทดสอบ (Test Data Setup) ได้ล่วงหน้า เช่น การเพิ่มข้อมูลสำหรับกรณีทดสอบพิเศษ การลบข้อมูลที่ซ้ำซ้อน หรือการจำลองสถานะของระบบเพื่อทดสอบความผิดปกติ (Negative Testing) นอกจากนี้ QA ยังต้องตรวจสอบความถูกต้องของข้อมูลหลังจากดำเนินการทดสอบ เพื่อยืนยันว่าระบบสามารถจัดเก็บและประมวลผลข้อมูลได้ตรงตามข้อกำหนดและความคาดหวัง (Expected Result) ซึ่งช่วยให้การทดสอบระบบมีความน่าเชื่อถือและครอบคลุมทุกกรณี

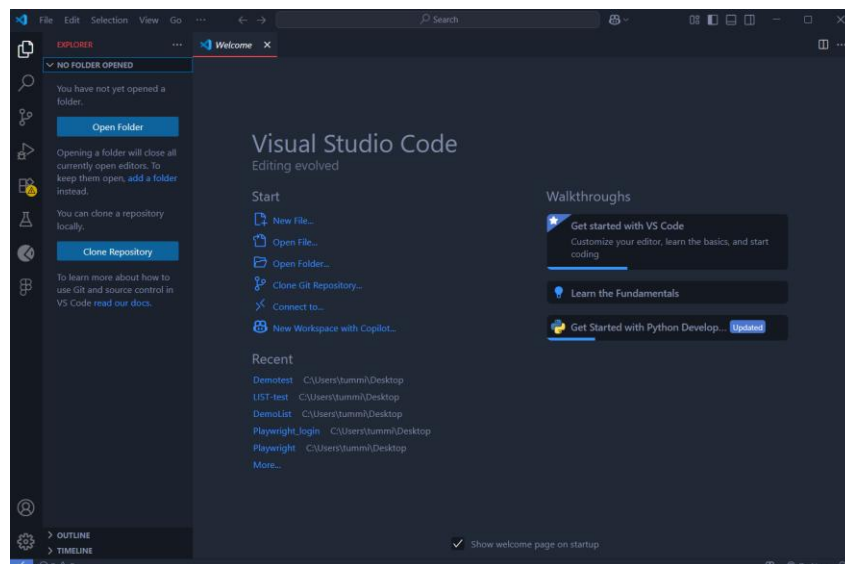


รูปที่ 3.2 เช็คนำฐานข้อมูลของ Firebase

### 3.3 ซอฟต์แวร์ภาษาและเครื่องมือที่ใช้ในการพัฒนาโครงการ

#### 3.3.1 Visual Studio

Visual Studio Code เป็นโปรแกรมแก้ไขโค้ด (Code Editor) ที่พัฒนาโดย Microsoft มีความโดดเด่นด้วยความเร็ว ความเสถียร และการรองรับส่วนขยาย (Extensions) จำนวนมาก ทำให้เหมาะสำหรับการเขียนและพัฒนาซอฟต์แวร์ในหลากหลายภาษา รวมถึง JavaScript และ TypeScript ที่ใช้ในโครงงานนี้ VS Code ยังมีฟีเจอร์สำคัญ เช่น การเติมโค้ดอัตโนมัติ (IntelliSense), ระบบ Debugger ที่ช่วยตรวจสอบและแก้ไขข้อผิดพลาด รวมถึงการจัดการเวอร์ชันด้วย Git Integration ซึ่งช่วยให้การพัฒนาและทดสอบโค้ดเป็นไปอย่างมีประสิทธิภาพ



รูปที่ 3.3 หน้าโปรแกรม Visual Studio Code

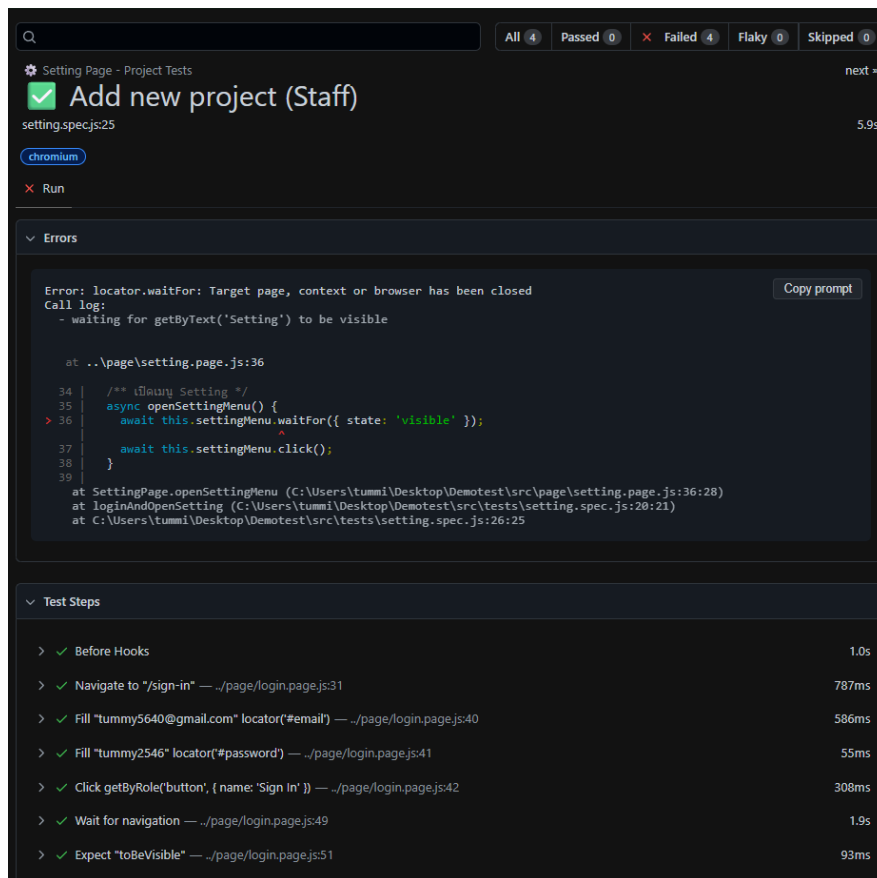
#### 3.3.2 ภาษาและ framework ที่ใช้

1) JavaScript เป็นภาษาหลักสำหรับพัฒนาเว็บแอปพลิเคชันและเขียนสคริปต์ทดสอบในโครงงานนี้ ด้วยความสามารถในการทำงานทั้งฝั่ง Client และ Server ผ่าน Node.js ช่วยให้สร้างชุดทดสอบแบบอัตโนมัติได้อย่างมีประสิทธิภาพ รองรับการทำงานแบบ asynchronous ซึ่งสำคัญสำหรับการจัดการเหตุการณ์บนเว็บ เช่น การโหลดหน้าเว็บ การคลิกปุ่ม และการกรอกข้อมูลแบบไดนามิก

2) Playwright เป็นเครื่องมือสำหรับการทดสอบระบบแบบ End-to-End ที่พัฒนาโดย Microsoft รองรับการทดสอบบนหลายเบราว์เซอร์ ได้แก่ Chromium, Firefox และ WebKit (Safari) จุดเด่นของ Playwright คือสามารถจำลองพฤติกรรมผู้ใช้ เช่น การคลิก การกรอกข้อมูล และการนำทางผ่านหน้าเว็บได้อย่างแม่นยำ นอกจากนี้ยังรองรับฟีเจอร์ขั้นสูง เช่น การจับภาพหน้าจอ การบันทึกวิดีโอ และการทดสอบแบบขนาน (Parallel Testing) ที่ช่วยลดระยะเวลาในการรันชุดทดสอบ อีกทั้ง Playwright ยังออกแบบ API ที่เข้าใจง่ายและใช้งานสะดวก ทำให้เหมาะกับทีม QA ที่ต้องการสร้างชุดทดสอบที่แข็งแกร่งและยืดหยุ่น

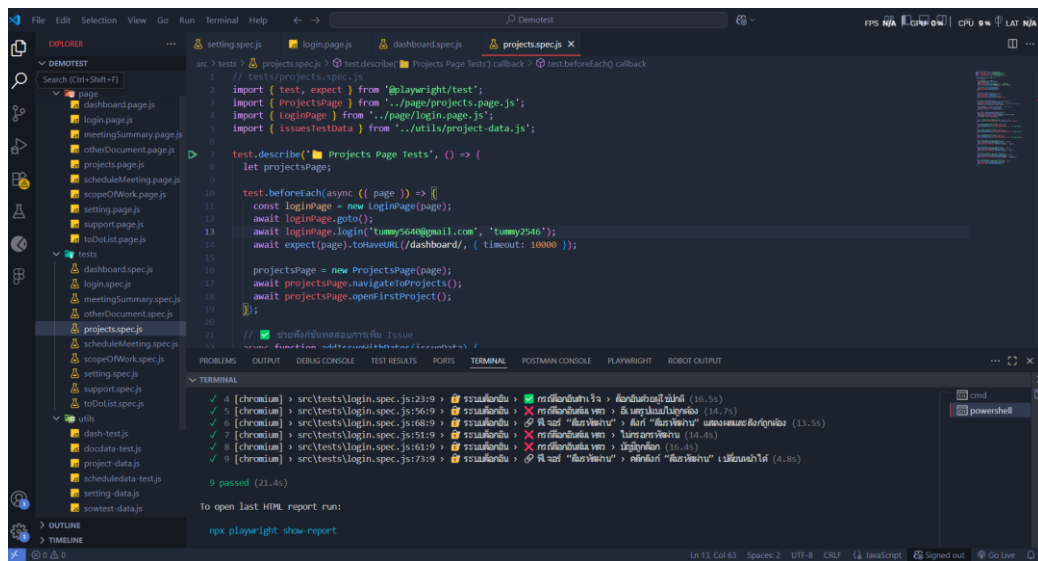
Project: chromium	8/8/2025, 1:17:11 PM	Total time: 21.4s
<div> <div>🔍</div> <div> All 9 Passed 9 Failed 0 Flaky 0 Skipped 0 </div> </div>		
login.spec.js		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; ✓ กรณีล็อกอินสำเร็จ &gt; ล็อกอินด้วยผู้ใช้ปกติ</div> <div>login.spec.js:23</div> <div>16.5s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; ✗ กรณีล็อกอินล้มเหลว &gt; กรอกรหัสผิด</div> <div>login.spec.js:36</div> <div>15.4s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; ✗ กรณีล็อกอินล้มเหลว &gt; กรอกรหัสผ่านผิด</div> <div>login.spec.js:41</div> <div>16.3s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; ✗ กรณีล็อกอินล้มเหลว &gt; ไม่กรอกรหัส</div> <div>login.spec.js:46</div> <div>17.0s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; ✗ กรณีล็อกอินล้มเหลว &gt; ไม่กรอกรหัสผ่าน</div> <div>login.spec.js:51</div> <div>14.4s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; ✗ กรณีล็อกอินล้มเหลว &gt; อีเมลรูปแบบไม่ถูกต้อง</div> <div>login.spec.js:56</div> <div>14.7s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; ✗ กรณีล็อกอินล้มเหลว &gt; บัญชีถูกล็อก</div> <div>login.spec.js:61</div> <div>16.4s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; 🔄 ฟีเจอร์ "ลิมิตผ่าน" &gt; ลิงก์ "ลิมิตผ่าน" แสดงผลและลิงก์ถูกต้อง</div> <div>login.spec.js:68</div> <div>13.5s</div> </div>		
<div> <div>✓</div> <div>🔒 ระบบล็อกอิน &gt; 🔄 ฟีเจอร์ "ลิมิตผ่าน" &gt; คลิกลิงก์ "ลิมิตผ่าน" เปลี่ยนหน้าได้</div> <div>login.spec.js:73</div> <div>4.8s</div> </div>		

รูปที่ 3.4 หน้าแสดงผลการทดสอบระบบของ Playwright เมื่อผ่านการทดสอบ



รูปที่ 3.5 หน้าแสดงผลการทดสอบระบบของ Playwright เมื่อเกิดข้อผิดพลาด

3) Page Object Model คือรูปแบบการออกแบบชุดทดสอบที่แยกส่วนความรับผิดชอบระหว่างส่วนของหน้าเว็บ (Page Object) กับตรรกะการทดสอบ (Test Logic) โดยแต่ละหน้าจะถูกแทนด้วยคลาสหรือโมดูลที่ประกอบด้วยตัวแปรองค์ประกอบ (Elements) และเมธอดการกระทำ (Actions) เช่น การคลิกปุ่ม การกรอกข้อมูล และการตรวจสอบข้อความ การใช้ POM ช่วยเพิ่มความสามารถในการดูแลรักษา (Maintainability) ลดความซ้ำซ้อน และทำให้การแก้ไขโค้ดเมื่อ UI เปลี่ยนแปลงเป็นไปได้ง่ายขึ้น



รูปที่ 3.6 การจัดทำ Page Object Model

### 3.3.3 ซอฟต์แวร์สื่อสาร

- 1) Microsoft Teams คือแพลตฟอร์มสำหรับการสื่อสารและประชุมออนไลน์ เหมาะกับองค์กรที่ใช้ระบบ Microsoft 365 มีฟีเจอร์ครบถ้วน เช่น การแชท วิดีโอคอล การแชร์ไฟล์ และการจัดการงานที่เชื่อมต่อกับ Outlook และ OneDrive ช่วยให้ทีม QA ทำงานร่วมกันได้อย่างมีประสิทธิภาพ
- 2) Discord เป็นแพลตฟอร์มสื่อสารที่เน้นการสนทนาแบบเรียลไทม์ผ่านข้อความและเสียง เหมาะสำหรับทีมขนาดเล็กหรือทีมที่ต้องการความคล่องตัวสูงในการประสานงาน
- 3) GitHub แพลตฟอร์มสำหรับควบคุมเวอร์ชันและจัดเก็บซอร์สโค้ดที่ใช้ในการพัฒนาและทดสอบ โดยในโครงการนี้ GitHub ถูกใช้สำหรับการจัดเก็บสคริปต์การทดสอบ ทำงานร่วมกันในทีม QA และใช้ในการติดตามข้อผิดพลาด (Issues), Review Code และบันทึกประวัติการพัฒนา

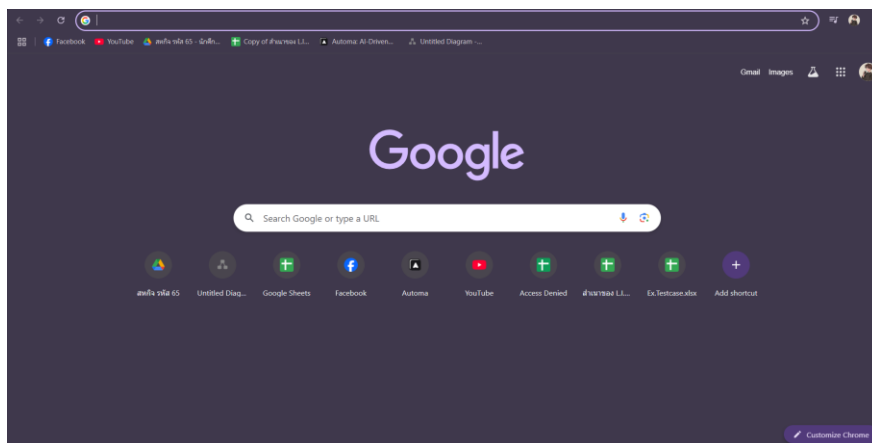
### 3.3.4 ซอฟต์แวร์อื่นๆ

1) Google Sheets เป็นโปรแกรมสเปรดชีตออนไลน์ที่พัฒนาโดย Google สามารถใช้งานผ่านเว็บเบราว์เซอร์และรองรับการทำงานแบบเรียลไทม์ ช่วยให้ผู้ใช้หลายคนสามารถแก้ไขและดูข้อมูลในไฟล์เดียวกันได้พร้อมกัน ในกระบวนการทดสอบซอฟต์แวร์ Google Sheets ถูกใช้เป็นเครื่องมือหลักในการจัดทำแผนการทดสอบ (Test Plan) และออกแบบกรณีทดสอบ (Test Case) โดยสามารถจัดเก็บรายละเอียดกรณีทดสอบ ขั้นตอนการทดสอบ ผลลัพธ์ที่คาดหวัง และผลการทดสอบได้อย่างเป็นระบบ อีกทั้งยังสามารถแชร์และควบคุมสิทธิ์การเข้าถึงข้อมูลได้อย่างมีประสิทธิภาพ ช่วยให้ทีม QA และผู้เกี่ยวข้องติดตามและอัปเดตข้อมูลได้อย่างรวดเร็ว ส่งผลให้การจัดการงานทดสอบมีความโปร่งใสและมีประสิทธิภาพ

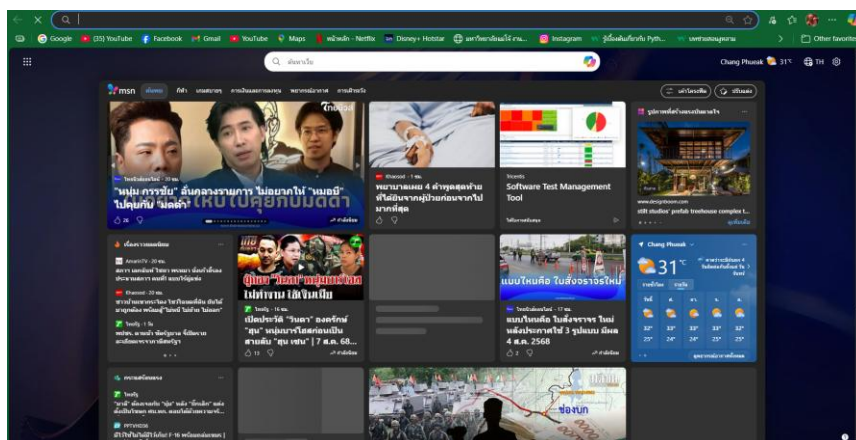
Dashboard								
Test case ID	Test Scenario	Test Case Description	Test Step	Expected Result	Test Result	Remark Defect Severity*	Testing date	Tester
LIST-001	แสดงภาพรวมทั้งหมด		1. เปิดหน้า Login 2. เข้าหน้า Dashboard	แสดงภาพรวมทั้งหมด	Passed			
Project								
Test case ID	Test Scenario	Test Case Description	Test Step	Expected Result	Test Result	Remark Defect Severity*	Testing date	Tester
LIST-001	แสดงรายการโปรเจกต์ทั้งหมด		1. เข้าสู่ระบบ 2. ไปที่เมนู Projects	แสดงรายการโปรเจกต์ทั้งหมดที่มีในระบบอย่างครบถ้วน พร้อมข้อมูลหลักของแต่ละโปรเจกต์	Passed			
LIST-002	เลือกโปรเจกต์เพื่อดูรายละเอียด		1. ที่หน้า Projects 2. คลิกเลือกโปรเจกต์ใดโปรเจกต์หนึ่ง	แสดงรายละเอียดของโปรเจกต์ที่เลือก เช่น ProjectID, ProjectName, Scope, Start Date, Due Date ฯลฯ	Passed			เลือกโปรเจกต์ตัวอย่าง เช่น Project A
LIST-003	เพิ่ม Issue ลงในโปรเจกต์		1. เลือกโปรเจกต์ที่หน้า Projects 2. คลิกเพิ่ม Issue ใหม่ 3. กรอกข้อมูล Issue ครบถ้วน 4. กดบันทึก	Issue ถูกเพิ่มในโปรเจกต์นั้น และแสดงในรายการ Issue ของโปรเจกต์	Passed			Issue Code: 1001 Title: Bug Fix Description: Fix login error
LIST-004	แสดงรายการ Issue ของโปรเจกต์		1. เลือกโปรเจกต์ใดโปรเจกต์หนึ่ง 2. ดูรายการ Issue ที่เกี่ยวข้อง	แสดง Issue ทั้งหมดที่ถูกเพิ่มในโปรเจกต์นั้น พร้อมรายละเอียด Issue Code, Title, Status ฯลฯ	Passed			
LIST-005	กดแก้ไข		1. เลือกโปรเจกต์ใดโปรเจกต์หนึ่ง 2. ดูรายการ Issue ที่เกี่ยวข้อง 3. แก้ไข 4. กดบันทึก		Failed			ตรวจสอบสถานะ: Active, Closed หรือวันที่ Due Date
LIST-006	กดทำสำเนา		1. เลือกโปรเจกต์ใดโปรเจกต์หนึ่ง 2. ดูรายการ Issue ที่เกี่ยวข้อง 3. ทำสำเนา 4. กดบันทึก		Not Test			
LIST-007	กดลบรายการ		1. เลือกโปรเจกต์ใดโปรเจกต์หนึ่ง 2. ดูรายการ Issue ที่เกี่ยวข้อง 3. ลบ 4. กดบันทึก		Not Test			
Scope Of Work								
Test case ID	Test Scenario	Test Case Description	Test Step	Expected Result	Test Result	Remark Defect Severity*	Testing date	Tester
				แสดงรายการ Scope of Work ทั้งหมด พร้อมข้อมูล:				

รูปที่ 3.7 การจัดทำ Test Case และ Test Plan

2) Google Chrome และ Microsoft Edge เป็นเว็บเบราว์เซอร์ยอดนิยมที่ใช้สำหรับเปิดดูและทดสอบเว็บแอปพลิเคชัน ทั้งสองเบราว์เซอร์มีพื้นฐานจาก Chromium ซึ่งเป็นโปรเจกต์โอเพ่นซอร์สที่ช่วยให้มีประสิทธิภาพสูงและรองรับมาตรฐานเว็บสมัยใหม่ สำหรับการทดสอบระบบเว็บแอปพลิเคชัน Google Chrome และ Microsoft Edge เป็นเบราว์เซอร์หลักที่ใช้ในการรันชุดทดสอบอัตโนมัติ (Automated Testing) เนื่องจากเครื่องมืออย่าง Playwright รองรับการทำงานแบบข้ามเบราว์เซอร์ (Cross-Browser Testing) บนแพลตฟอร์มเหล่านี้ จึงช่วยยืนยันว่าระบบสามารถทำงานได้อย่างถูกต้องและมีเสถียรภาพในสภาพแวดล้อมที่หลากหลาย เพื่อให้ผู้ใช้ได้รับประสบการณ์ที่ดีไม่ว่าจะเลือกใช้งานเบราว์เซอร์ใด



รูปที่ 3.8 หน้า Google Chrome



รูปที่ 3.9 หน้า Microsoft Edge



3) **Includia Accessibility Checker** เป็นส่วนขยาย (Extension) สำหรับเบราว์เซอร์ Google Chrome ที่ใช้ตรวจสอบความเข้าถึงเว็บไซต์ (Accessibility Testing) ตามมาตรฐาน WCAG (Web Content Accessibility Guidelines) โดยช่วยสแกนหน้าเว็บ วิเคราะห์องค์ประกอบต่าง ๆ เช่น ข้อความแทนภาพ (alt text), ความคมชัดของสี (color contrast), โครงสร้างหัวเรื่อง (heading structure) และการตั้งค่าองค์ประกอบ HTML ที่ถูกต้อง พร้อมรายงานผลที่ชัดเจนและคำแนะนำในการแก้ไข ช่วยให้ทีมพัฒนาสามารถตรวจสอบและปรับปรุงระบบให้สอดคล้องกับมาตรฐานตั้งแต่ช่วงแรกของการพัฒนา

4) **WAVE Evaluation Tool** พัฒนาโดยองค์กร WebAIM เป็นเครื่องมือสำหรับวิเคราะห์และประเมินปัญหาด้าน Accessibility อย่างรวดเร็วและแม่นยำ ใช้งานได้ทั้งในรูปแบบเว็บไซต์ออนไลน์และส่วนขยายของเบราว์เซอร์ เช่น Chrome และ Firefox มีฟีเจอร์แสดงผลแบบซ้อนทับ (Visual Overlay) บนหน้าเว็บจริง ช่วยให้ผู้พัฒนาและผู้ทดสอบเห็นจุดปัญหาได้อย่างชัดเจน พร้อมแถบสรุปรายงานที่มีรายละเอียดข้อผิดพลาดและคำแนะนำ

5) **axe DevTools Browser Extension** คือส่วนขยายที่ติดตั้งบนเบราว์เซอร์ เช่น Chrome, Firefox หรือ Edge เพื่อช่วยนักพัฒนาและผู้ทดสอบตรวจสอบปัญหาเรื่อง Accessibility บนเว็บไซต์ได้โดยตรงผ่านเครื่องมือ Developer Tools ของเบราว์เซอร์ โดยใช้ engine ที่ชื่อว่า axe-core ซึ่งเป็น JavaScript library ที่ได้รับการพัฒนาตามมาตรฐาน WCAG 2.1 และมาตรฐานอื่น ๆ ที่เกี่ยวข้อง ช่วยให้สามารถตรวจจับปัญหาการเข้าถึงเว็บไซต์ได้อย่างรวดเร็วและแม่นยำ เช่น การตรวจสอบความแตกต่างของสี (Color Contrast) การตรวจสอบการมีอยู่ของข้อความบรรยายรูปภาพ (alt text) การตรวจสอบการตั้งค่า label ของฟอร์ม รวมถึงการจัดลำดับหัวข้อ (Heading Structure) ที่ถูกต้อง พร้อมทั้งแสดงคำแนะนำในการแก้ไขปัญหา ทำให้ง่ายต่อการนำไปปรับปรุงแก้ไข

6) **Accessibility Screen Reader** คือโปรแกรมที่แปลงข้อมูลบนหน้าจอคอมพิวเตอร์หรืออุปกรณ์มือถือให้อ่านออกเสียง (Text-to-Speech) หรือแสดงผลในรูปแบบอักษรเบรลล์ ซึ่งช่วยให้ผู้พิการทางสายตาสามารถรับรู้ข้อมูลและโต้ตอบกับอินเทอร์เฟซดิจิทัลได้อย่างเต็มประสิทธิภาพ โดย Screen Reader จะทำงานผ่านการสแกนโครงสร้างของหน้าเว็บหรือแอปพลิเคชัน (DOM - Document Object Model) เพื่ออ่านเนื้อหาและบ่งบอกบทบาทขององค์ประกอบ เช่น ปุ่ม ลิงก์ ฟอร์ม หัวข้อ รวมถึงแจ้งสถานะต่าง ๆ เช่น โฟกัสที่ตำแหน่งใด หรือปุ่มใดถูกเลือกอยู่ การแสดงผล

ของ Screen Reader มีสองรูปแบบหลัก ได้แก่ การแปลงข้อความเป็นเสียงพูด (Text-to-Speech) และการแสดงผลผ่านอุปกรณ์อักษรเบรลล์ที่สามารถอ่านด้วยปลายนิ้ว ซึ่งทั้งสองรูปแบบนี้ช่วยให้ผู้ใช้ที่มีข้อจำกัดทางสายตาสามารถเข้าถึงข้อมูลและใช้งานระบบได้อย่างมีประสิทธิภาพ