

## บทที่ 5

### การทดสอบระบบ

#### 5.1 การทดสอบระบบแบบ Automated Testing ด้วย Playwright

Playwright เป็นเครื่องมือสำหรับการทดสอบระบบอัตโนมัติ (Automated Testing Framework) ซึ่งพัฒนาโดยบริษัท Microsoft ออกแบบมาเพื่อทดสอบ เว็บแอปพลิเคชัน บนเบราว์เซอร์หลากหลายชนิด ได้แก่ Chromium (Google Chrome, Microsoft Edge), Firefox และ WebKit (Safari) ซึ่งช่วยให้ผู้พัฒนามั่นใจว่าแอปพลิเคชันสามารถทำงานได้อย่างถูกต้องและสอดคล้องกันในทุกแพลตฟอร์ม ซึ่งรองรับการทดสอบหลายลักษณะ เช่น End-to-End Testing (E2E) ซึ่งใช้ตรวจสอบกระบวนการทำงานทั้งหมดของระบบตั้งแต่ต้นจนจบ, UI Testing สำหรับตรวจสอบการแสดงผลและปฏิสัมพันธ์ของผู้ใช้งาน และ Web Scraping สำหรับดึงข้อมูลจากเว็บไซต์โดยอัตโนมัติ ทั้งนี้ Playwright จึงเป็นเครื่องมือที่มีความยืดหยุ่นและครอบคลุมสำหรับการทดสอบเว็บแอปพลิเคชันอย่างครบวงจร คุณสมบัติเด่นของ Playwright ได้แก่

##### 5.1.1 Cross-Browser Testing

รองรับการทดสอบบนเบราว์เซอร์หลายชนิด ได้แก่ Chrome, Firefox และ Safari รวมทั้งสามารถทดสอบได้ทั้งโหมด Desktop และ Mobile

##### 5.1.2 รองรับหลายภาษาโปรแกรม

สามารถเขียนสคริปต์ทดสอบด้วย JavaScript, TypeScript, Python, Java หรือ .NET ตามความถนัดของทีมงาน

##### 5.1.3 Automated Waiting

มีระบบ Auto-wait ที่รอให้ element ปรากฏและพร้อมใช้งานก่อนดำเนินการโต้ตอบช่วยลดความล้มเหลวของการทดสอบเนื่องจากความล่าช้าในการโหลดหน้าเว็บ

##### 5.1.4 Headless & Headed Mode

สามารถรันการทดสอบแบบไม่มี UI (Headless) เพื่อเพิ่มความเร็ว หรือแบบมี UI (Headed) เพื่อสังเกตการทำงานของสคริปต์อย่างละเอียด

### 5.1.5 Network Interception

สามารถดักจับหรือจำลองการตอบสนองของ API เพื่อทดสอบสถานการณ์ต่าง ๆ เช่น การตอบสนองล่าช้า หรือข้อผิดพลาดของเซิร์ฟเวอร์

### 5.1.6 Parallel Execution

รองรับการรันชุดการทดสอบหลายชุดพร้อมกัน ช่วยประหยัดเวลาและเพิ่มประสิทธิภาพการทดสอบระบบ

ด้วยคุณสมบัติข้างต้น Playwright จึงถือเป็นเครื่องมือที่ได้รับความนิยมจากนักพัฒนาและทีมงาน QA/Tester ในการสร้างการทดสอบอัตโนมัติสำหรับเว็บแอปพลิเคชัน ทั้งนี้เพื่อเพิ่มความน่าเชื่อถือของระบบและลดความเสี่ยงจากข้อผิดพลาดที่อาจเกิดขึ้นในการทำงานของระบบ

## 5.2 วิเคราะห์การออกแบบการทดสอบระบบ

ในการทดสอบระบบแบบอัตโนมัติ ระบบนี้เป็นระบบภายในองค์กรซึ่งมีสิทธิ์การใช้งาน 2 ระดับ ได้แก่ ผู้ดูแลระบบ (Admin) และผู้ใช้งานทั่วไป (User) ดังแสดงในรูปที่ 5.1 ซึ่งสรุปขั้นตอนการทำงานของระบบได้ดังนี้

### 5.2.1 การเพิ่มบัญชีผู้ใช้งาน Staff

ผู้ดูแลระบบ (Admin) มีหน้าที่ในการเพิ่มบัญชีผู้ใช้งานประเภท Staff เข้าสู่ระบบ ซึ่ง Staff อาจเป็นผู้ที่ได้รับมอบหมายให้ดูแลการแก้ไขปัญหาหรือจัดการข้อมูลอื่น ๆ ในระบบ

### 5.2.2 การเพิ่มข้อมูลโครงการ

ทั้งผู้ดูแลระบบ (Admin) และผู้ใช้งาน (User) สามารถเพิ่มโครงการหรืองานของลูกค้าเข้าสู่ระบบ โดยโครงการเหล่านี้อาจเป็นระบบที่ต้องการการบำรุงรักษาหรือดูแลภายใต้สัญญาบำรุงรักษา

### 5.2.3 การเพิ่มเอกสารสัญญาและเอกสารที่เกี่ยวข้อง

ทั้งผู้ดูแลระบบ (Admin) และผู้ใช้งาน (User) สามารถเพิ่มเอกสารสัญญาและเอกสารที่เกี่ยวข้องกับโครงการเพื่อเป็นหลักฐาน และเพื่อให้มั่นใจว่าระบบสามารถจัดการการบำรุงรักษาโครงการได้อย่างมีประสิทธิภาพ

#### 5.2.4 การเพิ่มรายละเอียดของโครงการ

ทั้งผู้ดูแลระบบ (Admin) และผู้ใช้งาน (User) สามารถเพิ่มรายละเอียดหรือขอบเขตของโครงการหรืองานของลูกค้าเข้าสู่ระบบ โดยรายละเอียดนี้จะช่วยให้การบำรุงรักษาหรือการดูแลโครงการเป็นไปตามเอกสารสัญญาบำรุงรักษา

#### 5.2.5 การแจ้งรายงานปัญหา

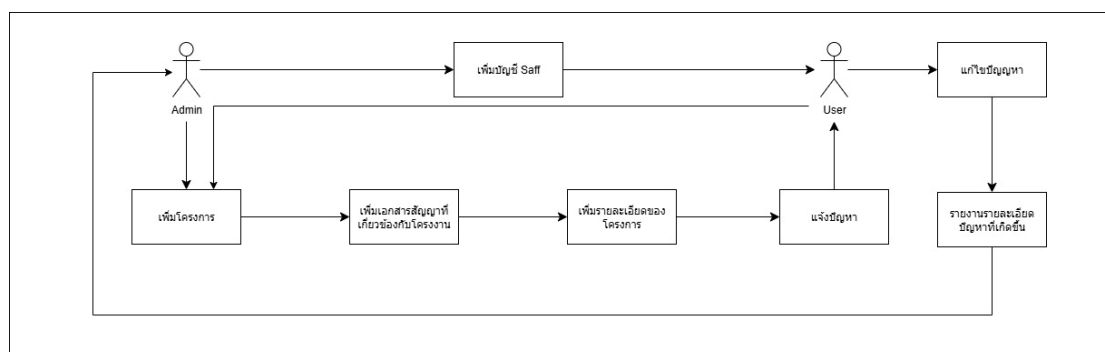
ทั้งผู้ดูแลระบบ (Admin) และผู้ใช้งาน (User) ซึ่งอาจเป็นผู้รับผิดชอบในการดูแลโครงการ สามารถแจ้งปัญหาที่เกิดขึ้นในระบบที่ได้รับรายงานจากลูกค้า กรอกลงในระบบ และสามารถมอบหมายให้ทีมงานใด ๆ เป็นผู้รับผิดชอบในการแก้ไขปัญหา พร้อมทั้งติดตามสถานะการแก้ไขปัญหาผ่านระบบ

#### 5.2.6 การแก้ไขปัญหา

ผู้ดูแลระบบ (Admin) หรือ ผู้ใช้งาน (User) แก้ไขปัญหา เมื่อมีการแจ้งปัญหาจากผู้ใช้งาน Admin หรือทีมพัฒนาได้รับแจ้งเตือนการมอบหมายให้เป็นผู้รับผิดชอบจะดำเนินการแก้ไขปัญหาที่เกิดขึ้น รวมถึงสามารถแจ้งสถานการณ์แก้ไขด้วย

#### 5.2.7 การบันทึกรายละเอียดของปัญหา

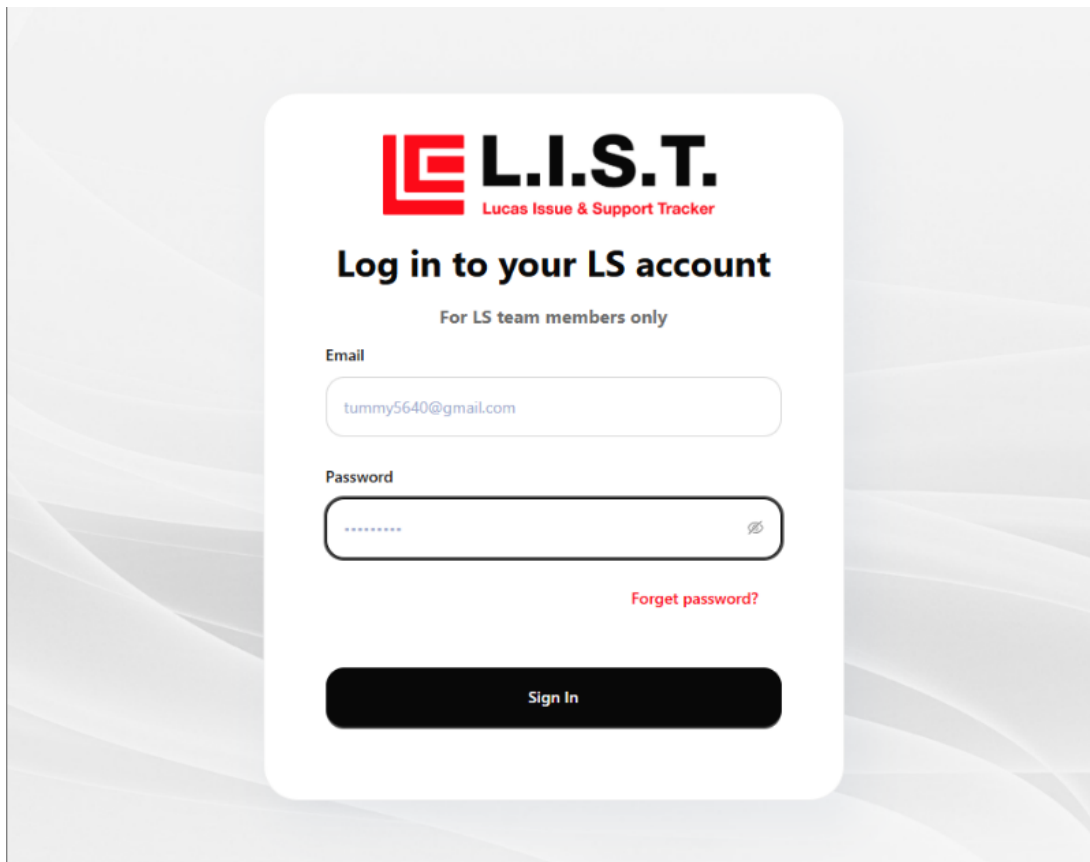
หลังจากการแก้ไขปัญหาเสร็จสิ้น ระบบจะบันทึกรายละเอียดของปัญหาที่ได้รับการแก้ไขเพื่อเป็นประวัติการแก้ไขปัญหาในอนาคต ขั้นตอนเหล่านี้ครอบคลุมตั้งแต่การเพิ่มโครงการข้อมูลเอกสาร และรายละเอียดทั้งหมดที่เกี่ยวข้องกับโครงการของลูกค้า ไปจนถึงกระบวนการแจ้งและแก้ไขปัญหา โดย Admin และ User ทำงานร่วมกันอย่างเป็นระบบ



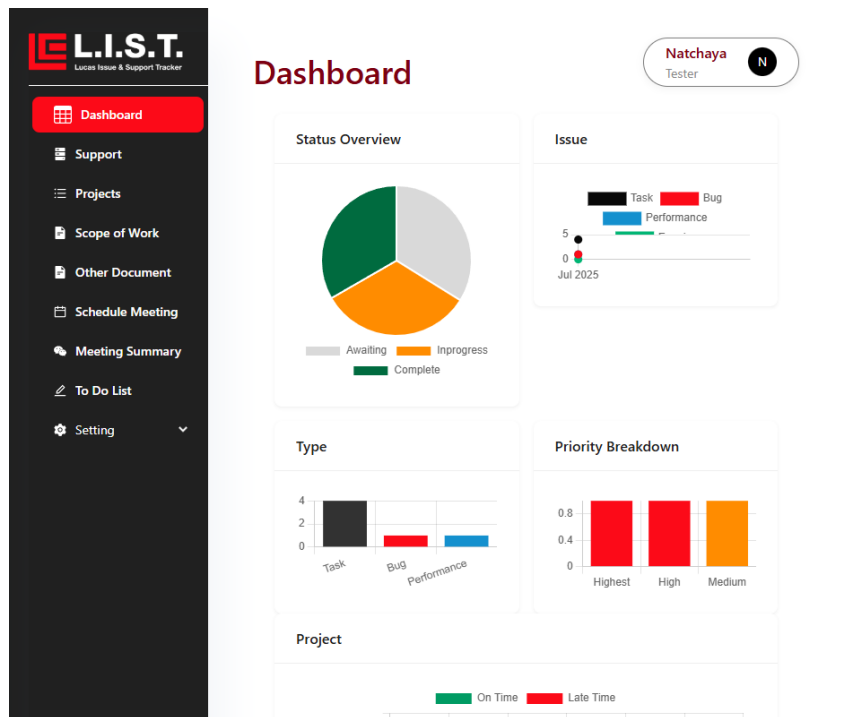
รูปที่ 5.1 รูปภาพแสดงขั้นตอนการทำงานของระบบ

### 5.3 การทดสอบระบบ

5.3.1 เข้าสู่ระบบ ในการทดสอบการเข้าสู่ระบบเริ่มต้นจากเข้าสู่หน้าเข้าสู่ระบบ จากนั้นกรอก Email และ Password มีการตรวจสอบการกรอกข้อมูลว่าถูกกรอกลงในช่องหรือไม่ จากนั้นกดปุ่ม Sign In ถ้า Email หรือ Password ถูกต้อง จะเข้าสู่ระบบแล้วไปที่หน้า Dashboard ทันที และ ถ้า Email หรือ Password ไม่ถูกต้องจะตรวจสอบว่ามีข้อความ Error Message แสดงขึ้นมา



รูปที่ 5.2 ขั้นตอนการทดสอบในการกรอก Email และ Password เข้าสู่ระบบ



รูปที่ 5.3 ขั้นตอนการทดสอบในการกรอก Email และ Password ถูกต้อง  
จะเข้าสู่ระบบแล้วไปที่หน้า Dashboard ทันที

**L.I.S.T. Lucas Issue & Support Tracker**

**Log in to your LS account**

For LS team members only

Email

tummy50@gmail.com

Password

\*\*\*\*\*

[Forget password?](#)

**Sign In**

Invalid email or password

รูปที่ 5.4 ขั้นตอนการทดสอบในการกรอก Email และ Password ผิดพลาด

5.3.2 การทดสอบการเพิ่มโครงการเริ่มจากการเข้าสู่หน้า Dashboard ของระบบ จากนั้นเลือกเมนู “Setting” และเลือกฟังก์ชัน “Add Project” เมื่อเข้าถึงหน้าจัดการข้อมูลโครงการแล้วให้คลิกปุ่ม “Add Project” เพื่อให้ระบบแสดงฟอร์มสำหรับเพิ่มโครงการขึ้นมา จากนั้นกรอกข้อมูลโครงการลงในฟอร์ม และตรวจสอบว่าข้อมูลสามารถกรอกลงในช่องต่าง ๆ ได้อย่างถูกต้อง หลังจากกรอกข้อมูลครบถ้วน คลิกปุ่ม “บันทึก” (Save) และตรวจสอบว่าข้อมูลโครงการที่เพิ่มปรากฏอยู่ในตารางรายการโครงการอย่างถูกต้อง

รูปที่ 5.5 ขั้นตอนการทดสอบการเพิ่มโครงการ

5.3.3 การทดสอบการเพิ่มเอกสารสัญญาและเอกสารที่เกี่ยวข้องเริ่มจากการเข้าสู่หน้า “Other Document” ของระบบ จากนั้นคลิกปุ่ม “Add Other Document” เพื่อให้ระบบแสดงฟอร์มสำหรับเพิ่มเอกสารขึ้นมา จากนั้นกรอกข้อมูลเอกสารลงในฟอร์ม และตรวจสอบว่าข้อมูลสามารถกรอกลงในช่องต่าง ๆ ได้อย่างถูกต้อง หลังจากกรอกข้อมูลครบถ้วน คลิกปุ่ม “บันทึก” (Save) และตรวจสอบว่าข้อมูลที่เพิ่มปรากฏอยู่ในตารางรายการ Other Document อย่างถูกต้อง นอกจากนี้ยังตรวจสอบความสามารถในการเปิดและดาวน์โหลดไฟล์แนบเพื่อยืนยันความสมบูรณ์ของเอกสาร

A screenshot of a file selection dialog box. The title bar at the top says "Other Document". Below the title bar, there is a header text "เลือกไฟล์ที่ต้องการดาวน์โหลด" (Select file to download) and a close button (X) on the right. The list of files includes "PROJECT.pdf", "Automated Testing 3.pdf", and "ตัวอย่าง ค่าเข้า.xlsx".

รูปที่ 5.7 ขั้นตอนการทดสอบในการจะดาวน์โหลดไฟล์ที่ต้องการของหน้า Other Document

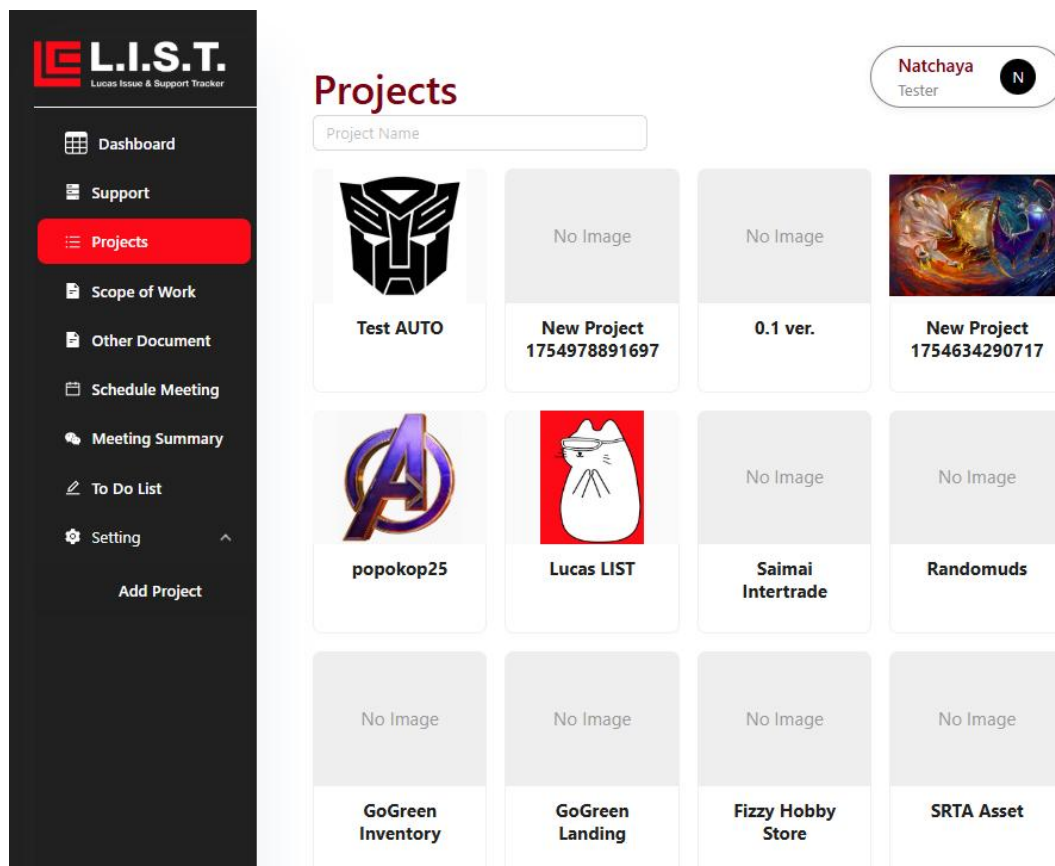
5.3.4 การทดสอบการเพิ่มรายละเอียดของโครงการเริ่มต้นจากการเข้าสู่หน้า “Scope of Work” ของระบบ จากนั้นคลิกปุ่ม “Add SOW” เพื่อให้ระบบแสดงฟอร์มสำหรับเพิ่มรายละเอียดของโครงการขึ้นมากรอกข้อมูลรายละเอียดของโครงการลงในฟอร์ม และตรวจสอบว่าข้อมูลสามารถกรอกลงในช่องต่าง ๆ ได้อย่างถูกต้อง หลังจากกรอกข้อมูลครบถ้วน คลิกปุ่ม “บันทึก” (Save) และตรวจสอบว่าข้อมูลที่เพิ่มปรากฏอยู่ในตารางรายการ Scope of Work อย่างถูกต้อง นอกจากนี้ยังตรวจสอบความสามารถในการเปิดและดาวน์โหลดไฟล์แนบเพื่อยืนยันความสมบูรณ์ของเอกสาร

รูปที่ 5.8 3 ขั้นตอนการทดสอบการเพิ่มรายละเอียดของโครงการ


รูปที่ 5.9 ขั้นตอนการทดสอบในการจะดาวน์โหลดไฟล์ที่ต้องการของหน้า Scope of Work



5.3.5 การทดสอบการแจ้งรายงานปัญหาเริ่มต้นจากการเข้าสู่หน้า “Projects” ของระบบ จากนั้นเลือกโครงการที่ต้องการรายงานปัญหา เมื่อเลือกโครงการเรียบร้อยแล้ว ให้คลิกปุ่ม “Add Issue” เพื่อเพิ่มรายงานปัญหา จากนั้นกรอกข้อมูลรายละเอียดของปัญหาที่เกิดขึ้น พร้อมกำหนดผู้รับผิดชอบโครงการ และตรวจสอบความถูกต้องของข้อมูลที่กรอกลงในฟอร์ม หลังจากกรอกข้อมูลครบถ้วน คลิกปุ่ม “บันทึก” และตรวจสอบว่าข้อมูลรายงานปัญหาที่เพิ่มเข้ามาปรากฏอยู่ในตารางรายงานปัญหาอย่างถูกต้อง



รูปที่ 5.8 ขั้นตอนการทดสอบในการเลือกโครงการ เพื่อที่จะดำเนินการแจ้งรายงานปัญหา



Lucas Issue & Support Tracker

- Dashboard
- Support
- Projects**
- Scope of Work
- Other Document
- Schedule Meeting
- Meeting Summary
- To Do List
- Setting
- Add Project

## Add Issue

เพิ่ม Issue ใหม่ในโปรเจกต์ #Test-005

\* Issue Code

Test-005-082025-002

Issue Date

15/08/25

Type

Bug

Priority

Medium

Title

Implement Login Feature

Description

Create secure login system

\* Status

Awaiting

Start Date

15/08/25

Due Date

29/08/25

Complete Date

Select date

Developer

Thorarin


BA/Test

Natchaya

Remark

Additional Document

รูปที่ 5.8 ขั้นตอนการทดสอบการแจ้งรายงานปัญหา



Lucas Issue & Support Tracker

- Dashboard
- Support
- Projects**
- Scope of Work
- Other Document
- Schedule Meeting
- Meeting Summary
- To Do List
- Setting
- Add Project

## Project Detail

Test AUTO. Issue Log

+ Add Issue

No.	Issue Code	Issue Date	Title	Description	Status	Start Date	Due Date	Complete Date	On/Late Time	Developer
3	Test-005-082025-003	15/08/25	Optimize Dashboard Loading Spe...	ปัญหาที่: Dashboard โหลดช้า ...	Inprogress	15/08/25	28/08/25	-		Thorarin
2	Test-005-082025-002	15/08/25	Fix Payment Gateway Timeout	Fix payment timeout bug	Inprogress	15/08/25	28/08/25	-		Thorarin
1	Test-005-082025-001	15/08/25	Implement Login Feature	Create secure login system	Awaiting	14/08/25	28/08/25	-		Thorarin

ทั้งหมด 3 รายการ

1

รูปที่ 5.8 ขั้นตอนการทดสอบการแจ้งรายงานปัญหาในโครงการนั้น

5.3.6 การทดสอบการตอบรายงานปัญหาเริ่มต้นจากการเข้าสู่หน้า “Projects” ของระบบ จากนั้นเลือกโครงการที่ต้องการตอบกลับรายงานปัญหา เมื่อเข้าถึงโครงการเรียบร้อยแล้ว ให้เลือกรายงานปัญหาที่ต้องการตอบกลับ จากนั้นทำการแก้ไขข้อมูลที่จำเป็น เปลี่ยนสถานะของรายงานปัญหา และกรอกสาเหตุหรือรายละเอียดของปัญหา พร้อมตรวจสอบความถูกต้องของข้อมูลที่กรอกลงในฟอร์ม หลังจากกรอกข้อมูลครบถ้วน คลิกปุ่ม “บันทึก” และตรวจสอบว่าสถานะของรายงานปัญหาถูกอัปเดตในตารางรายงานปัญหาอย่างถูกต้อง

รูปที่ 5.8 ขั้นตอนการทดสอบการตอบรายงานปัญหา

No.	Issue Code	Issue Date	Title	Description	Status	Start Date	Due Date	Complete Date	On/Late Time	Developer
3	Test-005-082025-003	15/08/25	Optimize Dashboard Loading Spe...	ปัญหาที่พบ: Dashboard โหลดช้า ...	Complete	15/08/25	28/08/25	15/08/25	On Time (13 Day)	Thorarin
2	Test-005-082025-002	15/08/25	Fix Payment Gateway Timeout	Fix payment timeout bug	Inprogress	15/08/25	28/08/25	-		Thorarin
1	Test-005-082025-001	15/08/25	Implement Login Feature	Create secure login system	Awaiting	14/08/25	28/08/25	-		Thorarin

รูปที่ 5.8 ขั้นตอนการทดสอบการตอบรายงานปัญหาในโครงการนั้นแล้ว

## 5.4 การแสดงผลการทดสอบระบบแบบ Automated Testing ด้วย Playwright

การแสดงผลการทดสอบระบบด้วย Playwright สามารถนำเสนอได้หลายรูปแบบ โดยมุ่งเน้นให้ผู้ทดสอบและผู้พัฒนาสามารถตรวจสอบความถูกต้องของระบบได้อย่างรวดเร็วและชัดเจน ทั้งนี้ การแสดงผลหลักของ Playwright มีดังนี้

### 5.4.1 ผลการทดสอบผ่าน Command Line Interface (CLI)

เมื่อทำการรันชุดการทดสอบโดยใช้คำสั่ง `npx playwright test` ระบบจะแสดงผลลัพธ์ในรูปแบบข้อความบนหน้าจอ CLI โดยจะระบุไว้แต่ละ Test Case มีสถานะ “ผ่าน” (Passed), “ไม่ผ่าน” (Failed) หรือ “ถูกข้าม” (Skipped) พร้อมระบุเวลาในการรันแต่ละกรณี ซึ่งช่วยให้สามารถประเมินความถูกต้องของระบบได้ทันที

### 5.4.2 รายงานผลการทดสอบ (Test Report)

Playwright รองรับการสร้างรายงานในรูปแบบ HTML Report ที่สามารถเปิดดูผ่านเบราว์เซอร์ได้ รายงานดังกล่าวจะสรุปจำนวนกรณีทดสอบทั้งหมด จำนวนที่ผ่าน จำนวนที่ไม่ผ่าน รวมถึงรายละเอียดเชิงลึก เช่น stack trace ของ error เพื่อให้ผู้ทดสอบสามารถตรวจสอบสาเหตุของข้อผิดพลาดได้อย่างละเอียด

### 5.4.3 การบันทึกหลักฐานการทดสอบ (Evidence)

เพื่อเพิ่มความน่าเชื่อถือของการทดสอบ Playwright มีความสามารถในการบันทึกผลลัพธ์ในรูปแบบต่าง ๆ เช่น Screenshot, Video และ Trace log ของทุกการทำงาน (เช่น การคลิก ป้อนข้อมูล หรือการโหลดหน้าเว็บ) หลักฐานดังกล่าวช่วยให้ทีมทดสอบสามารถวิเคราะห์ย้อนหลังและยืนยันผลการทำงานของระบบได้อย่างถูกต้อง

### 5.4.4 การนำเสนอผลลัพธ์ในการรายงานโครงการ

ในการจัดทำรายงานโครงการ สามารถนำภาพหน้าจอจาก CLI, HTML Report หรือ Screenshot ของหน้าจกระบบมาประกอบ เพื่อแสดงให้เห็นตัวอย่างผลการทดสอบจริง โดยมีการอธิบายรายละเอียดประกอบภาพว่า Test Case ใดที่ผ่านการทดสอบหรือพบข้อผิดพลาด ซึ่งทำให้การนำเสนอมีความสมบูรณ์และเข้าใจง่าย

การแสดงผลการทดสอบด้วย Playwright ช่วยให้ผู้ที่ทดสอบสามารถตรวจสอบผลลัพธ์ได้อย่างครบถ้วน ในหลายมิติ ทั้งการแสดงผลแบบข้อความผ่าน Command Line Interface ที่บอกสถานะการทดสอบอย่างชัดเจน การสร้างรายงานในรูปแบบ HTML ที่นำเสนอรายละเอียดเชิงลึก และการบันทึกหลักฐานการทดสอบในรูปแบบ Screenshot, Video และ Trace log ซึ่งสามารถใช้ตรวจสอบย้อนหลังได้อย่างมีประสิทธิภาพ ดังนั้น การแสดงผลของ Playwright จึงมีความเหมาะสมอย่างยิ่งต่อการนำไปประยุกต์ใช้ในกระบวนการทดสอบระบบจริง เนื่องจากมีความน่าเชื่อถือ โปร่งใส และสนับสนุนการทำงานร่วมกันระหว่างทีมพัฒนาและทีมทดสอบได้อย่างมีประสิทธิภาพ

Project	Test File	Test Name	Status	Duration
chromium	login.spec.js	ระบบล็อกอิน > กรณีล็อกอินสำเร็จ > ล็อกอินด้วยผู้ใช้ปกติ	Passed	9.4s
chromium	login.spec.js	login.spec.js:23	Passed	
chromium	login.spec.js	ระบบล็อกอิน > กรณีล็อกอินล้มเหลว > กรอกอีเมลผิด	Failed	8.3s
chromium	login.spec.js	login.spec.js:35	Failed	
chromium	login.spec.js	ระบบล็อกอิน > กรณีล็อกอินล้มเหลว > กรอกรหัสผ่านผิด	Failed	8.3s
chromium	login.spec.js	login.spec.js:40	Failed	
chromium	login.spec.js	ระบบล็อกอิน > กรณีล็อกอินล้มเหลว > ไม่กรอกอีเมล	Failed	8.9s
chromium	login.spec.js	login.spec.js:45	Failed	
chromium	login.spec.js	ระบบล็อกอิน > กรณีล็อกอินล้มเหลว > ไม่กรอกรหัสผ่าน	Failed	9.0s
chromium	login.spec.js	login.spec.js:50	Failed	
chromium	login.spec.js	ระบบล็อกอิน > กรณีล็อกอินล้มเหลว > อีเมลรูปแบบไม่ถูกต้อง	Failed	8.8s
chromium	login.spec.js	login.spec.js:55	Failed	
chromium	login.spec.js	ระบบล็อกอิน > กรณีล็อกอินล้มเหลว > บัญชีถูกล็อก	Failed	9.1s
chromium	login.spec.js	login.spec.js:60	Failed	
chromium	login.spec.js	ระบบล็อกอิน > พิเจอร์ "ลืมรหัสผ่าน" > ลิงก์ "ลืมรหัสผ่าน" แสดงผลและลิงก์ถูกต้อง	Passed	4.8s
chromium	login.spec.js	login.spec.js:67	Passed	
chromium	login.spec.js	ระบบล็อกอิน > พิเจอร์ "ลืมรหัสผ่าน" > คลิกลิงก์ "ลืมรหัสผ่าน" เปลี่ยนหน้าได้	Passed	6.6s
chromium	login.spec.js	login.spec.js:72	Passed	

รูปที่ 5.9 ภาพแสดงผลการทดสอบระบบด้วย Playwright หน้า Login

Project	Test File	Test Name	Status	Duration
chromium	dashboard.spec.js	Dashboard Page Tests > ตรวจสอบ header ของ Dashboard เมื่อเข้าสู่ระบบสำเร็จ	Passed	4.6s
chromium	dashboard.spec.js	dashboard.spec.js:21	Passed	
chromium	dashboard.spec.js	Dashboard Page Tests > เมนู Dashboard แสดงข้อมูลภาพรวม	Passed	4.8s
chromium	dashboard.spec.js	dashboard.spec.js:38	Passed	

รูปที่ 5.10 ภาพแสดงผลการทดสอบระบบด้วย Playwright หน้า Dashboard

All4

Passed4

Failed0

Flaky0

Skipped0

Project: chromium

8/15/2025, 4:36:22 PM   Total time: 14.0s

▼

setting.spec.js

✓

⚙️

Setting Page - Project Tests >

✓

Add new project (Staff)

10.0s

setting.spec.js:25

✓

⚙️

Setting Page - Project Tests >

🗑️

Delete project from More menu and confirm Yes

11.7s

setting.spec.js:36

✓

👤

User Access Tests >

🚫

Staff should not see Add User menu

6.2s

setting.spec.js:50

✓

🔍

View/Edit Project Tests >

✏️

View and edit project details

11.9s

setting.spec.js:57

รูปที่ 5.11 ภาพแสดงผลการทดสอบระบบด้วย Playwright ในขั้นตอนการเพิ่มข้อมูลของโครงการ

Q

All 4Passed 4Failed 0Flaky 0Skipped 0

Project: chromium8/29/2025, 10:50:15 AMTotal time: 12.2s

▼ otherDocument.spec.js

✓📄 Other Document Management > ✓เพิ่ม Other Document สำเร็จ

otherDocument.spec.js:28

10.4s

✓📄 Other Document Management > ⚠️แสดง validation errors เมื่อไม่กรอกข้อมูล

otherDocument.spec.js:36

9.0s

✓✏️ Other Document View/Edit > ✏️แก้ไขแถวแรก

otherDocument.spec.js:56

9.9s

✓🗑️ Other Document Delete > 🗑️ลบแถวแรก

otherDocument.spec.js:77

8.2s

รูปที่ 5.12 ภาพแสดงผลการทดสอบระบบด้วย Playwright หน้า Other Document ในขั้นตอนเพิ่มเอกสารสัญญาและเอกสารที่เกี่ยวข้อง

Q

All 4

Passed 4

Failed 0

Flaky 0

Skipped 0

Project: chromium

7/29/2025, 4:08:06 PM Total time: 12.2s

▼ scopeOfWork.spec.js

✓📁 Scope of Work Management > ✓ เพิ่ม Scope of Work: ทรากฟาร์มครบและบันทึกสำเร็จ

scopeOfWork.spec.js:21

10.0s

✓📁 Scope of Work Management > ✗ แสดงข้อความ error เมื่อกรอกข้อมูลไม่ครบ

scopeOfWork.spec.js:41

8.0s

✓📁 Scope of Work Management > ✏️ คลิก More แล้ว View/Edit แถวแรก และแก้ไขข้อมูล

scopeOfWork.spec.js:53

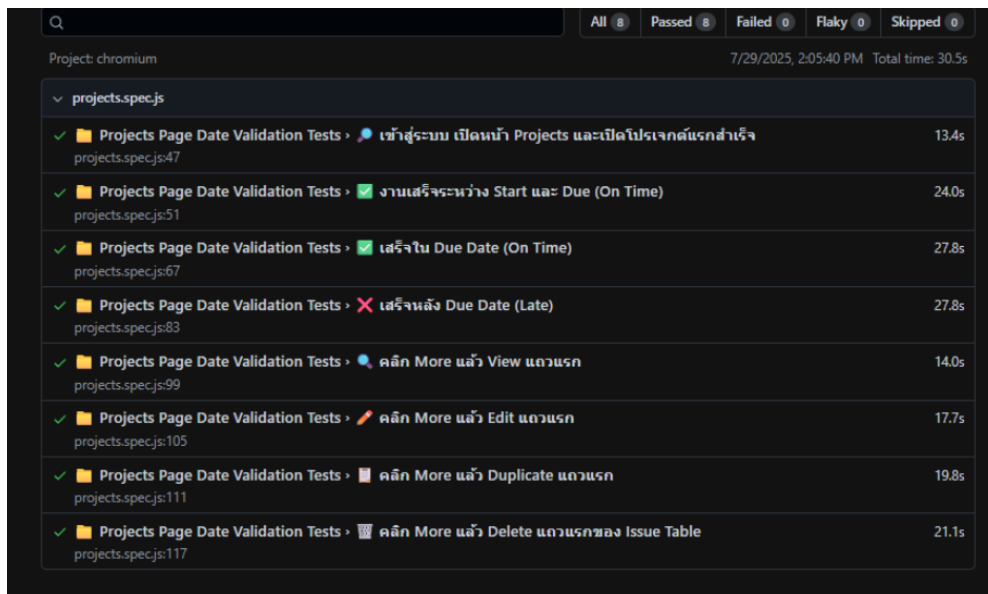
10.5s

✓📁 Scope of Work Management > 🗑️ คลิก More แล้ว Delete แถวแรกของ Scope of Work

scopeOfWork.spec.js:77

7.7s

รูปที่ 5.13 ภาพแสดงผลการทดสอบระบบด้วย Playwright หน้า Scope of Work ในขั้นตอนเพิ่มรายละเอียดของโครงการ



รูปที่ 5.14 ภาพแสดงผลการทดสอบระบบด้วย Playwright ในขั้นตอนแจ้งรายงานของปัญหา และตอบกลับการรายงานของปัญหา

## 5.5 ผลการทดสอบระบบ

ตารางที่ 5.1 ตารางแสดงการทดสอบระบบของหน้า Login

#	Test Scenario	Expected Result	Test Date	Test Result
Login				
1	Login สำเร็จด้วยข้อมูลถูกต้อง	เข้าสู่ระบบสำเร็จและแสดงหน้า Dashboard	11/7/68	Pass
2	Login ล้มเหลว กรอก Email หรือ Password ผิด	แสดงข้อความแจ้งเตือน "อีเมลหรือรหัสผ่านไม่ถูกต้อง"	11/7/68	Pass
3	ล้มเหลว กรอก Email หรือ Password ว่าง	แสดงข้อความแจ้งเตือน "อีเมลหรือรหัสผ่านไม่ถูกต้อง"	11/7/68	Pass
4	แสดงปุ่ม "ลืมรหัสผ่าน" และคลิกได้	นำผู้ใช้ไปยังหน้ารีเซตรหัสผ่าน	11/7/68	Pass
5	ตรวจสอบการล็อกอินหลังกรอกข้อมูลครบ	เข้าสู่ระบบสำเร็จเหมือนกรณีกดปุ่ม Login	11/7/68	Pass

ตารางที่ 5.2 ตารางแสดงการทดสอบระบบของหน้า การเพิ่มโครงการ

#	Test Scenario	Expected Result	Test Date	Test Result
Add Project				
1	แสดงรายการ Project ทั้งหมด	รายการ Project ปรากฏพร้อม Project ID, ชื่อ และโลโก้ (ถ้ามี)	16/7/68	Pass
2	เพิ่มโปรเจกต์ใหม่ด้วยข้อมูลครบถ้วน	โปรเจกต์ใหม่ถูกเพิ่มและแสดงในรายการ พร้อมโลโก้ถูกต้อง	16/7/68	Pass
3	เพิ่มโปรเจกต์ใหม่โดยไม่กรอกข้อมูลจำเป็น	แสดงข้อความแจ้งเตือนให้กรอกข้อมูลจำเป็น	16/7/68	Pass
4	แก้ไขโปรเจกต์	สามารถแก้ไขข้อมูลโปรเจกต์ได้	16/7/68	Pass
5	ลบโปรเจกต์	โปรเจกต์ถูกลบออกจากรายการ	29/7/68	Pass

ตารางที่ 5.3 ตารางแสดงการทดสอบระบบของหน้า การเพิ่มบัญชีผู้ใช้งาน

#	Test Scenario	Expected Result	Test Date	Test Result
Add User				
1	แสดงรายการ User ทั้งหมด	รายการ User ปรากฏครบพร้อมข้อมูล	16/7/68	Pass
2	เพิ่ม User ใหม่ด้วยข้อมูลครบถ้วน	User ถูกเพิ่มและแสดงข้อมูลครบ	16/7/68	Pass
3	เพิ่ม User โดยไม่กรอกข้อมูลจำเป็น	แสดงข้อความแจ้งเตือนให้กรอกข้อมูลจำเป็น	16/7/68	Pass
4	แก้ไข User	ข้อมูล User ถูกแก้ไขและปรากฏในรายการ	16/7/68	Pass
5	ลบ User	User ถูกลบออกจากรายการ	16/7/68	Pass



ตารางที่ 5.4 ตารางแสดงการทดสอบระบบของหน้า การเพิ่มเอกสารสัญญาและเอกสารที่เกี่ยวข้อง

#	Test Scenario	Expected Result	Test Date	Test Result
Other Document				
1	แสดงรายการ Other Document ทั้งหมด	รายการ Other Document ปรากฏครบถ้วน	8/8/68	Pass
2	เพิ่ม Other Document พร้อมข้อมูลครบ	Other Document ถูกเพิ่มและแสดงข้อมูลครบ	8/8/68	Pass
3	เพิ่ม Other Document โดยไม่กรอกข้อมูลจำเป็น	แสดงข้อความแจ้งเตือนให้กรอกข้อมูลก่อนบันทึก	8/8/68	Pass
4	อัปโหลดไฟล์แนบ	ไฟล์ถูกอัปโหลดและชื่อไฟล์ปรากฏในรายการ	8/8/68	Pass
5	แก้ไขข้อมูล	ข้อมูลถูกแก้ไขและแสดงผล	8/8/68	Pass
6	ลบข้อมูล	ถูกลบออกจากรายการ	8/8/68	Pass

ตารางที่ 5.5 ตารางแสดงการทดสอบระบบของหน้า การเพิ่มรายละเอียดของโครงการ

#	Test Scenario	Expected Result	Test Date	Test Result
Scope Of Work				
1	แสดงรายการ Scope of Work ทั้งหมด	รายการ Scope of Work ปรากฏครบถ้วน	24/7/68	Pass
2	เพิ่ม Scope of Work ใหม่พร้อมข้อมูลครบถ้วน	Scope of Work ถูกเพิ่มและแสดงข้อมูลครบ	24/7/68	Pass
3	เพิ่ม Scope of Work โดยไม่กรอกข้อมูลจำเป็น	แสดงข้อความแจ้งเตือนให้กรอกข้อมูลก่อนบันทึก	24/7/68	Pass
4	อัปโหลดไฟล์แนบ	ไฟล์ถูกอัปโหลดและชื่อไฟล์ปรากฏในรายการ	24/7/68	Pass
5	แก้ไขข้อมูล	ข้อมูลถูกแก้ไขและแสดงผล	24/7/68	Pass
6	ลบข้อมูล	ถูกลบออกจากรายการ	24/7/68	Pass

ตารางที่ 5.6 ตารางแสดงการทดสอบระบบของหน้า การแจ้งรายงานปัญหา

#	Test Scenario	Expected Result	Test Date	Test Result
Projects				
1	แสดงรายการโปรเจกต์ทั้งหมด	รายการโปรเจกต์ปรากฏครบถ้วน	29/7/68	Pass
2	แสดงรายการ Issue ของโปรเจกต์	Issue ของโปรเจกต์ปรากฏครบ	29/7/68	Pass
3	เพิ่ม Issue ใหม่พร้อมข้อมูลครบถ้วน	Issue ถูกเพิ่มและแสดงข้อมูลครบ	29/7/68	Pass
5	แก้ไขข้อมูล พร้อมอัปเดตสถานะของ Issue	ข้อมูลถูกแก้ไขและแสดงผล	29/7/68	Pass
6	สำเนาข้อมูล	ข้อมูลถูกสำเนาและแสดงผล	29/7/68	Pass
7	ลบข้อมูล	ข้อมูลถูกลบออกจากรายการ	29/7/68	Pass

## 5.6 วิเคราะห์ผลการทำ Accessibility Testing

Accessibility Testing คือกระบวนการทดสอบเว็บไซต์หรือระบบซอฟต์แวร์เพื่อให้แน่ใจว่าผู้ใช้ทุกกลุ่มสามารถเข้าถึงข้อมูลและใช้งานฟังก์ชันต่าง ๆ ได้อย่างเท่าเทียม ไม่จำกัดความสามารถทางร่างกายหรือประสาทสัมผัส เช่น ผู้มีความบกพร่องทางสายตา การได้ยิน การเคลื่อนไหว หรือความเข้าใจระบบ ในโครงการนี้ การทดสอบความสามารถในการเข้าถึงจะใช้เครื่องมืออัตโนมัติหลายชนิดเพื่อประเมินองค์ประกอบต่าง ๆ ของเว็บไซต์ ได้แก่

1. Includia Accessibility Checker เป็นเครื่องมือออนไลน์ที่วิเคราะห์หน้าเว็บตามหลักเกณฑ์ WCAG 2.1 โดยสามารถตรวจสอบองค์ประกอบหลายด้าน เช่น การใช้สี (Color Contrast), การจัดโครงสร้างเนื้อหา (Content Structure), การใช้งานผ่านคีย์บอร์ด (Keyboard Navigation), และการเข้าถึงองค์ประกอบด้วยเครื่องมือช่วยเหลือต่าง ๆ (Assistive Technology)

การแสดงผลและรายงานของ Includia Accessibility เมื่อทำการทดสอบหน้าเว็บ เครื่องมือจะแสดงผลเป็นรายงานที่แบ่งตาม ระดับความรุนแรงของปัญหา (Severity Level) ได้แก่

1. Critical ปัญหาที่ร้ายแรงมาก มีผลกระทบต่อการใช้งานของผู้ใช้ที่มีความบกพร่องทางการเข้าถึงอย่างรุนแรง เช่น ปุ่มไม่สามารถเข้าถึงได้ด้วยคีย์บอร์ด หรือข้อความไม่สามารถอ่านได้ด้วย screen reader
2. High ปัญหาที่สำคัญ มีผลกระทบต่อการเข้าใจหรือการใช้งานของผู้ใช้บางกลุ่ม เช่น ความแตกต่างของสีระหว่างข้อความกับพื้นหลังต่ำเกินไป
3. Medium ปัญหาที่มีผลกระทบปานกลาง เช่น การจัดลำดับ heading ไม่ถูกต้อง หรือองค์ประกอบบางส่วนขาดคำอธิบาย
4. Low ปัญหาเล็กน้อย ไม่กระทบต่อการใช้งานหลัก แต่เป็นข้อแนะนำในการปรับปรุง เช่น ข้อความ placeholder ขาดความชัดเจน

Includia Accessibility Checker เป็นเครื่องมือที่มีประสิทธิภาพในการระบุปัญหาการเข้าถึงเว็บไซต์อย่างรวดเร็ว โดยช่วยให้สามารถจัดลำดับความสำคัญของปัญหาและวางแผนการแก้ไขได้อย่างมีระบบ เครื่องมือช่วยให้ทีมพัฒนาสามารถปรับปรุงหน้าเว็บให้สอดคล้องกับมาตรฐาน WCAG 2.1 ได้อย่างมีประสิทธิภาพและลดความเสี่ยงที่ผู้ใช้กลุ่มมีความบกพร่องทางการเข้าถึงจะไม่สามารถใช้งานระบบได้

2. axe DevTools เป็นเครื่องมือสำหรับนักพัฒนาที่ช่วยตรวจสอบ DOM structure, ARIA roles, semantic HTML, และ interactive elements โดยรายงาน violations พร้อมระดับความรุนแรง (Critical, Serious, Moderate, Minor) และให้คำแนะนำการแก้ไขเชิงเทคนิค เครื่องมือนี้เหมาะสำหรับตรวจสอบปัญหาเชิงลึกในระดับองค์ประกอบของหน้าเว็บ

การแสดงผลและรายงานของ axe DevTools จะแสดงผลในรูปแบบ Violation Report บน DevTools ของเบราว์เซอร์ โดยจะแยกตามองค์ประกอบ, impact, และคำแนะนำในการแก้ไข เช่น ปัญหา color contrast จะแสดงค่า contrast ratio ปัจจุบัน และค่าที่ควรเป็นเพื่อให้เป็นไปตามมาตรฐาน WCAG 2.1 AA นอกจากนี้ยังสามารถดาวน์โหลดผลการสแกนเป็น ไฟล์ JSON เพื่อนำไปวิเคราะห์เพิ่มเติมหรือใช้ในกระบวนการทดสอบอัตโนมัติ (Automation Testing) ได้

3. WAVE Evaluation Tool เป็นเครื่องมือที่ใช้ตรวจสอบความสามารถในการเข้าถึงเว็บไซต์ตามมาตรฐาน Web Content Accessibility Guidelines (WCAG) 2.1 โดยทำงานในลักษณะของการสแกนโครงสร้าง HTML ของหน้าเว็บและระบุปัญหาที่อาจส่งผลกระทบต่อผู้ใช้งานที่มีความต้องการพิเศษ เช่น ผู้พิการทางสายตา ผู้ใช้เครื่องช่วยอ่านหน้าจอ หรือผู้ที่มีปัญหาในการรับรู้สีและความคมชัดเครื่องมือนี้สามารถแสดงผลออกมาเป็นกราฟิกบนหน้าเว็บ พร้อมสัญลักษณ์และข้อความอธิบายประเภทของปัญหา ทำให้สามารถเข้าใจได้ง่าย ว่ามีองค์ประกอบใดที่ละเมิดหลักการเข้าถึงข้อมูล และมีระดับความรุนแรง (Severity) ของแต่ละปัญหา การแสดงผลและรายงานจากการวิเคราะห์ด้วย WAVE จะแบ่งออกเป็นหลายประเภทหลัก ได้แก่

1. Errors (ข้อผิดพลาด) เป็นปัญหาที่ส่งผลโดยตรงต่อความสามารถในการเข้าถึงเว็บไซต์ เช่น การใช้สีที่มีความแตกต่างต่ำเกินไปสำหรับผู้มีปัญหาด้านการมองเห็น การขาดข้อความแทน (alt text) สำหรับรูปภาพ หรือการใช้ heading ที่ไม่ถูกต้อง

2. Alerts (คำเตือน) เป็นข้อสังเกตที่อาจสร้างความยากลำบากในการใช้งานสำหรับบางกลุ่มผู้ใช้ แต่ไม่ถือเป็นข้อผิดพลาดโดยตรง เช่น การใช้ลิงก์ซ้ำหลายครั้งโดยไม่ชี้ชัดถึงเนื้อหา หรือข้อความที่อาจไม่ชัดเจนสำหรับผู้ใช้บางกลุ่ม

3. Features (ฟีเจอร์ที่ช่วยในการเข้าถึง) WAVE จะแสดงองค์ประกอบที่ช่วยให้ผู้ใช้เข้าถึงเนื้อหาได้ง่ายขึ้น เช่น การใช้ heading ที่เหมาะสม การใช้ label กับฟอร์ม และการจัดลำดับ tab index ที่ถูกต้อง

4. Structural Elements (โครงสร้างของหน้าเว็บ) เป็นการตรวจสอบโครงสร้างของหน้า เช่น heading hierarchy, lists, tables ซึ่งมีผลต่อความเข้าใจและการนำทางของผู้ใช้ โดยเฉพาะผู้ใช้ screen reader

5. Contrast Errors (ข้อผิดพลาดเรื่องความแตกต่างของสี) เป็นการตรวจสอบว่าองค์ประกอบข้อความและพื้นหลังมี contrast เพียงพอต่อการอ่านตามมาตรฐาน WCAG AA หรือไม่

การใช้ WAVE Evaluation Tool เป็นเครื่องมือที่ช่วยให้ผู้พัฒนาเว็บไซต์สามารถระบุปัญหา Accessibility ได้อย่างรวดเร็วและมีประสิทธิภาพ โดยผลการวิเคราะห์จะช่วยให้สามารถปรับปรุงเว็บไซต์ให้ตอบสนองต่อผู้ใช้ทุกกลุ่มได้ดียิ่งขึ้น ซึ่งเป็นส่วนสำคัญของการออกแบบเว็บไซต์ที่ครอบคลุม (inclusive design) และสอดคล้องกับมาตรฐานสากล WCAG

### 5.6.1 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Dashboard

การตรวจสอบความสามารถในการเข้าถึงของหน้า Dashboard ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools พบว่ามีประเด็นที่ควรปรับปรุงหลายประการ แม้ว่าหน้าดังกล่าวจะไม่ปรากฏปัญหาในระดับ *Critical* แต่ยังคงมีประเด็นในระดับ *High* และ *Medium* ซึ่งอาจส่งผลกระทบต่อประสบการณ์การใช้งานของผู้ใช้ที่มีความบกพร่องทางสายตา

#### 1. ปัญหาเกี่ยวกับความแตกต่างของสี (Color Contrast)

ทั้งสองเครื่องมือระบุว่าความแตกต่างของสีระหว่างข้อความกับพื้นหลังไม่เป็นไปตามเกณฑ์ขั้นต่ำตามมาตรฐาน WCAG 2.1 AA (Contrast Ratio  $\geq 4.5:1$ ) ซึ่งอาจส่งผลกระทบต่อผู้ใช้ที่มีความบกพร่องทางการมองเห็น เช่น ผู้ที่มีภาวะตาบอดสีหรือการมองเห็นลดลง ทำให้ไม่สามารถรับรู้ข้อความหรือองค์ประกอบสำคัญบนหน้าจอได้อย่างชัดเจน

#### 2. ปัญหาเกี่ยวกับโครงสร้างเนื้อหา (Heading Structure)

การวิเคราะห์โดย Includia Accessibility Checker พบว่าหน้า Dashboard ขาดหัวข้อหลักในรูปแบบ Heading ระดับ 1 (H1) ซึ่งเป็นองค์ประกอบสำคัญในการสร้างโครงสร้างเนื้อหาเชิงลำดับชั้น การขาด H1 ทำให้ผู้ใช้ที่พึ่งพา *Screen Reader* ไม่สามารถเข้าใจโครงสร้างเนื้อหาหลักของหน้าได้ ส่งผลต่อประสิทธิภาพในการนำทางและการทำความเข้าใจเนื้อหาโดยรวม

#### 3. ปัญหาเกี่ยวกับองค์ประกอบภาพ (Image Elements)

การวิเคราะห์โดย axe DevTools ระบุว่าองค์ประกอบภาพ (`role="img"`) ขาด `attribute alt` ซึ่งส่งผลให้ผู้ใช้ที่พึ่งพาเทคโนโลยีช่วยเหลือ เช่น *Screen Reader* ไม่สามารถรับรู้ข้อมูลเชิงภาพได้ครบถ้วน ทำให้การตีความข้อมูลหรือบริบทเชิงภาพมีข้อจำกัด

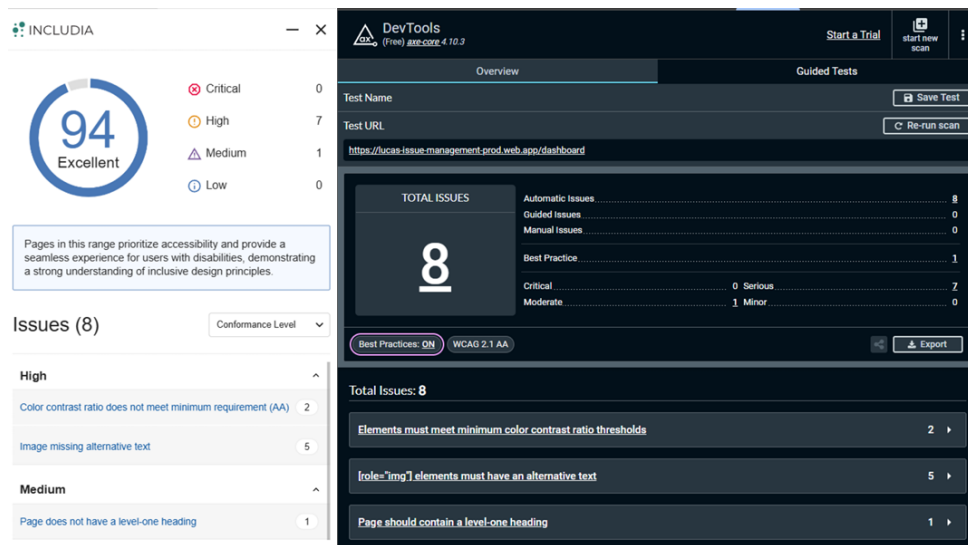
#### 4. สรุปเชิงวิเคราะห์

จาก

การตรวจสอบด้วยทั้งสองเครื่องมือ สามารถสรุปได้ว่าหน้า Dashboard มีปัญหาหลักเกี่ยวข้องกับความสามารถในการมองเห็นและการเข้าถึงเนื้อหา โดยแนวทางการปรับปรุงที่แนะนำ ได้แก่

1. ปรับปรุงอัตราส่วนความคมชัดของสีให้สอดคล้องกับมาตรฐาน WCAG
2. กำหนด Heading ระดับ 1 (H1) เพื่อสร้างโครงสร้างเนื้อหาเชิงลำดับชั้นที่ชัดเจน
3. เพิ่ม `attribute alt` ให้กับองค์ประกอบภาพเพื่อรองรับผู้ใช้เทคโนโลยีช่วยเหลือ

การแก้ไขปัญหาดังกล่าวนี้จะช่วยเพิ่มความเท่าเทียมในการเข้าถึงและประสบการณ์การใช้งานของผู้ใช้ทุกกลุ่ม โดยเฉพาะผู้ที่มีความบกพร่องทางสายตาและผู้ใช้ที่พึ่งพาอุปกรณ์ช่วยเหลือในการเข้าถึงข้อมูล



รูปที่ 5.14 การวิเคราะห์ความสามารถในการเข้าถึง ของหน้า Dashboard ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools

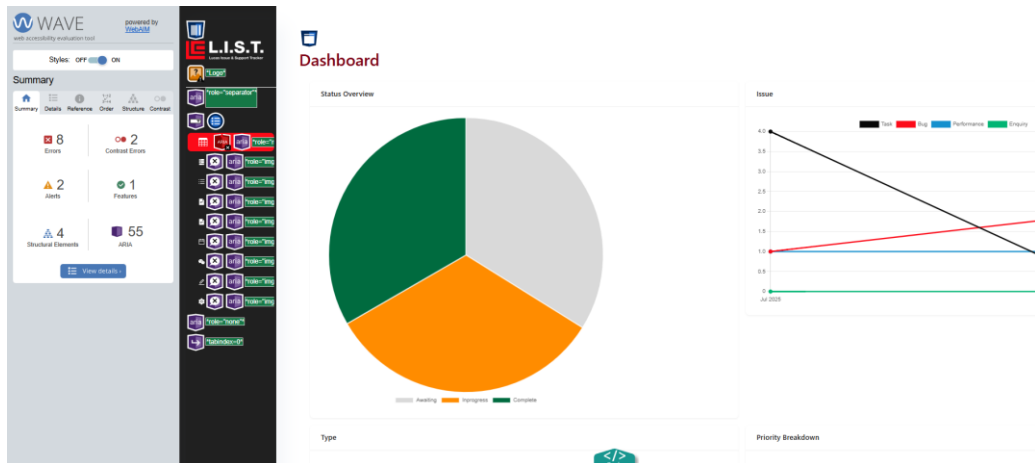
ผลการตรวจสอบของ WAVE แสดงให้เห็นว่าหน้านี้มีข้อผิดพลาดหลักๆ ได้แก่

- มีปุ่มและไอคอนที่ขาดคำอธิบาย (เช่น ไอคอนค้นหา)
- พบข้อความและกราฟที่มีความเปรียบต่างของสี (contrast) ต่ำ
- การจัดเรียงหัวข้อ (heading structure) ไม่สอดคล้องกับลำดับเชิงตรรกะ

การขาดคำอธิบายในปุ่มหรือไอคอน ส่งผลให้ผู้ใช้งานที่อาศัย screen reader ไม่สามารถเข้าใจการทำงานได้ ถือว่าขัดกับ WCAG 1.1.1 (Non-text Content) ซึ่งกำหนดว่าทุกองค์ประกอบที่ไม่ใช่ข้อความ ต้องมีทางเลือกในรูปแบบข้อความ (text alternative) ในด้านของสีที่มี contrast ต่ำ จะกระทบผู้ใช้งานที่มีปัญหาการมองเห็นหรือผู้สูงอายุ ไม่เป็นไปตาม WCAG 1.4.3 (Contrast - Minimum) ที่ระบุว่าข้อความควรมีอัตราส่วนความเปรียบต่างอย่างน้อย 4.5:1 ส่วน heading structure ที่ไม่ถูกต้องจะทำให้ผู้อ่านผ่าน screen reader เข้าใจลำดับเนื้อหาผิดพลาด ซึ่งขัดกับ WCAG 1.3.1 (Info and Relationships) ข้อเสนอแนะ

ข้อเสนอแนะ

1. เพิ่ม aria-label หรือคำอธิบายข้อความกำกับทุกปุ่มและไอคอน
2. ปรับสีข้อความและพื้นหลังให้มี contrast ตามเกณฑ์มาตรฐาน
3. จัดลำดับหัวข้อใหม่ให้เป็นไปตามลำดับเชิงตรรกะ (H1 → H2 → H3)



รูปที่ 5.15 การวิเคราะห์ความสามารถในการเข้าถึง ของหน้า Dashboard ด้วยเครื่องมือ WAVE

5.6.2 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้าการเพิ่มข้อมูลของโครงการ

จากการตรวจสอบความสามารถในการเข้าถึงของหน้า การเพิ่มข้อมูลของโครงการ ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools พบว่าหน้าเว็บดังกล่าวมีข้อจำกัดด้านการมองเห็นและการเข้าถึงเนื้อหา แม้ว่าจะไม่พบปัญหา *Critical* แต่ยังคงมีประเด็นระดับ *High (Serious)* ที่ควรแก้ไขเพื่อยกระดับความสามารถในการเข้าถึง

#### 1. ปัญหาเกี่ยวกับความแตกต่างของสี (Color Contrast)

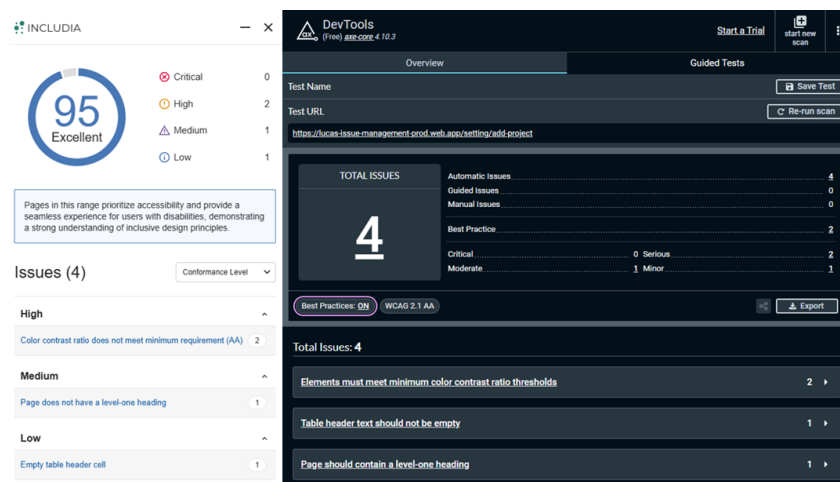
ทั้งสองเครื่องมือตรวจพบว่าการกำหนดสีของข้อความและพื้นหลังไม่สอดคล้องกับเกณฑ์มาตรฐาน WCAG 2.1 AA การเลือกสีที่เหมาะสมและมี *Contrast Ratio* เพียงพอ จะช่วยให้ผู้ใช้สามารถระบุข้อความและโต้ตอบกับองค์ประกอบฟอร์มได้อย่างชัดเจน ลดความสับสน และเพิ่มประสิทธิภาพการใช้งาน

#### 2. แนวทางแก้ไขและการยืนยันผล

1. ปรับอัตราส่วนความคมชัดของสีในข้อความและองค์ประกอบ UI ให้สอดคล้องกับมาตรฐาน WCAG 2.1 AA
2. ประเมินซ้ำด้วยเครื่องมือ *accessibility testing* เช่น axe DevTools และ Includia Accessibility Checker เพื่อตรวจสอบประสิทธิภาพของการแก้ไข
3. การดำเนินการดังกล่าวจะช่วยยืนยันว่าหน้าเว็บสามารถเข้าถึงได้สำหรับผู้ใช้ทุกกลุ่ม

### 3. สรุปเชิงวิเคราะห์

การปรับปรุงความแตกต่างของสีและการตรวจสอบซ้ำด้วยเครื่องมือทดสอบความสามารถในการเข้าถึง จะช่วยเพิ่มความเท่าเทียมและประสิทธิภาพในการเข้าถึงข้อมูลของผู้ใช้ทุกกลุ่ม พร้อมสร้างความมั่นใจว่าองค์ประกอบฟอร์มและข้อความบนหน้าเว็บสามารถรับรู้และใช้งานได้อย่างครบถ้วน



รูปที่ 5.16 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้าการเพิ่มข้อมูลของโครงการ ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools

ผลการตรวจสอบของ WAVE แสดงให้เห็นว่าหน้านี้มีข้อผิดพลาดหลักๆ โดยเฉพาะในส่วนของฟอร์มและตาราง เช่น

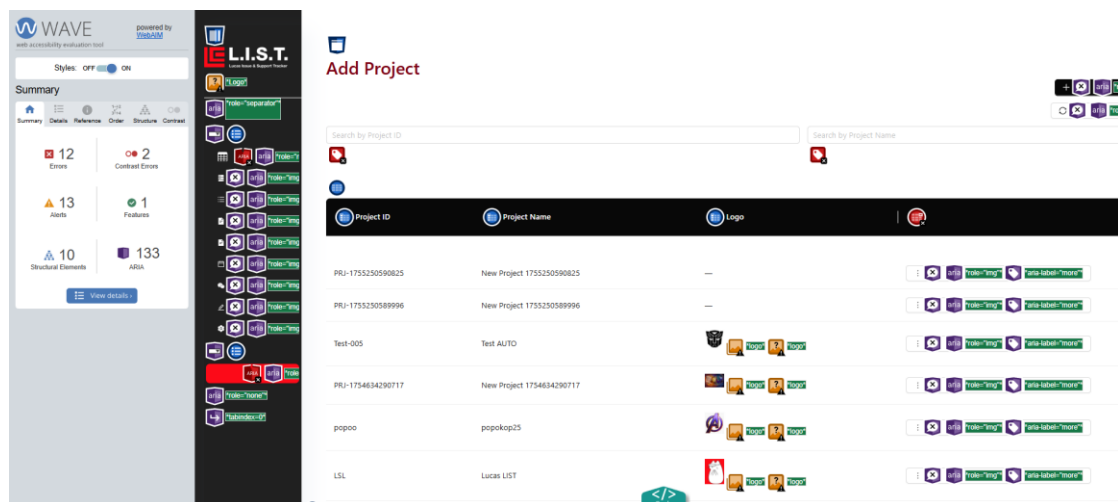
- ช่องกรอกข้อความ (input) ไม่มี label เชื่อมโยง
- ปุ่มที่ใช้เฉพาะไอคอนโดยไม่มีข้อความกำกับ
- ข้อความลิงก์ซ้ำ เช่น “Edit” หรือ “Delete” โดยไม่บอกว่าหมายถึงโปรเจกต์ใด

ฟอร์มเป็นองค์ประกอบสำคัญของระบบใด ๆ หากไม่มีการกำหนด label อย่างชัดเจน จะส่งผลให้ผู้ใช้ที่ใช้ screen reader ไม่สามารถทราบได้ว่าช่องนั้นต้องกรอกข้อมูลอะไร ซึ่งขัดกับ WCAG 3.3.2 (Labels or Instructions) นอกจากนี้ ลิงก์หรือปุ่มที่ซ้ำกัน เช่น ปุ่ม “Edit” หลายปุ่มในตาราง โดยไม่มีข้อความบอกว่าเป็นการแก้ไขของโปรเจกต์ใด จะทำให้ผู้ใช้ที่มีข้อจำกัดไม่สามารถใช้งานได้สะดวก ถือว่าไม่สอดคล้องกับ WCAG 2.4.4 (Link Purpose)



## ข้อเสนอแนะ

1. เพิ่ม <label> ให้กับทุกช่องกรอกข้อมูล
2. ระบุคำอธิบายเพิ่มเติมในปุ่ม เช่น “Edit Project Test Auto” แทนที่จะใช้เพียง “Edit”
3. ปรับสีข้อความในส่วนที่ contrast ต่ำให้สอดคล้องกับมาตรฐาน



รูปที่ 5.17 การวิเคราะห์ความสามารถในการเข้าถึง ของหน้า การเพิ่มข้อมูลของโครงการ ด้วยเครื่องมือ WAVE

### 5.6.3 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Other Document

จากการตรวจสอบความสามารถในการเข้าถึงของหน้า Other Document ในขั้นตอนเพิ่มเอกสารสัญญาและเอกสารที่เกี่ยวข้อง ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools พบว่าหน้าเว็บดังกล่าวมีข้อจำกัดที่สำคัญต่อความสามารถในการเข้าถึงเนื้อหาและองค์ประกอบ UI แม้ว่าหน้าเว็บจะไม่พบปัญหา *Critical* ในระดับ ARIA หรือโครงสร้างฟอร์ม แต่ยังคงมีประเด็นด้านการมองเห็นและการรับรู้ข้อมูลที่ต้องปรับปรุง

1. ปัญหาเกี่ยวกับความแตกต่างของสี (Color Contrast)

ทั้งสองเครื่องมือตรวจพบว่าการใช้สีของข้อความ ปุ่ม และไอคอนไม่สอดคล้องกับเกณฑ์ความคมชัดขั้นต่ำตามมาตรฐาน WCAG 2.1 AA ซึ่งจัดอยู่ในระดับ *High* การปรับปรุงสีให้มี *Contrast Ratio* สูงขึ้นจะช่วยให้ผู้ใช้ที่มีความบกพร่องทางสายตา เช่น ผู้ที่มีภาวะตาบอดสีหรือการมองเห็นลดลง สามารถรับรู้และใช้งานองค์ประกอบ UI ได้อย่างเต็มประสิทธิภาพ

## 2. ปัญหาเกี่ยวกับโครงสร้างเนื้อหา (Heading Structure)

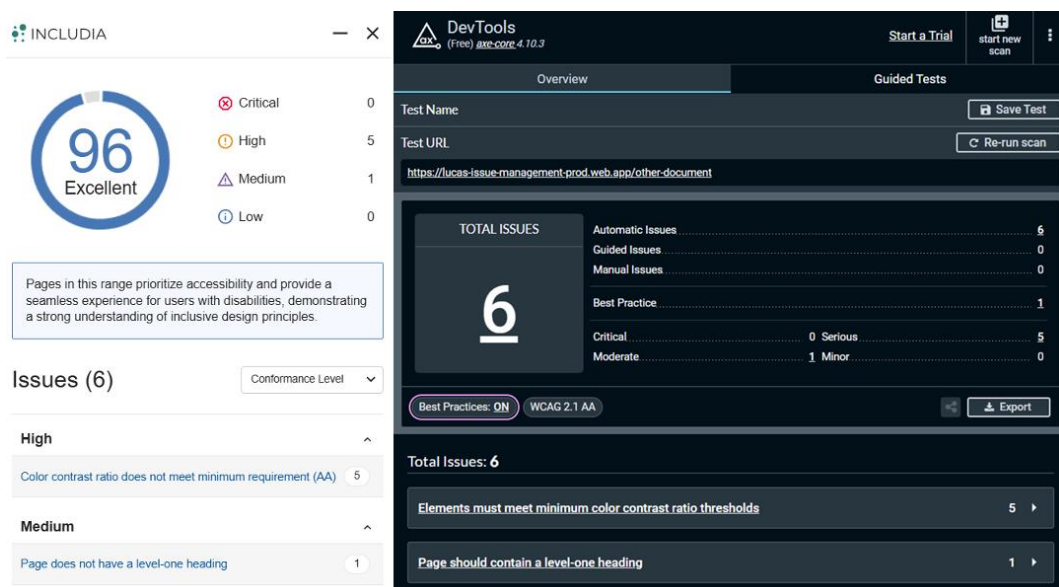
Includia Accessibility Checker ระบุว่าหน้า Other Document ขาดหัวข้อหลักในระดับ H1 ซึ่งจัดอยู่ในระดับ *Medium* การขาดหัวข้อหลักอาจสร้างอุปสรรคต่อผู้ใช้งานที่พึ่งพา *Screen Reader* ทำให้ไม่สามารถรับรู้สาระสำคัญของเนื้อหาได้ครบถ้วน

## 3. สรุปเชิงวิเคราะห์

หน้า Other Document มีปัญหาหลักเกี่ยวข้องกับความสามารถในการมองเห็น และการเข้าถึงองค์ประกอบ UI แนวทางการปรับปรุงที่แนะนำ ได้แก่:

1. ปรับอัตราส่วนความคมชัดของสีในข้อความ ปุ่ม และไอคอนให้สอดคล้องกับมาตรฐาน WCAG 2.1 AA
2. ตรวจสอบและแก้ไของค์ประกอบ UI ทุกตัวเพื่อให้ผู้ใช้ที่มีความบกพร่องทางสายตาสามารถใช้งานได้เต็มประสิทธิภาพ
3. กำหนด Heading ระดับ 1 (H1) อย่างชัดเจนเพื่อสร้างโครงสร้างเนื้อหาเชิงลำดับชั้น

การดำเนินการตามแนวทางเหล่านี้จะช่วยเพิ่มความเท่าเทียมและประสิทธิภาพในการเข้าถึงข้อมูลของผู้ใช้ทุกกลุ่ม โดยเฉพาะผู้ที่มีข้อจำกัดทางสายตา และสอดคล้องกับหลักการ *Accessible Design*



รูปที่ 5.18 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Other Document ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools

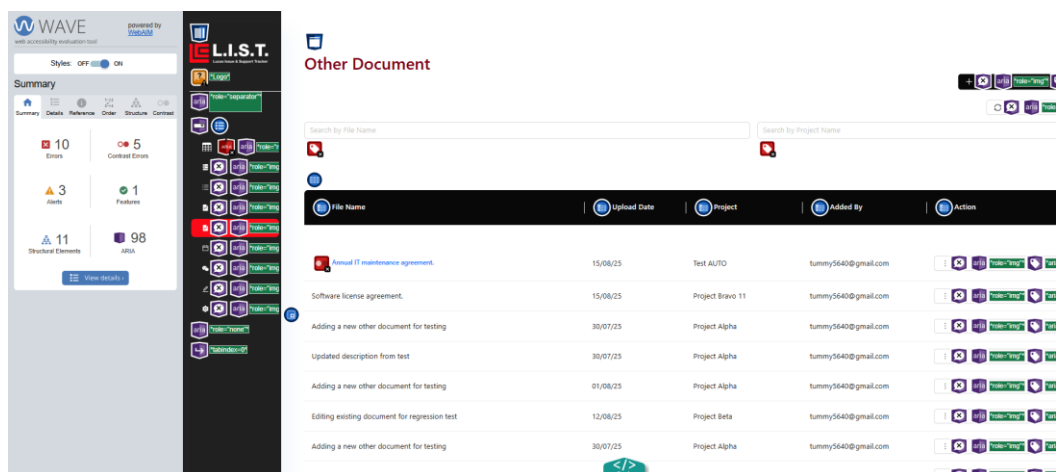
ผลการตรวจสอบของ WAVE แสดงให้เห็นว่าหน้านี้มีข้อผิดพลาดหลักๆ ได้แก่

- ช่องค้นหาไม่มี label
- ปุ่ม action เช่น “Edit” และ “Delete” ไม่มีคำอธิบายเฉพาะเจาะจง
- ข้อความลิงก์บางส่วนมี contrast ต่ำ
- ตารางเอกสารไม่มีการระบุ caption หรือ scope ที่ชัดเจน

ตารางเป็นโครงสร้างข้อมูลที่ซับซ้อน หากไม่มีการระบุ caption หรือ scope ที่สัมพันธ์ระหว่างคอลัมน์และแถว จะทำให้ผู้ใช้ screen reader ไม่สามารถเข้าใจข้อมูลได้ครบถ้วน ขัดกับ WCAG 1.3.1 (Info and Relationships) การขาด label ในช่องค้นหาที่เป็นอุปสรรคต่อผู้ใช้งานที่ไม่สามารถมองเห็นได้ เนื่องจากไม่รู้ว่าจะกรอกนี้มีวัตถุประสงค์อะไร ในขณะเดียวกัน contrast ต่ำก็เป็นอุปสรรคสำหรับผู้ที่มีสายตาเลือนราง

ข้อเสนอแนะ

1. เพิ่ม label ให้ช่องค้นหาอย่างชัดเจน
2. ปรับข้อความในปุ่ม action ให้เฉพาะเจาะจง เช่น “Delete document: Software License Agreement”
3. ปรับสีข้อความและพื้นหลังให้ตรงตามมาตรฐาน contrast
4. เพิ่ม <caption> และกำหนด scope="col" หรือ scope="row" ในตาราง เพื่อช่วยให้ screen reader เข้าใจความสัมพันธ์ของข้อมูลได้ชัดเจน



รูปที่ 5.19 การวิเคราะห์ความสามารถในการเข้าถึง ของหน้า Other Document ด้วยเครื่องมือ WAVE

#### 5.6.4 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Scope

จากการตรวจสอบความสามารถในการเข้าถึงของหน้า Scope ด้วยเครื่องมือ IncludiaAccessibility Checker และ axe DevTools พบว่าหน้าเว็บดังกล่าวมีปัญหาที่เกี่ยวข้องกับความสามารถในการเข้าถึงเนื้อหาและการรับรู้ข้อมูล แม้ว่าหน้าเว็บจะไม่พบปัญหา *Critical* เกี่ยวกับ ARIA หรือฟอร์มขาด label แต่ยังคงมีข้อจำกัดที่สำคัญต่อผู้ใช้ที่มีความบกพร่องทางสายตา

##### 1. ปัญหาเกี่ยวกับความแตกต่างของสี (Color Contrast)

ทั้งสองเครื่องมือตรวจพบว่าความแตกต่างของสีระหว่างข้อความหรือองค์ประกอบ UI กับพื้นหลังไม่เป็นไปตามเกณฑ์ขั้นต่ำของมาตรฐาน WCAG 2.1 AA การกำหนดสีที่ไม่เหมาะสมอาจส่งผลให้ผู้ใช้ที่มีความบกพร่องทางสายตา เช่น ผู้ที่มีภาวะตาบอดสีหรือมีการมองเห็นลดลง ไม่สามารถแยกแยะข้อความหรือองค์ประกอบสำคัญได้อย่างชัดเจน ส่งผลต่อประสิทธิภาพในการรับรู้ข้อมูลและการใช้งานโดยรวม

##### 2. ปัญหาเกี่ยวกับโครงสร้างเนื้อหา (Heading Structure)

การตรวจสอบโดย Includia Accessibility Checker ระบุว่าหน้า Scope ขาดหัวข้อหลักในระดับ H1 ซึ่งจัดอยู่ในระดับ *Medium* การขาดหัวข้อหลักอาจสร้างอุปสรรคต่อผู้ใช้งานที่พึ่งพา *Screen Reader* ทำให้ไม่สามารถเข้าถึงสาระสำคัญของเนื้อหาได้ครบถ้วน

##### 3. ปัญหาเกี่ยวกับตารางข้อมูล (Table Structure)

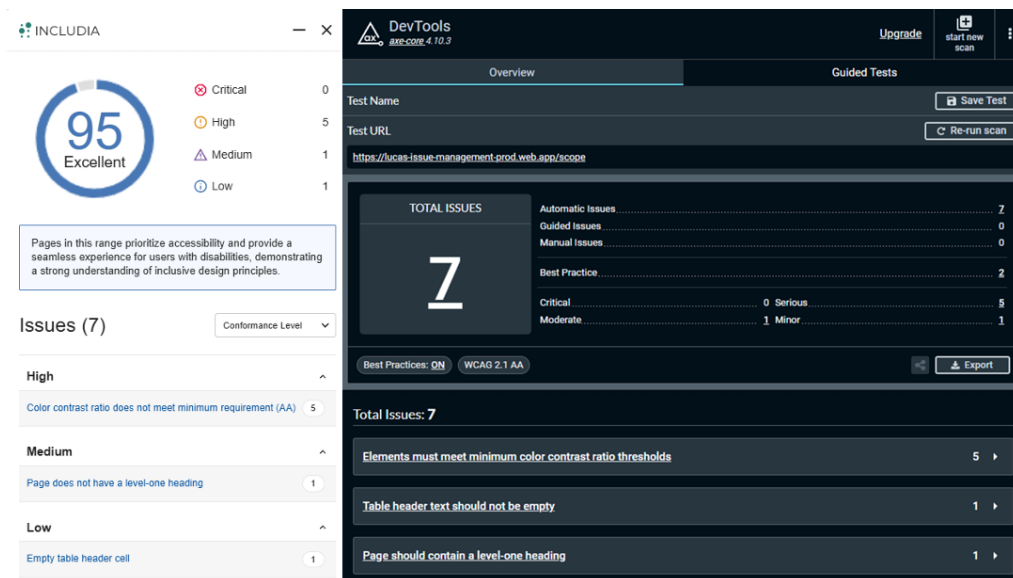
Includia Accessibility Checker พบปัญหาในระดับ *Low* คือ การเว้นช่องว่างในเซลล์หัวตาราง (*Empty Table Header Cell*) ซึ่งอาจทำให้การอ่านข้อมูลด้วย *Screen Reader* มีความไม่สมบูรณ์ ส่งผลต่อประสบการณ์การใช้งานของผู้พิการทางสายตา

##### 4. สรุปเชิงวิเคราะห์

หน้า Other Document มีปัญหาหลักเกี่ยวข้องกับการมองเห็นและการเข้าถึงองค์ประกอบ UI แนวทางการปรับปรุงที่แนะนำ ได้แก่:

1. ปรับอัตราส่วนความคมชัดของสีให้สอดคล้องกับมาตรฐาน WCAG 2.1 AA
2. กำหนด Heading ระดับ 1 (H1) อย่างชัดเจนเพื่อสร้างโครงสร้างเนื้อหาเชิงลำดับชั้น
3. เติมข้อมูลในเซลล์หัวตารางที่ว่างเพื่อให้ *Screen Reader* สามารถอ่านข้อมูลได้ครบถ้วนการ

การดำเนินการตามแนวทางเหล่านี้จะช่วยเพิ่มความเท่าเทียมและประสิทธิภาพในการเข้าถึงข้อมูลของผู้ใช้ทุกกลุ่ม โดยเฉพาะผู้ที่มีข้อจำกัดทางสายตา



รูปที่ 5.20 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Scope of Work ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools

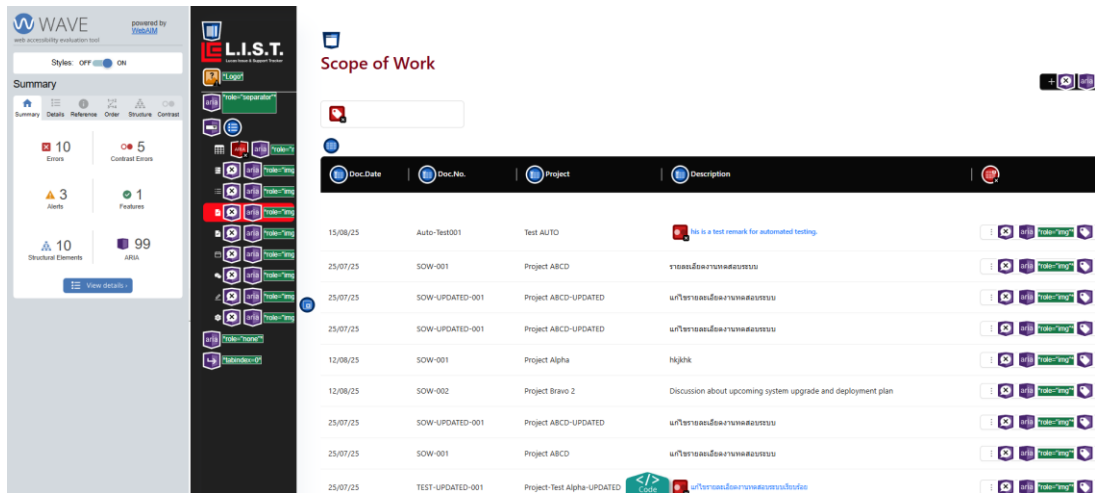
การตรวจสอบหน้า Scope of Work ด้วยเครื่องมือ WAVE พบปัญหาหลักดังนี้:

- ตารางไม่มีการเชื่อมโยงระหว่างหัวตารางและข้อมูลในแต่ละเซลล์
- ข้อความบางส่วนมีค่า contrast ต่ำ

หน้า Scope of Work เป็นหน้าที่ใช้ในการจัดการรายละเอียดและบันทึกขอบเขตงานของโครงการ การขาด <th> หรือ scope ทำให้ผู้ใช้งานที่อาศัย screen reader ไม่สามารถระบุความสัมพันธ์ระหว่างหัวตารางและข้อมูลในแต่ละเซลล์ได้ นอกจากนี้ ข้อความที่มี contrast ต่ำอาจทำให้ผู้ที่มีสายตาเลือนรางอ่านเนื้อหาได้ยาก

ข้อเสนอแนะ

1. เพิ่ม <th> และใช้ scope ให้สัมพันธ์กับ <td>
2. ปรับค่า contrast ของข้อความให้เป็นไปตามมาตรฐาน WCAG



รูปที่ 5.21 การวิเคราะห์ความสามารถในการเข้าถึง ของหน้า Scope of Work ด้วยเครื่องมือ WAVE

#### 5.6.5 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Projects

จากการตรวจสอบความสามารถในการเข้าถึงของหน้า Projects ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools พบว่าหน้าเว็บมีประเด็นที่ควรปรับปรุงหลายประการ แม้ว่าการตรวจสอบโดย Includia Accessibility Checker จะไม่พบปัญหาในระดับ *Critical* แต่การวิเคราะห์โดย axe DevTools ระบุปัญหาระดับสูงสุด (*Critical*) ซึ่งสะท้อนถึงความจำเป็นในการปรับปรุงการออกแบบ UI ให้สอดคล้องกับมาตรฐานความสามารถในการเข้าถึง

##### 1. ปัญหาเกี่ยวกับความแตกต่างของสี (Color Contrast)

ทั้งสองเครื่องมือตรวจพบว่าความแตกต่างของสีระหว่างข้อความหรือองค์ประกอบ UI กับพื้นหลังไม่เป็นไปตามเกณฑ์ขั้นต่ำของมาตรฐาน WCAG 2.1 AA ส่งผลให้ผู้ใช้ที่มีความบกพร่องทางสายตา เช่น ผู้ที่มีภาวะตาบอดสีหรือมีการมองเห็นจำกัด ไม่สามารถอ่านหรือระบุข้อมูลบนหน้าจอได้อย่างชัดเจน

##### 2. ปัญหาเกี่ยวกับ ARIA Roles และ Attributes

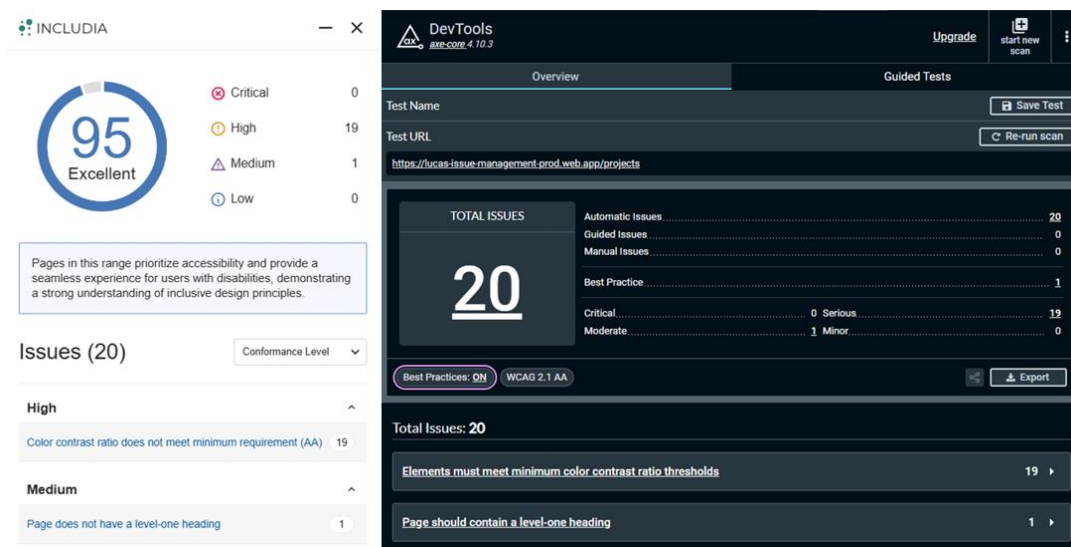
การตรวจสอบหน้า Add Issue โดย axe DevTools ระบุว่าการกำหนด ARIA Roles และ Attributes ไม่สอดคล้องกับองค์ประกอบ (*Non-conforming ARIA*) ซึ่งถือเป็นปัญหา *Critical* ทำให้ผู้ใช้ที่ใช้ *Screen Reader* หรือเทคโนโลยีช่วยเหลืออาจตีความ element ผิดพลาด ปัญหาดังกล่าวสะท้อนถึงการออกแบบ UI ที่ไม่สอดคล้องกับหลักการ Accessible Rich Internet Applications (WAI-ARIA) และมาตรฐาน WCAG 2.1

### 3. สรุปเชิงวิเคราะห์

แม้ว่าหน้า Projects จะไม่พบปัญหา Critical ทุกส่วน แต่ประเด็นระดับ High และ Critical ที่ตรวจพบชี้ให้เห็นข้อจำกัดด้านความสามารถในการมองเห็น, โครงสร้างเนื้อหา, และการสื่อสารผ่าน ARIA แนวทางการปรับปรุงที่แนะนำ ได้แก่

1. ปรับอัตราส่วนความคมชัดของสีให้สอดคล้องกับมาตรฐาน WCAG 2.1 AA
2. กำหนด Heading ระดับ 1 (H1) อย่างชัดเจนเพื่อสร้างโครงสร้างเนื้อหาเชิงลำดับชั้น
3. ตรวจสอบและแก้ไขการกำหนด ARIA Roles และ Attributes ให้สอดคล้องกับองค์ประกอบและหลัก WAI-ARIA

การดำเนินการตามแนวทางเหล่านี้จะช่วยเพิ่มความเท่าเทียมและประสิทธิภาพในการเข้าถึงข้อมูลของผู้ใช้ทุกกลุ่ม โดยเฉพาะผู้ที่มีข้อจำกัดทางสายตา



รูปที่ 5.22 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Projects ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools

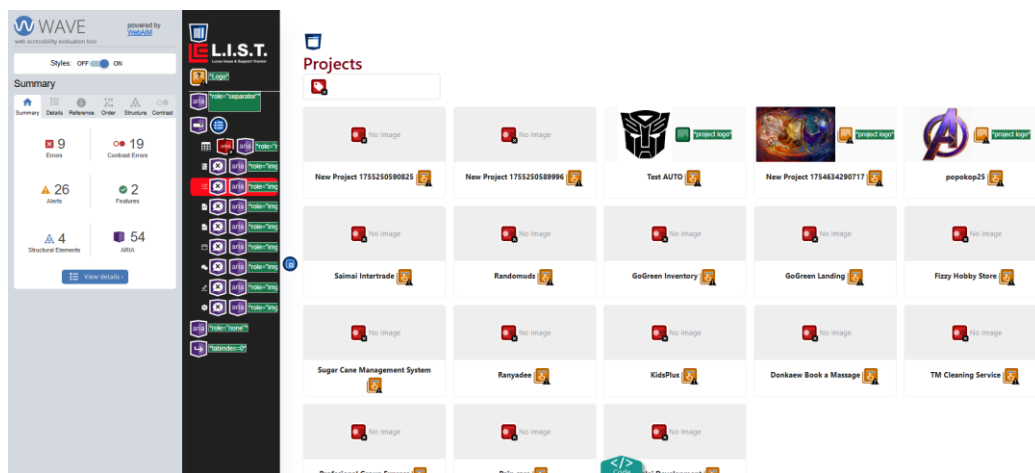
หน้า Projects Page เป็นหน้าที่ใช้ในการแสดงรายการโครงการ พบปัญหาดังนี้:

- ภาพโครงการและโลโก้ไม่มี alt text กำกับ
- Contrast ของข้อความชื่อโครงการบางส่วนต่ำ

การไม่มี alt text ทำให้ผู้ใช้งานที่ใช้ screen reader ไม่สามารถรับรู้ข้อมูลภาพได้ ข้อความป้อนซ้ำโดยไม่มีบริบททำให้ผู้ใช้งานสับสน ข้อความที่ contrast ต่ำทำให้ผู้มีปัญหาทางสายตาด่านเนื้อหาได้ยาก ซึ่งไม่สอดคล้องกับมาตรฐาน WCAG 1.1.1 และ 2.4.4

ข้อเสนอแนะ

1. เพิ่ม alt text ให้กับทุกภาพ เช่น alt="Project Avengers Logo"
2. ปรับสีข้อความให้ contrast ผ่านเกณฑ์ขั้นต่ำ



รูปที่ 5.23 การวิเคราะห์ความสามารถในการเข้าถึง ของหน้า Projects ด้วยเครื่องมือ WAVE

#### 5.6.6 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Project Detail

จากการตรวจสอบความสามารถในการเข้าถึงของหน้า Project Detail ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools พบว่าหน้าเว็บมีคะแนนรวมในระดับ Excellent (95 คะแนน) สะท้อนว่ามีการออกแบบที่รองรับผู้ใช้งานทั่วไปและผู้ใช้ที่มีข้อจำกัดได้ดี อย่างไรก็ตาม การตรวจสอบโดย axe DevTools ระบุปัญหาจำนวน 4 ประเด็น ซึ่งมีทั้งระดับ Serious, Moderate และ Minor ที่ควรได้รับการแก้ไขเพื่อให้เป็นไปตามมาตรฐาน WCAG 2.1 ระดับ AA อย่างสมบูรณ์



### 1. ปัญหาเกี่ยวกับความแตกต่างของสี (Color Contrast)

ตรวจพบว่าองค์ประกอบข้อความและพื้นหลังบางส่วนมีค่าความแตกต่างของสี (Contrast Ratio) ไม่ถึงเกณฑ์ขั้นต่ำที่กำหนดในมาตรฐาน WCAG 2.1 AA (4.5:1 สำหรับข้อความปกติ และ 3:1 สำหรับข้อความขนาดใหญ่) ส่งผลให้ผู้ใช้ที่มีข้อจำกัดด้านการมองเห็น เช่น ผู้ที่สายตาสั้นหรืออยู่ในสภาวะแสงจ้า อาจไม่สามารถอ่านข้อความได้อย่างชัดเจน

### 2. ปัญหาเกี่ยวกับตาราง (Table Headers)

การตรวจสอบพบว่าตารางข้อมูลบางส่วนไม่มีการกำหนด Table Header ( <th> ) อย่างชัดเจน หรือมี Header ที่ว่างเปล่า ทำให้ผู้ใช้เทคโนโลยีช่วยเหลือ เช่น Screen Reader ไม่สามารถระบุความหมายของข้อมูลแต่ละคอลัมน์ได้ครบถ้วน ส่งผลกระทบต่อการรับรู้เชิงโครงสร้างของเนื้อหาในตาราง

### 3. ปัญหาเกี่ยวกับ Accessible Names ขององค์ประกอบ

องค์ประกอบบางรายการ เช่น ปุ่ม, ไอคอน หรือภาพ ไม่ได้ถูกกำหนด Accessible Name หรือข้อความที่สามารถอ่านได้โดย Screen Reader ส่งผลให้ผู้ใช้ไม่สามารถเข้าใจความหมายหรือการทำงานขององค์ประกอบนั้น ๆ ได้อย่างถูกต้อง

### 4. ปัญหาเกี่ยวกับโครงสร้างหัวเรื่อง (Heading Structure)

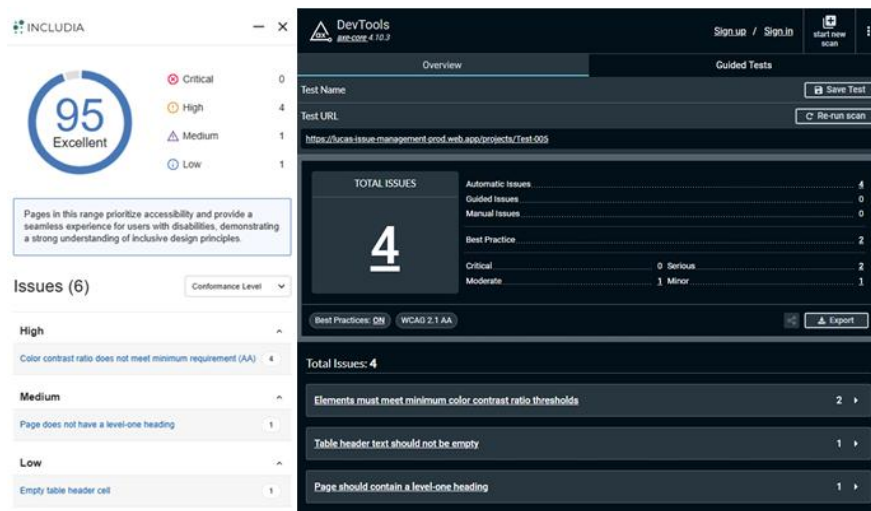
หน้า Project Detail ไม่มีการกำหนด Heading ระดับ 1 (H1) ซึ่งถือเป็นหัวข้อหลักของหน้า ทำให้โครงสร้างเชิงลำดับชั้นของเนื้อหาไม่สมบูรณ์ และอาจกระทบต่อการทำงานของ Screen Reader รวมถึงด้าน SEO

## 3. สรุปเชิงวิเคราะห์

แม้ว่า Includia Accessibility Checker จะให้คะแนนระดับ Excellent แต่การวิเคราะห์เชิงลึกโดย axe DevTools ชี้ให้เห็นประเด็นสำคัญด้านการรับรู้ (Perceivability) และการเข้าใจ (Understandability) ได้แก่ ปัญหาด้าน Contrast, Table Headers, Accessible Names และ Heading Structure แนวทางการปรับปรุงที่แนะนำ ได้แก่

1. ปรับอัตราส่วนความคมชัดของสี (Color Contrast) ให้ผ่านเกณฑ์ WCAG 2.1 AA
2. กำหนด Table Headers ให้ครบถ้วนและมีข้อความที่สื่อความหมาย
3. เพิ่ม Accessible Names ให้กับปุ่ม ไอคอน และภาพ
4. เพิ่ม Heading ระดับ 1 (H1) เพื่อสร้างโครงสร้างเนื้อหาเชิงลำดับชั้นที่สมบูรณ์

การดำเนินการตามแนวทางดังกล่าวจะช่วยให้น้ำ Project Detail มีความสอดคล้องกับมาตรฐาน WCAG 2.1 AA อย่างครบถ้วน และเพิ่มความเท่าเทียมในการเข้าถึงสำหรับผู้ใช้งานทุกกลุ่ม โดยเฉพาะผู้ที่มีข้อจำกัดด้านการมองเห็นและผู้พึ่งพาเทคโนโลยีช่วยเหลือ



รูปที่ 5.22 การวิเคราะห์ความสามารถในการเข้าถึง (Accessibility) ของหน้า Project Detail ด้วยเครื่องมือ Includia Accessibility Checker และ axe DevTools

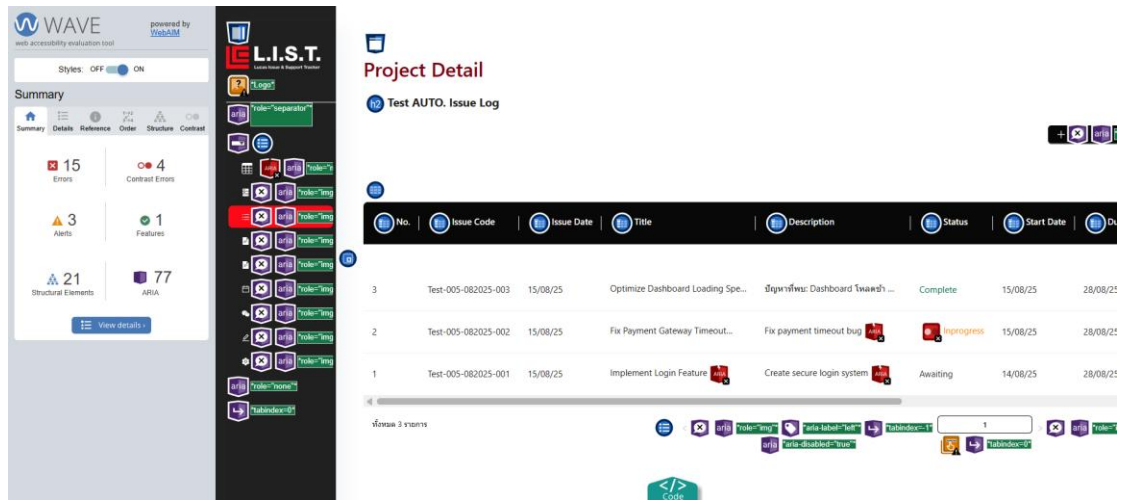
หน้า รายละเอียดโครงการ พบข้อผิดพลาดดังนี้:

- ฟอรั่มและตารางไม่ได้ระบุ <label> ให้ครบถ้วน
- การใช้ ARIA role ไม่ถูกต้อง เช่น role="img" ซ้ำซ้อน
- การจัดลำดับ Tabindex ไม่เป็นระบบ
- Contrast ของข้อความสถานะ เช่น “In Progress” ต่ำ

หน้า Project Details แสดงข้อมูลเชิงลึกของแต่ละโครงการ ช่องฟอรั่มและตารางต้องมี label เพื่อให้ screen reader สามารถอ่านและเข้าใจข้อมูล การใช้ ARIA role ซ้ำซ้อนอาจทำให้เกิดความสับสน การจัดลำดับ Tabindex ไม่เป็นระบบทำให้ผู้ใช้งานคีย์บอร์ดเข้าถึงฟังก์ชันได้ยาก ข้อความที่ contrast ต่ำทำให้ผู้ที่มีสายตาเลือนรางอ่านข้อมูลยาก

ข้อเสนอแนะ

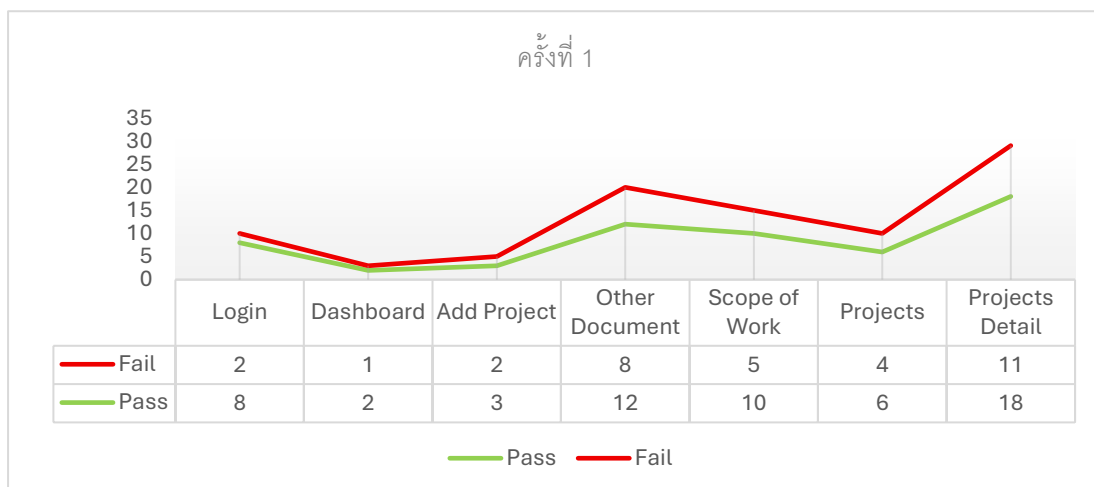
1. กำหนด <label> ให้ครบถ้วนทุกช่องฟอรั่มและคอลัมน์
2. ใช้ ARIA role เฉพาะที่จำเป็นและถูกต้องตามบริบท
3. จัดลำดับ Tabindex ให้สอดคล้องกับลำดับการอ่านของหน้า
4. ปรับสีข้อความสถานะให้ contrast เพียงพอตามมาตรฐาน WCAG



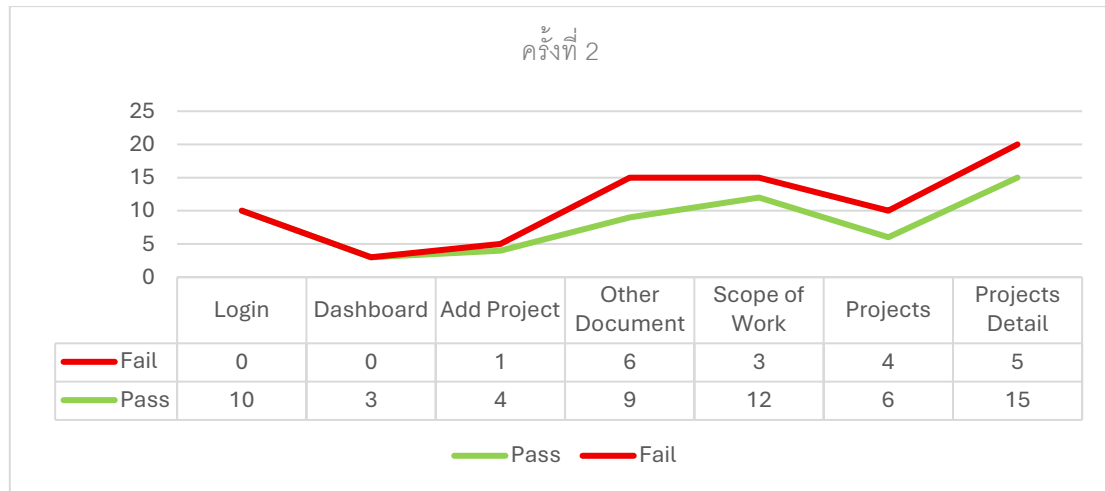
รูปที่ 5.23 การวิเคราะห์ความสามารถในการเข้าถึง ของหน้า Project Details ด้วยเครื่องมือ WAVE

## 5.7 สรุปผลการทดสอบระบบ Issue & Support Management System

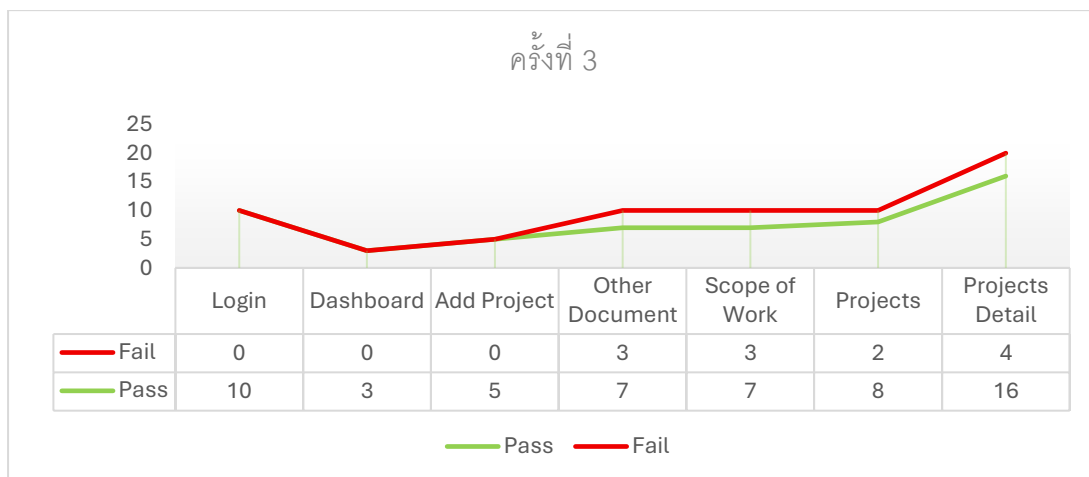
การทดสอบระบบ Issue & Support Management System มีวัตถุประสงค์เพื่อประเมินประสิทธิภาพ ความถูกต้อง และความเสถียรของฟังก์ชันหลักทั้งหมดของระบบ ครอบคลุมตั้งแต่การเข้าสู่ระบบ การจัดการโครงการ การจัดการผู้ใช้งาน การจัดการเอกสาร Scope of Work ไปจนถึงการแจ้งและตอบรายงานปัญหา การทดสอบดำเนินการทั้งแบบแมนนวล (Manual Testing) และแบบอัตโนมัติ (Automated Testing) โดยใช้เครื่องมือ Playwright เพื่อเก็บข้อมูลจำนวนครั้งที่ทดสอบ จำนวนครั้งที่ผ่านเกณฑ์ และจำนวนครั้งที่เกิดข้อผิดพลาด



รูปที่ 5.24 การทดสอบระบบอัตโนมัติ ครั้งที่ 1 ด้วยเครื่อง Playwright



รูปที่ 5.25 การทดสอบระบบอัตโนมัติ ครั้งที่ 2 ด้วยเครื่อง Playwright



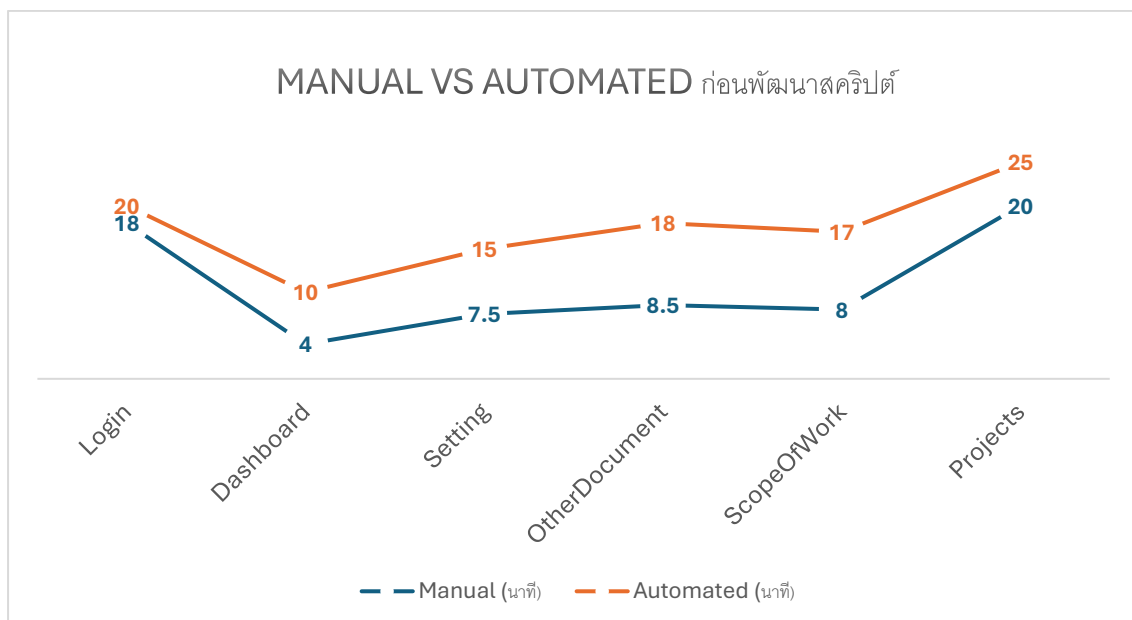
รูปที่ 5.26 การทดสอบระบบอัตโนมัติ ครั้งที่ 3 ด้วยเครื่อง Playwright

ผลการทดสอบชี้ให้เห็นว่าฟังก์ชันพื้นฐาน เช่น การเข้าสู่ระบบ (Login) และหน้าแดชบอร์ด (Dashboard) มีความเสถียรสูง โดย Login ผ่านการทดสอบ 28 ครั้งจากทั้งหมด 30 ครั้ง (ร้อยละ 93.3) และ Dashboard ผ่าน 8 ครั้งจาก 9 ครั้ง (ร้อยละ 88.9) ขณะที่ฟังก์ชันที่ซับซ้อนมากขึ้น เช่น การเพิ่มโครงการ (Add Project) ผ่านการทดสอบ 12 ครั้งจาก 15 ครั้ง (ร้อยละ 80) ส่วนการจัดการเอกสารอื่น ๆ (Other Document) ผ่าน 28 ครั้งจาก 50 ครั้ง (ร้อยละ 56) และรายละเอียดโครงการ (Projects Detail) ผ่าน 49 ครั้งจาก 69 ครั้ง (ร้อยละ 71) ฟังก์ชัน Scope of Work และ Projects มีอัตราการผ่านร้อยละ 64.4 และ 66.7 ตามลำดับ ซึ่งสะท้อนว่าฟังก์ชันที่เกี่ยวข้องกับการจัดการข้อมูลจำนวนมากหรือมีหลายขั้นตอนยังมีข้อผิดพลาดอยู่บ้าง

ในช่วงเริ่มต้นของการทดสอบระบบ การดำเนินการด้วย Manual Testing มีความรวดเร็วกว่า Automated Testing เนื่องจากไม่ต้องพัฒนาสคริปต์ ผู้ทดสอบสามารถดำเนินการได้ทันทีตามโมดูลต่าง ๆ โดยเวลา Manual ต่อโมดูลอยู่ระหว่าง 3–20 นาที ในขณะที่ Automated Testing ในช่วงแรกใช้เวลาน้อย แต่ยังไม่ครอบคลุมทุกขั้นตอนของการทดสอบ ซึ่งสะท้อนว่าการทดสอบอัตโนมัติในระยะแรกยังไม่สามารถลดเวลาอย่างมีนัยสำคัญ

ตารางที่ 5.7 ช่วงเริ่มต้นของการทดสอบ (Manual vs Automated ก่อนพัฒนาสคริปต์เต็ม)

หน้าที่ทำการทดสอบ	Test Case	Manual	Automated	% ประหยัดเวลา
login	9	18 นาที	20 นาที	-11.1%
dashboard	2	4 นาที	10 นาที	150%
setting	4	7.5 นาที	15 นาที	-100%
otherDocument	4	8.5 นาที	18 นาที	-111.8%
scopeOfWork	4	8 นาที	17 นาที	-112.5%
projects	8	20 นาที	25 นาที	-25%

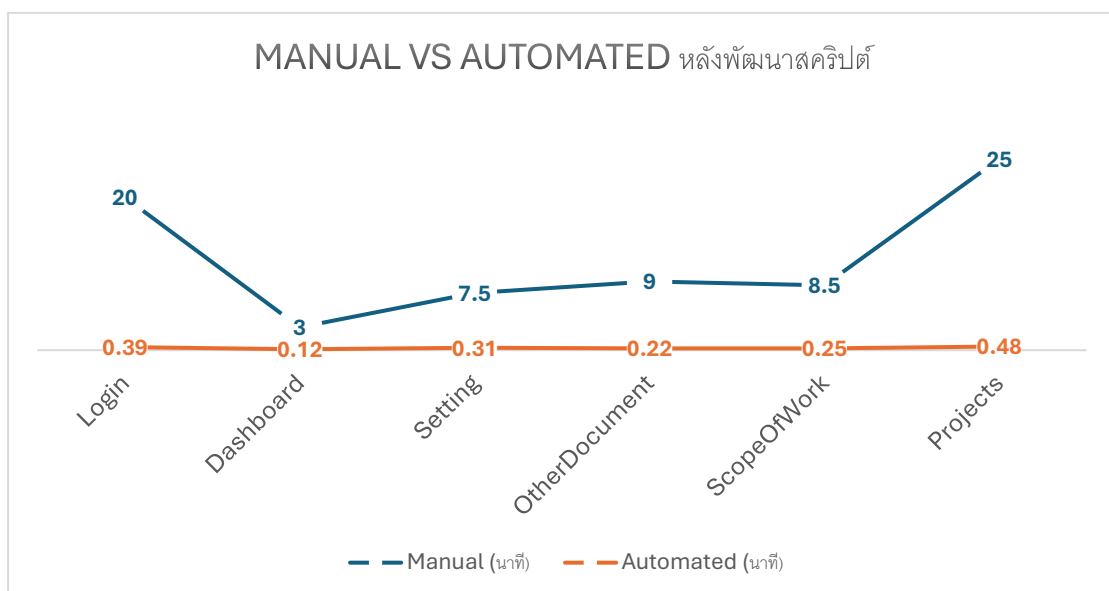


รูปที่ 5.27 Manual vs Automated ก่อนพัฒนาสคริปต์

หลังจากพัฒนาสคริปต์อัตโนมัติครบถ้วน การทดสอบอัตโนมัติสามารถรันซ้ำได้ต่อเนื่อง ลดความผิดพลาดจากมนุษย์ และสร้างหลักฐานการทดสอบที่ตรวจสอบย้อนหลังได้อย่างครบถ้วน ตารางหลังพัฒนาสคริปต์แสดงว่า Automated Testing สามารถลดเวลาการทดสอบได้อย่างมีประสิทธิภาพ โดยเปอร์เซ็นต์เวลาที่ประหยัดเมื่อเทียบกับ Manual อยู่ระหว่าง 96–98% ต่อโมดูล เช่น การทดสอบโมดูล Login ลดเวลาจาก 20 นาที เหลือ 0.39 นาที และโมดูล Projects ลดจาก 25 นาที เหลือ 0.48 นาที

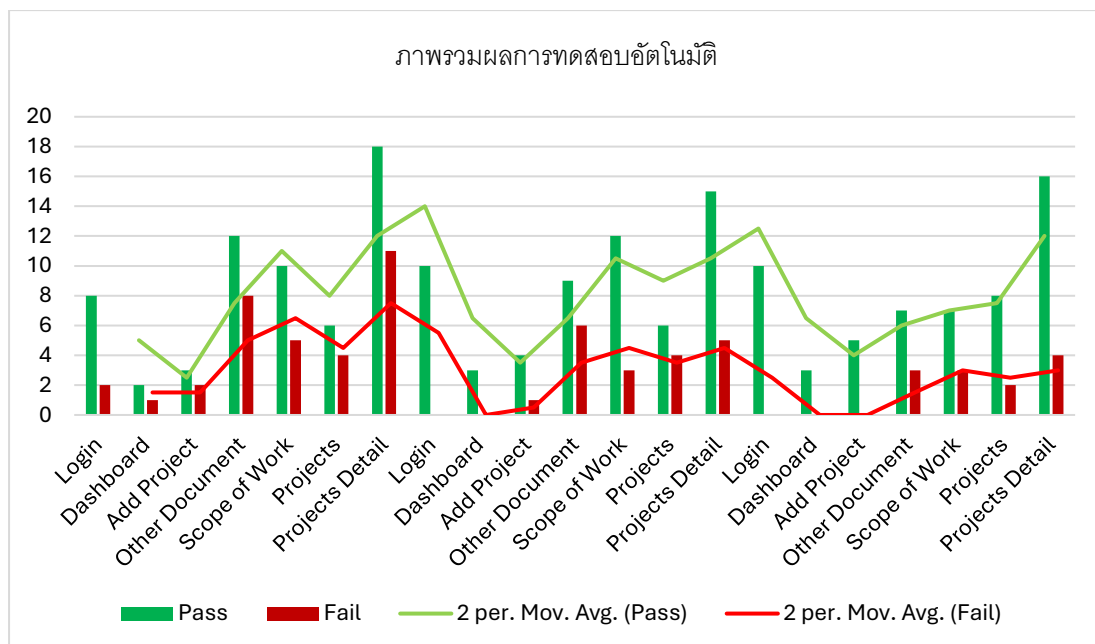
ตารางที่ 5.8 หลังพัฒนาสคริปต์อัตโนมัติครบถ้วน (Manual vs Automated Run)

หน้าที่ทำการทดสอบ	Test Case	Manual	Automated	% ประหยัดเวลา
login	9	20 นาที	0.39 นาที	98%
dashboard	2	3 นาที	0.12 นาที	96%
setting	4	7.5 นาที	0.31 นาที	96%
otherDocument	4	9 นาที	0.22 นาที	98%
scopeOfWork	4	8.5 นาที	0.25 นาที	97%
projects	8	25 นาที	0.48 นาที	98%



รูปที่ 5.28 Manual vs Automated หลังพัฒนาสคริปต์

อย่างไรก็ตาม Manual Testing ยังคงเหมาะสมสำหรับฟังก์ชันใหม่หรือกรณีทดสอบเฉพาะที่ต้องอาศัยการวิเคราะห์และการตัดสินใจของผู้ทดสอบ ตารางเปรียบเทียบดังกล่าวชี้ให้เห็นศักยภาพของ Automated Testing ในการเพิ่มประสิทธิภาพ ลดเวลาในการทดสอบซ้ำหลายครั้ง และสร้างความต่อเนื่องในการทดสอบ แม้ว่าผลการทดสอบทั้งหมดจะผ่านเกณฑ์มาตรฐานแล้ว แต่เพื่อยกระดับประสิทธิภาพและประสบการณ์ผู้ใช้งาน ควรพิจารณาปรับปรุงเพิ่มเติม เช่น การเพิ่มคำแนะนำบนแบบฟอร์ม การปรับปรุงประสิทธิภาพการโหลดข้อมูลขนาดใหญ่ และการเสริมมาตรการความปลอดภัย เช่น การยืนยันตัวตนแบบสองขั้นตอน



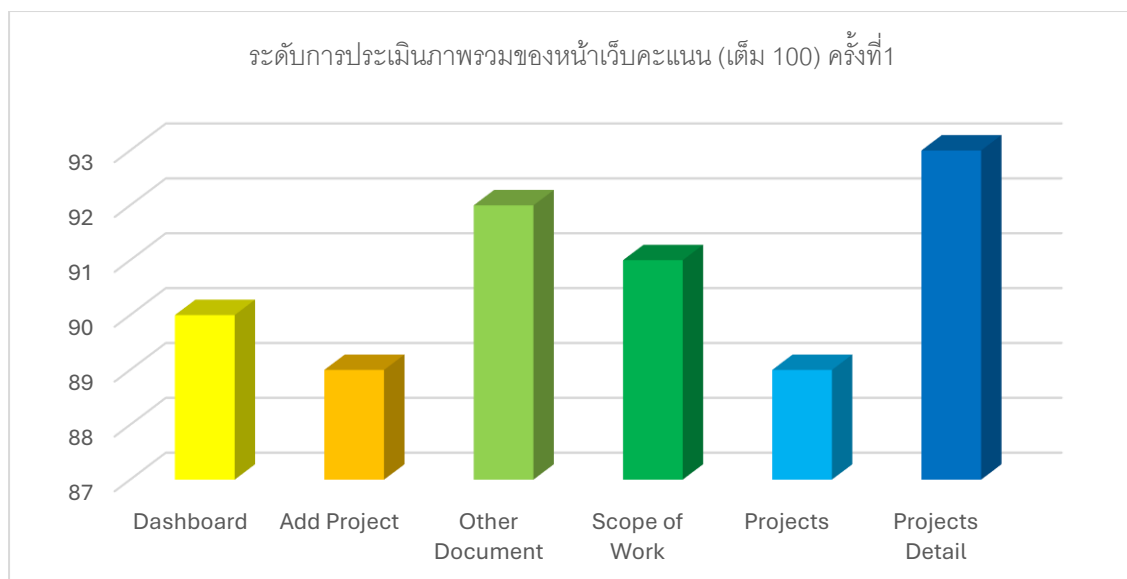
รูปที่ 5.29 ภาพรวมการทดสอบระบบอัตโนมัติ ด้วยเครื่อง Playwright

จึงสรุปผลได้ว่า การทดสอบระบบ Issue & Support Management System ชี้ให้เห็นว่าระบบสามารถตอบสนองความต้องการขององค์กรได้อย่างมีประสิทธิภาพ ทั้งในด้านความถูกต้อง ความรวดเร็ว และความเสถียร อีกทั้งระบบมีความพร้อมในการใช้งานจริง และสามารถเป็นแนวทางสำหรับการปรับปรุงหรือขยายฟังก์ชันเพิ่มเติมได้อย่างเหมาะสม นอกจากนี้ ระบบยังมีศักยภาพที่จะพัฒนาต่อยอดและขยายฟังก์ชันในอนาคต เพื่อรองรับความต้องการที่เพิ่มขึ้นขององค์กร

## 5.8 สรุปผลการทดสอบ Accessibility Testing

การทดสอบความสามารถในการเข้าถึงเว็บไซต์ (Web Accessibility Testing) เป็นกระบวนการประเมินเพื่อให้มั่นใจว่าเว็บไซต์สามารถรองรับการใช้งานของผู้ใช้ทุกกลุ่มได้อย่างเท่าเทียม โดยเฉพาะผู้ใช้ที่มีข้อจำกัดทางร่างกายหรือประสาทสัมผัส เช่น ผู้พิการทางสายตา การได้ยิน หรือการเคลื่อนไหว การทดสอบดังกล่าวมุ่งเน้นตรวจสอบความสามารถในการเข้าถึงข้อมูล ฟังก์ชันการทำงาน และเนื้อหาสำคัญของระบบ เพื่อให้ผู้ใช้ทุกกลุ่มสามารถใช้งานเว็บไซต์ได้อย่างครบถ้วน มีประสิทธิภาพ และสอดคล้องกับหลักการสากล

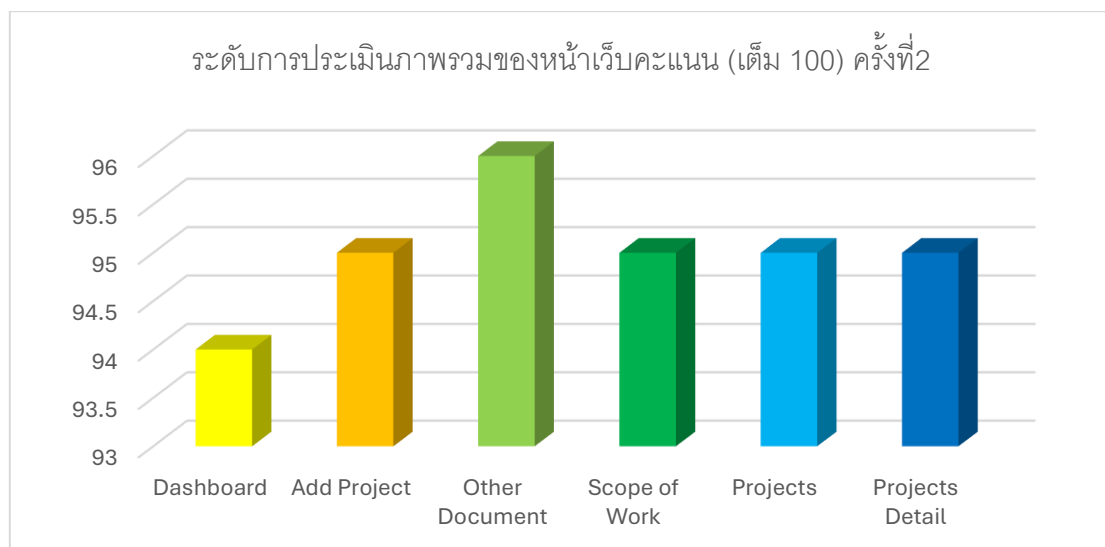
ผลการทดสอบครั้งแรกแสดงให้เห็นว่าเว็บไซต์ยังมีข้อจำกัดบางประการในการรองรับผู้ใช้งานที่มีความหลากหลาย โดยเฉพาะผู้พิการทางสายตาและผู้ที่ใช้งานผ่านคีย์บอร์ด เนื่องจากองค์ประกอบบางส่วนยังขาดข้อความแทนภาพ (alt text) ปัญหาความคมชัดของสี (color contrast) ไม่สอดคล้องกับมาตรฐาน และปุ่มบางรายการไม่สามารถเข้าถึงได้ด้วยการกดปุ่ม Tab ส่งผลให้การเข้าถึงเนื้อหาบางส่วนไม่สมบูรณ์ สำหรับผลคะแนนการประเมินในแต่ละหน้า ได้แก่ หน้า Dashboard 90/100, หน้า Add Project 89/100, หน้า Other Document 92/100, หน้า Scope of Work 91/100, หน้า Projects 89/100 และหน้า Projects Detail 93/100 ซึ่งสะท้อนว่าเว็บไซต์อยู่ในระดับ “Good” แต่ยังมีข้อบกพร่องบางประการที่ควรได้รับการปรับปรุงเพื่อสอดคล้องกับมาตรฐานสากลและเพิ่มความปลอดภัยในการเข้าถึงข้อมูล



รูปที่ 5.30 ภาพรวมระดับการประเมินภาพรวมของหน้าเว็บคะแนน (เต็ม 100) ครั้งที่ 1



หลังจากดำเนินการปรับปรุงองค์ประกอบที่เกี่ยวข้องกับ Accessibility ผลการทดสอบครั้งที่สองแสดงให้เห็นถึงพัฒนาการที่ชัดเจน เว็บไซต์มีคุณสมบัติด้านการเข้าถึงที่ดีขึ้นอย่างมีนัยสำคัญ โดยคะแนนการประเมินในแต่ละหน้าสูงขึ้นอยู่ในระดับ “Excellent” ได้แก่ หน้า Dashboard 94/100, หน้า Add Project 95/100, หน้า Other Document 96/100, หน้า Scope of Work 95/100, หน้า Projects 95/100 และหน้า Projects Detail 95/100 ซึ่งสะท้อนว่าเว็บไซต์สามารถรองรับการใช้งานของผู้ใช้ทุกกลุ่มได้อย่างมีประสิทธิภาพสูง



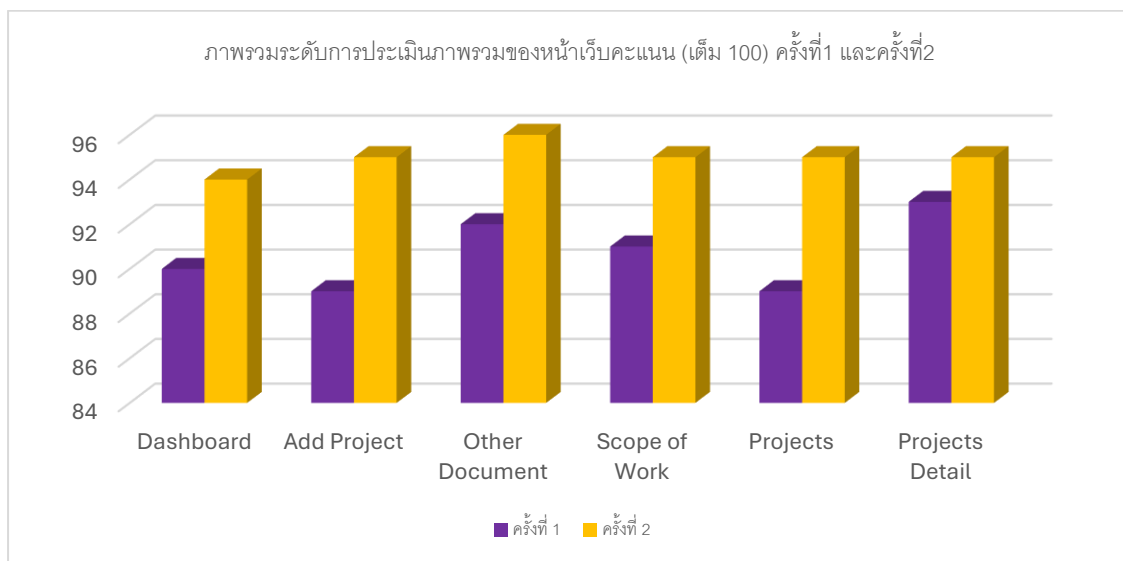
รูปที่ 5.31 ภาพรวมระดับการประเมินภาพรวมของหน้าเว็บคะแนน (เต็ม 100) ครั้งที่ 2

อย่างไรก็ดี การวิเคราะห์ผลการทดสอบครั้งที่สองยังพบข้อจำกัดบางประการที่ควรปรับปรุงเพิ่มเติม เช่น ความแตกต่างของสี (Color Contrast) ในบางส่วนที่ยังไม่สอดคล้องกับมาตรฐาน WCAG 2.1 และโครงสร้างตารางหรือการจัดเรียงข้อมูลที่อาจไม่เหมาะสมสำหรับโปรแกรมอ่านหน้าจอ (Screen Reader) การปรับปรุงประเด็นเหล่านี้จะช่วยยกระดับประสบการณ์การใช้งานให้สมบูรณ์ยิ่งขึ้น

ตารางที่ 5.9 ผลการประเมิน Accessibility ก่อนและหลังปรับปรุง

หน้าเว็บ	คะแนนครั้งที่ 1	ระดับ	คะแนนครั้งที่ 2	ระดับ
Dashboard	90	Good	94	Excellent
Add Project	89	Good	95	Excellent
OtherDocument	92	Good	96	Excellent
Scope of Work	91	Good	95	Excellent
Projects	89	Good	95	Excellent

จากตารางพบว่าระบบได้รับการปรับปรุงจากระดับ “Good” ไปสู่ “Excellent” สะท้อนถึงการพัฒนาที่สอดคล้องกับมาตรฐาน WCAG โดยเฉพาะด้านการจัดโครงสร้างข้อมูลและการปรับปรุงสีอย่างใดก็ตาม ยังคงมีประเด็นที่ต้องพัฒนาเพิ่มเติม เช่น การปรับ color contrast และการใช้งานร่วมกับ screen reader



รูปที่ 5.32 ภาพรวมระดับการประเมินภาพรวมของหน้าเว็บคะแนน (เต็ม 100) ครั้งที่1 และครั้งที่2

สรุปผลการเปรียบเทียบทั้งสองครั้งพบว่าเว็บไซต์มีพัฒนาการด้าน Accessibility อย่างชัดเจน จากระดับ “Good” (คะแนนเฉลี่ยประมาณ 91/100) ในครั้งแรกไปสู่ระดับ “Excellent” (คะแนนเฉลี่ย 95/100) ในครั้งที่สอง การปรับปรุงดังกล่าวไม่เพียงแก้ไขข้อบกพร่องด้าน Accessibility เท่านั้น แต่ยังทำให้เว็บไซต์สามารถรองรับการใช้งานของผู้ใช้ทุกกลุ่มได้อย่างครอบคลุม และเท่าเทียม พร้อมทั้งสอดคล้องกับมาตรฐานสากลด้านการเข้าถึงข้อมูล (Web Content Accessibility Guidelines: WCAG) อย่างครบถ้วน