Test Report API

1. บทนำ (Introduction)

1.1 ภาพรวมของระบบ (System overview)

Painamnae เป็นเว็ปไซต์ที่ใช้สำหรับการจองเส้นทาง สร้างเส้นทาง เพื่อเป็นการเดินทางร่วมกับผู้อื่นได้ ผู้ใช้จะต้องกรอกข้อมูล ชื่อ-นามสกุล อีเมล เบอร์โทรศัพท์ที่ติดต่อได้ เพศ รูปถ่ายบัตรประชาชน รหัสบัตรประชาชน วันหมดอายุบัตรประชาชน รูปถ่ายหน้าตัวเอง และกดยืนยันการใช้ข้อมูลส่วนตัว จึงจะสามารถลงทะเบียนได้สำเร็จ

1.2 วัตถุประสงค์ในการทดสอบ (Purpose of Test)

เพื่อการทดสอบ Application Programming Interface (API) ของระบบ Traffic Log, Export Log และการลบบัญชีผู้ใช้งาน

2. วิธีการทดสอบและกระบวนการทดสอบ (Test methodology and test process)

ผู้ทดสอบทำการทดสอบระดับ User Acceptance Test (UAT) โดยมุ่งเน้นการทดสอบผ่านส่วนต่อประสานโปรแกรมประยุกต์ (API Testing) ด้วยวิธีการทดสอบแบบอัตโนมัติ (Automated Test) ผ่านเครื่องมือ Bruno เพื่อตรวจสอบความถูกต้องของระบบตามกรณีดังต่อไปนี้:

2.1 ขอบเขตและกรณีการทดสอบ (Test Scenarios)

1. การตรวจสอบสิทธิ์เข้าใช้งาน (Authentication): ทดสอบการเข้าสู่ระบบเพื่อรับค่า JWT Token สำหรับใช้ใน Request ถัดไป.
2. การจัดการบันทึกข้อมูล (Log Management):
    - การดึงข้อมูลประวัติการใช้งาน (Traffic Logs) สำหรับผู้ดูแลระบบ.
    - การดึงข้อมูลประวัติการส่งออก (Export Logs).
    - การบันทึกประวัติการส่งออกข้อมูลใหม่เข้าสู่ระบบ.
3. การจัดการข้อมูลส่วนบุคคล (Privacy & Deletion): การลบบัญชีและข้อมูลผู้ใช้งานออกจากระบบเมื่อไม่ต้องการใช้งานอีกต่อไป.

2.2 ขั้นตอนและกระบวนการทดสอบ (Test Process)

ในการทดสอบแบบอัตโนมัติด้วย Bruno มีขั้นตอนการดำเนินงานดังนี้:

1. Preparation: กำหนดค่า Environment Variable สำหรับ URL ของ API (Base URL) และข้อมูล Credentials (Email/Password).

2. Authentication Step: ส่ง HTTP POST Request ไปที่ Endpoint Login เพื่อตรวจสอบความถูกต้องของรหัสผ่านและรับ JWT Token.
3. Token Management: ใช้ Bruno Script ในการดึงค่า Token จาก Response Body มาเก็บไว้ในตัวแปรระบบโดยอัตโนมัติ เพื่อใช้แนบไปกับ Header (Authorization: Bearer Token) ในการทดสอบ API เส้นอื่น ๆ.
4. Execution: ส่ง Request ไปยัง Endpoint ที่ต้องการทดสอบ (GET/POST) พร้อมแนบ Header และ Body ตามที่ออกแบบไว้.
5. Validation: ตรวจสอบผลลัพธ์ที่ตอบกลับมา (Assertions) โดยพิจารณาจาก:
    - HTTP Status Code: เช่น 200 OK หรือ 201 Created.
    - Response Body: ตรวจสอบความถูกต้องของ JSON Schema, ค่า success = true และความครบถ้วนของ Object data.
6. Reporting: บันทึกผลการทดสอบลงใน Test Result Table

2. สภาพแวดล้อมในการทดสอบ (Testing Environment)

3.1 ฮาร์ดแวร์ (Hardware)

คอมพิวเตอร์

CPU : Intel Core i5-11400F

RAM : 24 GB

3.2 ซอฟต์แวร์ (Software)

Visual Studio code

3.3 ผู้ทดสอบ (Human)

นางสาวกมลวรรณ ดีคง รหัสนักศึกษา 663380197-8 sec2

นายวิธวินท์ บุญสมร รหัสนักศึกษา 663380233-0 sec 2

3. รายละเอียดการทดสอบ (Test scenario and test design)

4.1 คำอธิบาย (Description)

✓ หมายถึง ผ่าน (Pass)

✗ หมายถึง ไม่ผ่าน (Fail)

Test Scenario ID:	Item 1		**Project ID:**	PaiNamNaeWebAppSec2_6	
Test Scenario Name:	As an admin, I want a log that complies to the related law.		**Tested by:**	นางสาวกมลวรรณ ดีคง นายวิธวินท์ บุญสมร	
			Date of Test:	17 มีนาคม 2569	
No.	**Test Case and Steps**	**Expected Result**	**Actual Result**	**Test Result (Pass/Fail)**	**Remark**
TC-01	1. ดึงค่า email และ password จาก Environment Variable 2. ส่ง Request ไปยัง Endpoint /api/auth/login 3. ระบบตรวจสอบความถูกต้องของรหัสผ่าน 4. ระบบสร้าง JWT Token และส่งกลับมาพร้อมข้อมูลผู้ใช้ 5. Bruno Script ทำการดึงค่า Token จาก Response มาเก็บลงตัวแปร token	1. HTTP Status Code ต้องเป็น 200 OK 2. Response Body ต้องมีฟิลด์ data.token เป็น String (ไม่เป็นค่าว่าง) 3. Response Body ต้องมี Object data.user แสดงข้อมูลผู้ใช้ 4. ค่า Token ถูกบันทึกลง Environment Variable เรียบร้อย (ตรวจสอบได้จาก Console Log: "Token saved...")	(200 OK) - 359 ms Token saved to Environment Variable successfully. Token Received: eyJhbGciOi..... ✓ Status code is 200 ✓ Token is returned ✓ User data exists	**Pass**	
TC-02	1. ส่ง HTTP GET ไปที่ {{baseUrl}}/api/traffic-logs/admin 2. แนบ Bearer Token ใน Authorization Header 3. กำหนด Query Parameter page=1 และ limit=10	1.ระบบตอบกลับ HTTP Status 200 2.Response เป็น JSON 3.มี success = true 4. data เป็น Array (จำนวนข้อมูลไม่เกิน 10 รายการ)	ระบบตอบกลับ Status 200, success = true และ data เป็น Array ตามที่กำหนด	**Pass**	
TC-03	1.ส่ง HTTP GET ไปที่ {{baseUrl}}/api/export-logs/admin 2.แนบ Bearer Token ใน Authorization Header	1.ระบบตอบกลับ HTTP Status 200 2.Response เป็น JSON	ระบบตอบกลับ Status 200, success = true	**Pass**	

	3.กำหนด Query Parameter page=1 และ limit=10	3.มี success = true 4. data เป็น Array (จำนวนข้อมูลไม่เกิน 10 รายการ)	และ data เป็น Array ตามที่กำหนด		

TC-04	1.ส่ง HTTP POST ไปที่ {{baseUrl}}/api/export-logs/admin
	2.แนบ Bearer Token ใน Authorization Header
	3.กำหนด Header Content-Type: application/json และ Accept: application/json
	4.ส่ง Request Body แบบ JSON:{
	"startDate": "2026-01-01",
	"endDate": "2026-02-01",
	"rowCount": 150,
	"securityMeasure": "Password Protected CSV"
	}
		1.ระบบตอบกลับ HTTP Status 201
		2.Response เป็น JSON
		3.มี success = true
		4.มี message = "บันทึกประวัติการ Export สำเร็จ"
		5.มี property data แสดงข้อมูลที่ถูกบันทึก
			ระบบตอบกลับ HTTP Status Code = 201
			และมี success = true พร้อม message
			และ data ตามที่กำหนด
				Pass	

Test Scenario ID:	Item 16	**Project ID:**	PaiNamNaeWebAppSec2_6
Test Scenario Name:	As a user, I want my account and information to be removed from the system when I am no longer want to be apart of this community	**Tested by:**	นางสาวกมลวรรณ ดีคง นายวิธวินท์ บุญสมร
		Date of Test:	17 มีนาคม 2569

No.	Test Case and Steps	Expected Result	Actual Result	Test Result (Pass/Fail)	Remark

TC-01	1.ส่ง HTTP POST ไปที่ {{baseUrl}}/api/auth/login
	2.กำหนด Header Content-Type: application/json
	3.ส่ง Request Body แบบ JSON:{
	"email": "{{email}}",
	"password": "{{password}}"
	}
		1. HTTP Status Code ต้องเป็น 200 OK
		2. Response Body ต้องมีฟิลด์ data.token เป็น String (ไม่เป็นค่าว่าง)
		3. Response Body ต้องมี Object data.user แสดงข้อมูลผู้ใช้
		4. ค่า Token ถูกบันทึกลง Environment Variable เรียบร้อย (ตรวจสอบได้จาก Console Log: "Token saved...")
			ระบบตอบกลับ Status 200 และมี token กับ user object ครบถ้วน พร้อมบันทึก token ได้สำเร็จ	**Pass**	

TC-02	1.ส่ง HTTP GET ไปที่ {{baseUrl}}/api/traffic-logs/admin 2.แนบ Bearer Token ใน Authorization Header 3.กำหนด Header Accept: application/json 4.กำหนด Query Parameter page=1 และ limit=10	1.ระบบตอบกลับ HTTP Status Code = 200 2.Response เป็น JSON 3.มี success = true 4.มี property data และ data ต้องเป็น Array	ระบบตอบกลับ Status 200 และมี success = true พร้อม data เป็น Array ตามที่กำหนด	**Pass**	

		5.จำนวนข้อมูลไม่เกิน 10 รายการ (ตามค่า limit)			
TC-03	1.ส่ง HTTP GET ไปที่ {{baseUrl}}/api/export-logs/admin 2.แนบ Bearer Token ใน Authorization Header 3.กำหนด Header Accept: application/json 4.กำหนด Query Parameter page=1 และ limit=10	1.ระบบตอบกลับ HTTP Status Code = 200 2.Response เป็น JSON 3.มี success = true 4.มี property data และ data ต้องเป็น Array 5.จำนวนข้อมูลไม่เกิน 10 รายการ (ตามค่า limit)	ระบบตอบกลับ Status 200 และมี success = true พร้อม data เป็น Array ตามที่กำหนด	**Pass**	

TC-04	1.ส่ง HTTP POST ไปที่ {{baseUrl}}/api/export-logs/admin
	2.แนบ Bearer Token ใน Authorization Header
	3.กำหนด Header Content-Type: application/json และ Accept: application/json
	4.ส่ง Request Body แบบ JSON:{
	"startDate": "2026-01-01",
	"endDate": "2026-02-01",
	"rowCount": 150,
	"securityMeasure": "Password Protected CSV"
	}
		1.ระบบตอบกลับ HTTP Status Code = 201
		2.Response เป็น JSON
		3.มี success = true
		4.มี message = "บันทึกประวัติการ Export สำเร็จ"
		5.มี property data แสดงข้อมูลที่ถูกบันทึก
			ระบบตอบกลับ Status 201 และมี success = true พร้อม message และ data ตามที่กำหนด	**Pass**	

5. รายงานสรุปผลการทดสอบ (Test Summary Report)

Scenario ID	Scenario Name	Test Case#	Pass	Fail	No run	Block	Remark	Defect ID
Item 1	As an admin, I want a log that complies to the related law.		4	0			ตรงตามที่ออกแบบ	

Item 16	As a user, I want my account and information to be removed from the system when I am no longer want to be apart of this community.		4	0			ตรงตามที่ออกแบบ	
รวม			8	0				