

รายงานการปฏิบัติงานสหกิจศึกษา

การออกแบบและพัฒนาเว็บแอปพลิเคชัน Automate Daily Monitoring (ส่วนที่ 2)

นายธาม แก้วเกษ

ปฏิบัติงาน ณ
บริษัท เอ-โฮสต์ จำกัด
อาคารเอส เอ็ม ทาวเวอร์ ชั้น 21 เลขที่ 979/52-55 ถ. พหลโยธิน แขวงพญาไท
เขตพญาไท กรุงเทพมหานคร

รายงานนี้เป็นส่วนหนึ่งของวิชาสหกิจศึกษา หลักสูตรปริญญาวิทยาศาสตร์บัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ลิขสิทธิ์ของคณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
ปีการศึกษา 2568



COOPERATIVE EDUCATION REPORT

DESIGN AND DEVELOPMENT OF AN AUTOMATE DAILY MONITORING WEB APPLICATION (PART 2)

THAM KAEOKET

PERFORMED AT

A-HOST CO., LTD.

SM TOWER, 21ST FLOOR 979/52–55 PHAHONYOTHIN ROAD PHAYATHAI SUBDISTRICT

PHAYATHAI DISTRICT BANGKOK, THAILAND

A REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE BACHELOR DEGREE OF SCIENCE IN COMPUTER SCIENCE

COPYRIGHT OF FACULTY OF INFORMATICS, BURAPHA UNIVERSITY

2025.



คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

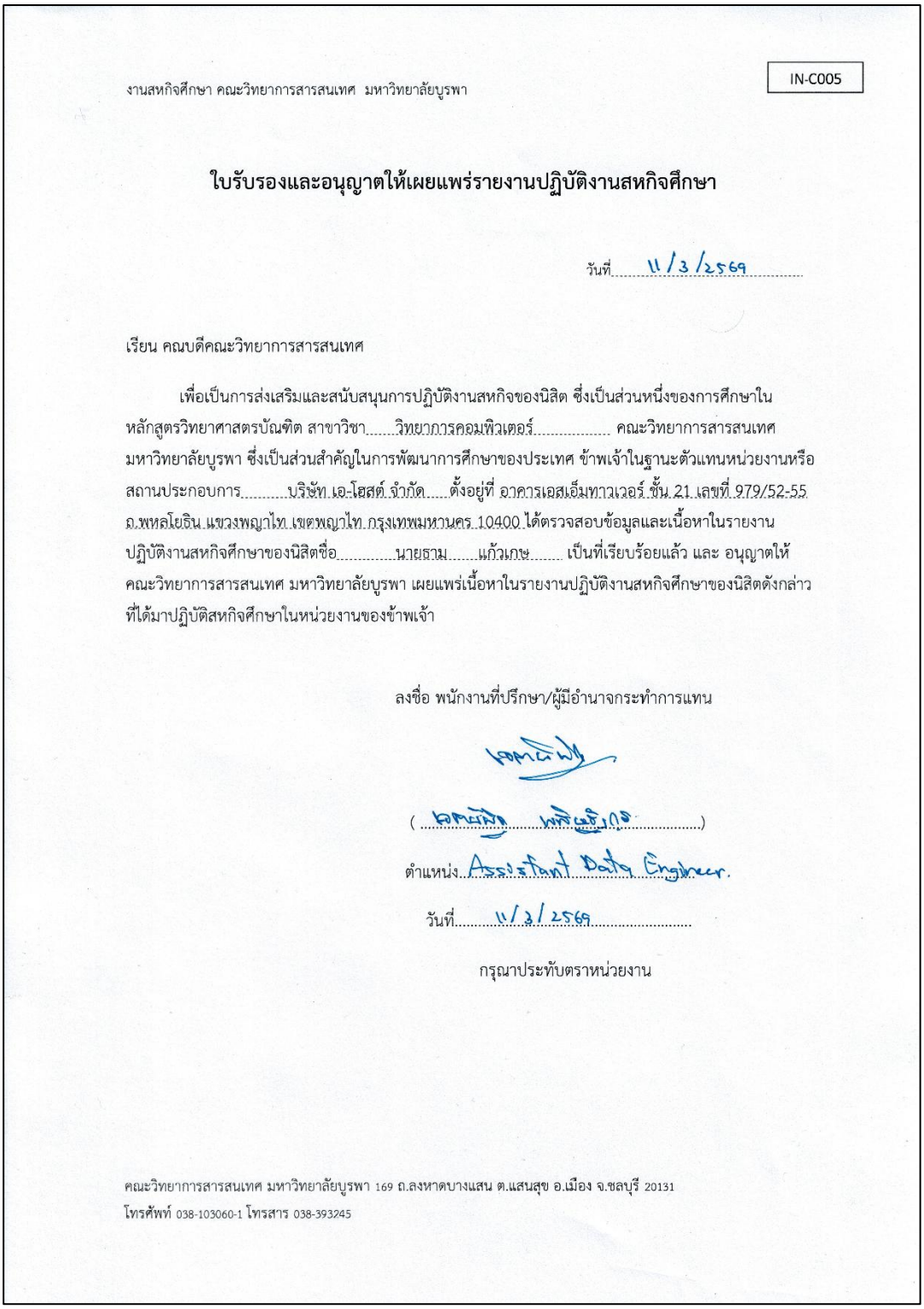
ใบรับรองรายงานการปฏิบัติงานสหกิจศึกษา

รายงาน เรื่อง	การออกแบบและพัฒนาเว็บแอปพลิเคชัน Automate Daily Monitoring
ชื่อนิสิต	นายธาม แก้วเกษ
รหัสประจำตัว	65160172
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ เบญจภรณ์ จันทรกองกุล
วันที่สอบ	26 มีนาคม 2569

รายงานการปฏิบัติงานสหกิจศึกษานี้ได้ผ่านการเห็นชอบจากคณะกรรมการสอบ
ให้เป็นรายงานการปฏิบัติงานสหกิจศึกษาหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์

.....	.....
(ผู้ช่วยศาสตราจารย์ ดร. ประจักษ์ จิตเงินมะดัน)	(ผู้ช่วยศาสตราจารย์ เบญจภรณ์ จันทรกองกุล)
ประธานกรรมการ	กรรมการ

ใบรับรองและอนุญาตให้เผยแพร่รายงานปฏิบัติงานสหกิจศึกษา (IN-C005)



งานสหกิจศึกษา คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

IN-C005

ใบรับรองและอนุญาตให้เผยแพร่รายงานปฏิบัติงานสหกิจศึกษา

วันที่ 11/3/2569

เรียน คณบดีคณะวิทยาการสารสนเทศ

เพื่อเป็นการส่งเสริมและสนับสนุนการปฏิบัติงานสหกิจของนิสิต ซึ่งเป็นส่วนหนึ่งของการศึกษาในหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชา วิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ซึ่งเป็นส่วนสำคัญในการพัฒนาการศึกษาของประเทศ ข้าพเจ้าในฐานะตัวแทนหน่วยงานหรือสถานประกอบการ บริษัท เอ-โฮสต์ จำกัด ตั้งอยู่ที่ อาคารเอสเอ็มทาวเวอร์ ชั้น 21 เลขที่ 979/52-55 ถ.พหลโยธิน แขวงพญาไท เขตพญาไท กรุงเทพมหานคร 10400 ได้ตรวจสอบข้อมูลและเนื้อหาในรายงานปฏิบัติงานสหกิจศึกษาของนิสิตชื่อ นายธาม แก้วเกษ เป็นที่เรียบร้อยแล้ว และ อนุญาตให้คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา เผยแพร่เนื้อหาในรายงานปฏิบัติงานสหกิจศึกษาของนิสิตดังกล่าวที่ได้มาปฏิบัติสหกิจศึกษาในหน่วยงานของข้าพเจ้า

ลงชื่อ พนักงานที่ปรึกษา/ผู้มีอำนาจกระทำการแทน

( )

ตำแหน่ง Assistant Data Engineer.

วันที่ 11/3/2569

กรุณาประทับตราหน่วยงาน

คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา 169 ถ.ลงหาดบางแสน ต.แสนสุข อ.เมือง จ.ชลบุรี 20131
โทรศัพท์ 038-103060-1 โทรสาร 038-393245

กิตติกรรมประกาศ

รายงานสหกิจศึกษาฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาและการสนับสนุนจากหลายๆ ฝ่าย ข้าพเจ้าขอขอบพระคุณ บริษัท เอ-โฮสต์ จำกัด (A-HOST Co., Ltd.) ที่เปิดโอกาสให้ข้าพเจ้าได้เข้ามาปฏิบัติงานในฐานะนิสิตสหกิจศึกษา ซึ่งทำให้ข้าพเจ้าได้รับประสบการณ์การทำงานในสภาพแวดล้อมจริงทางธุรกิจที่หาไม่ได้จากในห้องเรียน

ขอขอบพระคุณ นายเจตนิพิฐ พสิษฐ์จังกูร พนักงานพี่เลี้ยง ที่คอยดูแลเอาใจใส่ ให้คำปรึกษา และถ่ายทอดความรู้ ตลอดจนทักษะและประสบการณ์การทำงานจริงในสายงาน Cloud Engineer / System Admin อย่างไม่ปิดบัง รวมถึงการชี้แนะแนวทางในการแก้ปัญหาต่างๆ ระหว่างการพัฒนาระบบ Automate Daily Monitoring จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี ตลอดจนขอขอบคุณพี่ๆ ทีมงานในแผนกทุกท่านที่ให้ความเป็นกันเอง คอยช่วยเหลือ และให้คำแนะนำที่เป็นประโยชน์แก่ข้าพเจ้าเสมอมา

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ เบญจภรณ์ จันทรกองกุล อาจารย์ที่ปรึกษาสหกิจศึกษา คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ที่สละเวลาให้คำปรึกษา แนะนำแนวทางที่ถูกต้องในการจัดทำรายงาน ช่วยตรวจสอบและแก้ไขข้อบกพร่องต่างๆ รวมถึงติดตามความก้าวหน้าในการปฏิบัติงานอย่างใกล้ชิดด้วยความเอาใจใส่

ขอขอบพระคุณ คณาจารย์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา ทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้ วางรากฐานทางวิชาการ และเตรียมความพร้อมให้แก่ข้าพเจ้าก่อนออกไปปฏิบัติงานสหกิจศึกษา

ท้ายที่สุดนี้ คุณค่าและประโยชน์อันพึงมีจากรายงานและโครงงานสหกิจศึกษาฉบับนี้ ข้าพเจ้าขอมอบเป็นความกตัญญูกตเวทีแด่บิดา มารดา และครอบครัว ที่คอยให้ความรัก ความอบอุ่น เป็นกำลังใจสำคัญ และให้การสนับสนุนข้าพเจ้าในทุกๆ ด้านเสมอมา หากรายงานฉบับนี้มีข้อบกพร่องหรือผิดพลาดประการใด ข้าพเจ้าขอน้อมรับไว้และขออภัยมา ณ ที่นี้

นายธาม แก้วเกษ
ผู้จัดทำ

หัวข้อรายงาน การออกแบบและพัฒนาเว็บแอปพลิเคชัน Automate Daily Monitoring
นิสิต นายธาม แก้วเกษ
รหัสประจำตัว 65160172
อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ เบญจภรณ์ จันทรกองกุล
ระดับการศึกษา วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์
คณะ วิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
ปีการศึกษา 2568

บทคัดย่อ

การปฏิบัติงานสหกิจศึกษา ณ บริษัท เอ-โฮสต์ จำกัด (A-HOST Co., Ltd.) ผู้จัดทำได้รับมอบหมายให้พัฒนาโครงงาน "การออกแบบและพัฒนาเว็บแอปพลิเคชัน Automate Daily Monitoring" โดยมีวัตถุประสงค์เพื่อปรับปรุงและเพิ่มประสิทธิภาพกระบวนการตรวจสอบสถานะระบบเซิร์ฟเวอร์ประจำวัน (Morning Monitoring Check) จากเดิมที่ต้องดำเนินการด้วยตนเอง (Manual Process) ซึ่งใช้เวลา 1-2 ชั่วโมง และมีความเสี่ยงต่อความผิดพลาดจากบุคคล ให้เป็นระบบอัตโนมัติที่รวดเร็วและแม่นยำมากยิ่งขึ้น

ระบบถูกออกแบบและพัฒนาโดยใช้เทคโนโลยี n8n เป็นเครื่องมือหลักในการจัดการกระบวนการอัตโนมัติ (Workflow Automation) เพื่อเชื่อมต่อผ่าน SSH เข้าไปยังเซิร์ฟเวอร์ของลูกค้า ดึงข้อมูลสถานะการใช้ทรัพยากร (เช่น CPU, RAM, Disk Storage) และทำการคัดกรองแปลงข้อมูลให้อยู่ในรูปแบบ JSON จากนั้นจัดเก็บลงในฐานข้อมูลลงบนเครื่องและแสดงผลภาพรวมผ่านหน้าเว็บแอปพลิเคชันที่พัฒนาด้วย React ซึ่งมีระบบ Health Indicators แจ้งเตือนสถานะความผิดปกติด้วยสี และรองรับการดูข้อมูลสถิติย้อนหลัง รวมถึงการส่งออกรายงานในรูปแบบ PDF

ผลการดำเนินงานพบว่า ระบบสามารถทำงานได้อย่างถูกต้องและมีประสิทธิภาพ สามารถลดระยะเวลาการตรวจสอบระบบประจำวันลงเหลือเพียง 5 นาที (ลดลงกว่าร้อยละ 70) ช่วยลดภาระงานซ้ำซ้อน (Routine Task) ของทีมงาน ลดข้อผิดพลาดในการเก็บข้อมูล และช่วยให้สามารถนำข้อมูลสถิติไปวิเคราะห์เพื่อป้องกันปัญหาระบบล่ม (Proactive Management) ได้อย่างรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น

Report Title	Design and Development of an Automate Daily Monitoring Web Application
Student	Mr. Tham Kaeoket
Student ID	65160172
Advisor	Asst.Prof. Benchaporn Jantarakongkul
Level of Study	Bachelor degree of Science in Computer Science
Faculty	Faculty of Informatics, Burapha University
Year	2025

ABSTRACT

During the cooperative education program at A-HOST Co., Ltd., the author was assigned to develop a project titled "Design and Development of Automate Daily Monitoring Web Application." The primary objective was to improve and optimize the daily server status monitoring process (Morning Monitoring Check). The traditional manual process was time-consuming, taking approximately 1-2 hours daily, and was highly susceptible to human error.

The system was designed and developed utilizing n8n as the core workflow automation tool to automatically establish SSH connections to client servers. It retrieves resource utilization data (e.g., CPU, RAM, Disk Storage), parses the output into JSON format, and stores it in a local database. A frontend dashboard, built with React, was developed to visualize the overall server status. The application features color-coded health indicators to alert users of anomalies, historical data viewing capabilities, and automated PDF report generation.

The results demonstrate that the developed system operates accurately and efficiently. It successfully reduced the daily monitoring time to only 5 minutes (over a 70% reduction). The implementation significantly decreased routine workload, minimized data collection errors, and enabled the team to utilize statistical data for proactive management, thereby preventing potential system failures promptly and effectively.

สารบัญ

หน้า

กิตติกรรมประกาศ ..... ก

บทคัดย่อ ..... ข

ABSTRACT ..... ค

สารบัญ ..... ง

สารบัญภาพ ..... ช

สารบัญตาราง ..... ฎ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

1.5 ข้อมูลของบริษัท เอ-โฮสต์ จำกัด 2

1.5.1 สถานที่ตั้งของบริษัท เอ-โฮสต์ จำกัด 3

1.5.2 ผลิตภัณฑ์ ผลิตผล หรือการให้บริการของบริษัท เอ-โฮสต์ จำกัด 3

1.5.3 ตำแหน่งงานและลักษณะงานที่ได้รับมอบหมาย 4

1.5.4 ข้อมูลพนักงานที่ปรึกษา 4

1.6 ระยะเวลาการปฏิบัติงาน 4

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง 6

2.1 นิยามศัพท์ของคำสำคัญ 6

2.2 ทฤษฎีและบทความที่เกี่ยวข้อง 9

2.2.1 การเฝ้าระวังระบบ (Monitoring) 9

2.2.2 การตรวจสอบสุขภาพระบบ (Health Check) 9

2.2.3 Docker Container 9

สารบัญ (ต่อ)

หน้า

2.2.4 Secure Shell (SSH) 9
2.2.5 Low-Code 10
2.2.6 n8n Workflow Automation 10
2.2.7 VPN 10
2.3 เครื่องมือและเทคโนโลยีที่ใช้ในการปฏิบัติงาน 11
2.3.1 เครื่องมือที่ช่วยในการพัฒนาระบบ 11
2.3.2 เทคโนโลยีและภาษาที่ใช้ในการพัฒนาระบบ 15
2.3.3 เครื่องมือที่ช่วยอำนวยความสะดวกในการดำเนินงานเอกสาร 16
บทที่ 3 รายละเอียดของการปฏิบัติงาน 19
3.1 วิเคราะห์และออกแบบระบบ Automate Daily Monitoring 19
3.2 ขั้นตอนการออกแบบ Back-end ของ Automate Daily Monitoring 22
3.2.1 Refresh Data 22
3.2.2 File Log Monitoring 35
3.2.3 Use Case Diagram 43
3.2.4 Use Case Description 44
3.3 ขั้นตอนการออกแบบ Front-end ของ Automate Daily Monitoring 56
บทที่ 4 ผลการดำเนินงานสหกิจศึกษา 61
4.1 ผลการพัฒนาระบบส่วน Back-end 61
4.2 ผลการพัฒนาระบบส่วน Front-end 63
4.2.1 การแสดงผลกระดานข้อมูลภาพรวม 63
4.2.2 การแสดงผลข้อมูลเชิงลึกและการเปรียบเทียบ 64
บทที่ 5 สรุปและวิจารณ์ผลการปฏิบัติงานสหกิจศึกษา 68
5.1 สรุปผลการดำเนินงาน 68

สารบัญ (ต่อ)

หน้า

5.2 ประโยชน์ที่ได้รับจากการปฏิบัติงานสหกิจศึกษา 69
5.2.1 ประโยชน์ต่อตนเอง 69
5.2.2 ประโยชน์ต่อสถานประกอบการ 69
5.3 ปัญหาและอุปสรรคในการปฏิบัติงาน 69
5.4 ข้อเสนอแนะ 70
5.4.1 ข้อเสนอแนะในการนำระบบไปพัฒนาต่อ 70
5.4.2 ข้อเสนอแนะสำหรับนักศึกษารุ่นต่อไป 70
บรรณานุกรม 71
ภาคผนวก 72
ภาคผนวก ก ติดตั้งโปรแกรมและส่วนขยายที่เกี่ยวข้อง 73
ภาคผนวก ข คู่มือการใช้งานระบบ 88
ภาคผนวก ค ใบนำส่งและเอกสารสหกิจที่เกี่ยวข้อง 96
ประวัติย่อของผู้จัดทำ 100

สารบัญภาพ

หน้า

ภาพที่ 1-1 A-host Logo 2
ภาพที่ 1-2 แผนที่ บริษัท เอ-โฮสต์ จำกัด 3
ภาพที่ 2-1 n8n Logo 11
ภาพที่ 2-2 Docker Logo 12
ภาพที่ 2-3 Visual Studio Code Logo 12
ภาพที่ 2-4 GitHub Logo 13
ภาพที่ 2-5 GitHub Runner Logo 13
ภาพที่ 2-6 VPN Access Manager Logo 14
ภาพที่ 2-7 FortiClient Logo 14
ภาพที่ 2-8 JavaScript Logo 15
ภาพที่ 2-9 React Logo 15
ภาพที่ 2-10 PostgreSQL Logo 16
ภาพที่ 2-11 Microsoft Word Logo 16
ภาพที่ 2-12 Microsoft Excel Logo 17
ภาพที่ 2-13 Microsoft PowerPoint Logo 17
ภาพที่ 2-14 Microsoft Outlook Logo 18
ภาพที่ 2-15 Line Logo 18
ภาพที่ 3-1 Flowchart ขั้นตอนการทำ Daily Monitoring 20
ภาพที่ 3-2 Flowchart ขั้นตอนการทำ Daily Monitoring ด้วยระบบ Automate 21
ภาพที่ 3-3 ตัวอย่าง Flowchart Backend Refresh Data 22
ภาพที่ 3-4 Post Refresh Data Node 23
ภาพที่ 3-5 Config Post Refresh Data Node 23
ภาพที่ 3-6 Switch Server Node 24
ภาพที่ 3-7 Config Switch Server Node 24
ภาพที่ 3-8 Sub Workflow SSH Node 25
ภาพที่ 3-9 ภายใน Sub Workflow SSH Node 25
ภาพที่ 3-10 SSH Command top Node 26
ภาพที่ 3-11 Config SSH Command Node 26

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 3-12 IF SSH Node 27
ภาพที่ 3-13 Config IF SSH Node 27
ภาพที่ 3-14 Parse Data Node 28
ภาพที่ 3-15 Config Parse Data Node 28
ภาพที่ 3-16 Convert Data Node 29
ภาพที่ 3-17 Config Convert Data Node 29
ภาพที่ 3-18 Create Folder Node 30
ภาพที่ 3-19 Config Create Folder Node 30
ภาพที่ 3-20 Merge Save Node 31
ภาพที่ 3-21 Config Merge Save Node 31
ภาพที่ 3-22 Save File Node 32
ภาพที่ 3-23 Config Save File Node 32
ภาพที่ 3-24 Edit Fields Status Node 33
ภาพที่ 3-25 Config Edit Fields Status Node 33
ภาพที่ 3-26 Merge All Respond Node 34
ภาพที่ 3-27 All Respond Node 34
ภาพที่ 3-28 Flowchart Backend File Log Monitoring 35
ภาพที่ 3-29 Get List Node 35
ภาพที่ 3-30 Find File Node 36
ภาพที่ 3-31 Build Path List Node 36
ภาพที่ 3-32 Read All Node 37
ภาพที่ 3-33 Build List Node 37
ภาพที่ 3-34 Get File Node 38
ภาพที่ 3-35 Find File Node 38
ภาพที่ 3-36 Read Target File Node 39
ภาพที่ 3-37 Extract from File Node 39
ภาพที่ 3-38 Clean Data Node 40
ภาพที่ 3-39 Config Clean Data Node 40

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ 3-40 Download File Node 41
ภาพที่ 3-41 Find File Node 41
ภาพที่ 3-42 Read Target File Node 42
ภาพที่ 3-43 Respond to Webhook Node 42
ภาพที่ 3-44 Use Case Diagram 43
ภาพที่ 3-45 ภาพหน้าจอหลัก (Home) 56
ภาพที่ 3-46 ภาพข้อมูล Statistics 57
ภาพที่ 3-47 ภาพข้อมูล Running Processes (Top) 58
ภาพที่ 3-48 ภาพข้อมูล Disk Storage 59
ภาพที่ 4-1 ตัวอย่าง Flowchart Backend Refresh Data 61
ภาพที่ 4-2 ภายใน Sub Workflow SSH Node 62
ภาพที่ 4-3 Flowchart Backend File Log Monitoring 63
ภาพที่ 4-4 หน้าจอแสดงข้อมูลภาพรวม Statistics 64
ภาพที่ 4-5 หน้าจอแสดงข้อมูล Running Processes (Top) 64
ภาพที่ 4-6 หน้าจอแสดงข้อมูล Disk Storage และทำการ Compare 65
ภาพที่ 4-7 ปุ่ม Export PDF 65
ภาพที่ 4-8 ผลลัพธ์การ Export PDF 65
ภาพที่ 4-9 ผลลัพธ์ไฟล์ PDF 66
ภาพที่ 4-10 ผลลัพธ์การ Download ไฟล์ 67
ภาพที่ 4-11 ผลลัพธ์ไฟล์ที่ทำการ Download 67
ภาพที่ ก-1 ค้นหา Visual Studio Code บนเว็บบราว์เซอร์ 74
ภาพที่ ก-2 Download Visual Studio Code บนเว็บบราว์เซอร์ 74
ภาพที่ ก-3 ผลการ Download Visual Studio Code 75
ภาพที่ ก-4 ผลการกด Accept agreement Visual Studio Code 75
ภาพที่ ก-5 ผลการกด add tasks Visual Studio Code 75
ภาพที่ ก-6 การกด install Visual Studio Code 76
ภาพที่ ก-7 Finish Install Visual Studio Code 76
ภาพที่ ก-8 ค้นหา Node.js บนเว็บบราว์เซอร์ 77

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ ก-9 Node.js Windows Installer (.msi) 77
ภาพที่ ก-10 ผลการ Download Node.js 77
ภาพที่ ก-11 Node.js Setup Wizard Next..... 78
ภาพที่ ก-12 ผลการกด Accept agreement Node.js..... 78
ภาพที่ ก-13 Folder Install Node.js 79
ภาพที่ ก-14 Custom Setup Node.js 79
ภาพที่ ก-15 Tools Node.js..... 80
ภาพที่ ก-16 Install Node.js 81
ภาพที่ ก-17 Node.js check 81
ภาพที่ ก-18 Node.js React 81
ภาพที่ ก-19 ค้นหา Docker บนเว็บบราว์เซอร์ 82
ภาพที่ ก-20 Download for Windows – AMD 64 82
ภาพที่ ก-21 เปิดโฟลเดอร์ของ Docker Desktop Installer ที่ดาวน์โหลดมา..... 83
ภาพที่ ก-22 เปิดไฟล์ Docker Desktop Installer.exe เพื่อทำการติดตั้ง 83
ภาพที่ ก-23 หน้าจอติดตั้ง Docker Desktop..... 84
ภาพที่ ก-24 Process Install Docker Desktop 84
ภาพที่ ก-25 หน้าจอติดตั้ง Docker Desktop เสร็จสิ้น..... 85
ภาพที่ ก-26 หน้าจอกดสร้าง Volumes..... 86
ภาพที่ ก-27 หน้าจอ Download Images n8n 86
ภาพที่ ก-28 images n8n 86
ภาพที่ ก-29 Config n8n container..... 87
ภาพที่ ก-30 หน้าจอเว็บ n8n 87
ภาพที่ ข-1 หน้าจอ Home 89
ภาพที่ ข-2 การเลือกช่วงเวลา (Date Range) Trend Charts..... 89
ภาพที่ ข-3 หน้าจอ Statistics 90
ภาพที่ ข-4 หน้าจอ Statistics รูปแบบการแสดงผลแบบ Donut และปรับค่า KB 90
ภาพที่ ข-5 หน้าจอ Top Running Processes เลือกวันที่ 91
ภาพที่ ข-6 หน้าจอ Top Running Processes เลือกเวลา..... 91

สารบัญภาพ (ต่อ)

หน้า

ภาพที่ ข-7 การกดปุ่ม Download 92
ภาพที่ ข-8 ผลการ Download 92
ภาพที่ ข-9 หน้าจอ Disk Storage 93
ภาพที่ ข-10 หน้าจอ Disk Storage Compare 93
ภาพที่ ข-11 หน้าจอ Disk Storage เลือกวันที่ Compare 94
ภาพที่ ข-12 หน้าจอ Home 94
ภาพที่ ข-13 การยืนยันที่จะ Export 95
ภาพที่ ข-14 ผลการ Export 95

สารบัญตาราง

หน้า

ตารางที่ 1-1 แผนการปฏิบัติงานสหกิจศึกษา 4
ตารางที่ 2-1 นิยามศัพท์ของคำสำคัญ 6
ตารางที่ 3-1 UC1 การ Monitor 45
ตารางที่ 3-2 UC 2 การ Refresh ข้อมูล 47
ตารางที่ 3-3 UC 3 การย้อนดูข้อมูล 49
ตารางที่ 3-4 UC 4 การเปรียบเทียบข้อมูล 51
ตารางที่ 3-5 UC 5 การ Export ข้อมูล 52
ตารางที่ 3-6 UC 6 การ Download ข้อมูล 54

บทที่ 1
บทนำ

รายงานการปฏิบัติงานสหกิจศึกษาฉบับนี้ เป็นส่วนหนึ่งของโครงงานพัฒนาระบบ Automate Monitoring ซึ่งมีการแบ่งขอบเขตการดำเนินงานออกเป็น 2 ส่วนหลัก เพื่อให้ครอบคลุมวงจรการพัฒนาระบบซอฟต์แวร์อย่างสมบูรณ์

สำหรับรายงานฉบับนี้ จัดทำโดย นายธาม แก้วเกษ ซึ่งจะมุ่งเน้นการนำเสนอรายละเอียดในส่วนของ การพัฒนาส่วนจัดการข้อมูลเบื้องหลัง (Backend) ผ่านเครื่องมือ n8n และการพัฒนาเว็บแอปพลิเคชัน (Web Application)

โดยขอบเขตงานหลักจะครอบคลุมตั้งแต่การใช้เครื่องมือ n8n ในการสร้าง Workflow เพื่อทำหน้าที่เป็นระบบ Backend สำหรับเชื่อมต่อกับเซิร์ฟเวอร์ของลูกค้าผ่าน SSH และ VPN เพื่อรวบรวมและจัดการข้อมูล ทั้งนี้ เพื่อให้ระบบสามารถประมวลผลได้อย่างมีเสถียรภาพ และพร้อมรองรับการส่งต่อข้อมูลไปแสดงผลยังส่วนหน้าเว็บ (Front-end) ได้อย่างมีประสิทธิภาพ ซึ่งรายละเอียดการทำงานในส่วนของการวางโครงสร้างระบบ จะปรากฏในรายงานอีกฉบับหนึ่ง

1.1 ที่มาและความสำคัญ

ในปัจจุบันที่ระบบเทคโนโลยีสารสนเทศเป็นหัวใจสำคัญในการขับเคลื่อนธุรกิจ บริษัท เอ-โฮสต์ จำกัด ในฐานะผู้เชี่ยวชาญด้านการดูแลระบบไอที จึงมีภารกิจหลักในการบริหารจัดการและตรวจสอบความพร้อมใช้งานของตัวเครื่องและระบบปฏิบัติการให้แก่ลูกค้าทั้งภายในและภายนอกองค์กร กระบวนการที่ถือเป็นหัวใจสำคัญในแต่ละวันคือการทำ Morning Monitoring Check หรือการตรวจสอบสถานะระบบในช่วงเช้า เพื่อให้มั่นใจว่าเครื่องของลูกค้าทุกรายพร้อมสำหรับการดำเนินงานก่อนเริ่มเวลาทำการปกติ อย่างไรก็ตาม ด้วยฐานลูกค้าที่มีจำนวนเพิ่มมากขึ้นอย่างต่อเนื่อง ส่งผลให้ภาระงานในการตรวจสอบระบบเพิ่มขึ้นเป็นเงาตามตัว

ปัญหาหลักที่พบในปัจจุบันคือกระบวนการตรวจสอบส่วนใหญ่ยังคงดำเนินการด้วยแรงงานคน (Manual Process) ซึ่งเป็นงานในลักษณะซ้ำเดิมทุกวัน (Routine Task) ทำให้เจ้าหน้าที่ต้องสูญเสียเวลาในช่วงเช้าไปกับการเข้าถึงระบบต่าง ๆ ของลูกค้าทีละรายเพื่อเก็บข้อมูลสถานะ การทำงานในลักษณะนี้นอกจากจะใช้เวลานานและใช้ทรัพยากรบุคคลสูงแล้ว ยังมีความเสี่ยงที่จะเกิดความผิดพลาดจากตัวบุคคล (Human Error) รวมถึงความล่าช้าในการสรุปผลภาพรวมหากพบความผิดปกติของระบบในหลายจุดพร้อมกัน

ด้วยเหตุนี้ คณะผู้จัดทำจึงได้รับมอบหมายให้พัฒนาโครงงานระบบอัตโนมัติเพื่อเข้ามาแก้ไขปัญหาดังกล่าว โดยมุ่งเน้นการนำเครื่องมือ n8n ซึ่งเป็นแพลตฟอร์มการจัดการ Workflow ในรูปแบบ Low-code มาประยุกต์ใช้ในการสร้างระบบ Automate Monitoring ในส่วนของ Back-end เพื่อเชื่อมต่อข้อมูลและรายงานสถานะได้โดยอัตโนมัติ แม้ว่าในเบื้องต้น

คณะผู้จัดทำจะยังไม่มีความเชี่ยวชาญเชิงลึกเกี่ยวกับระบบ Automation หรือการใช้งาน n8n มาก่อน แต่ด้วยความตั้งใจที่จะลดขั้นตอนการทำงานที่ซ้ำซ้อนและเพิ่มความแม่นยำในการปฏิบัติงาน โครงงานนี้จึงริเริ่มขึ้นเพื่อปรับเปลี่ยนวิธีการทำงานแบบเดิมสู่ระบบดิจิทัลที่ทันสมัย ซึ่งจะช่วยประหยัดเวลาการทำงานลงได้อย่างมีนัยสำคัญ และเปิดโอกาสให้เจ้าหน้าที่สามารถนำเวลาไปพัฒนาและดูแลงานด้านเทคนิคในส่วนอื่น ๆ ที่มีความซับซ้อนได้ดียิ่งขึ้น

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อพัฒนาเว็บแอปพลิเคชันสำหรับการ Monitoring
2. เพื่อลดขั้นตอนการทำงานแต่ละขั้นตอนหลังจาก Monitoring ตอนเช้าให้น้อยลง
3. เพื่อศึกษาเรียนรู้เกี่ยวกับเรื่องของการ Automate กับ n8n

1.3 ขอบเขตของโครงงาน

1. สามารถลดเวลาการ Monitoring ตอนเช้าได้
2. สามารถลดขั้นตอนการทำงานได้
3. สามารถดูเป็นกราฟ หรือ สถิติได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มประสิทธิภาพในการทำงาน
2. การบริหารจัดการข้อมูลอย่างเป็นระบบ
3. ประหยัดเวลาการทำงาน

1.5 ข้อมูลของบริษัท เอ-โฮสต์ จำกัด

ชื่อสถานประกอบการ: บริษัท เอ-โฮสต์ จำกัด

เว็บไซต์: https://www.a-host.co.th/



ภาพที่ 1-1 A-host Logo

1.5.1 สถานที่ตั้งของบริษัท เอ-โฮสต์ จำกัด

สถานที่ตั้ง: อาคารเอสเอ็มทาวเวอร์ ชั้น 21 เลขที่ 979/52-55 ถ.พหลโยธิน แขวงพญาไท เขตพญาไท กรุงเทพมหานคร 10400

เบอร์ติดต่อ: 02 298 0625



ภาพที่ 1-2 แผนที่ บริษัท เอ-โฮสต์ จำกัด

1.5.2 ผลิตภัณฑ์ ผลิตผล หรือการให้บริการของบริษัท เอ-โฮสต์ จำกัด

บริษัท เอ-โฮสต์ จำกัด เป็นผู้เชี่ยวชาญด้านการให้บริการไอทีโซลูชันครบวงจร (IT Solutions Provider) และเป็นตัวแทนจำหน่ายซอฟต์แวร์ระดับโลกอย่างเป็นทางการ โดยมีการให้บริการหลักแบ่งออกเป็นกลุ่มต่างๆ ดังนี้

1. Managed Services: เป็นบริการหลักที่เกี่ยวข้องกับโครงงานนี้ โดยเป็นการรับดูแลและบริหารจัดการระบบไอทีแบบครบวงจร (Outsourcing) ครอบคลุมการดูแลฐานข้อมูล (Database), ระบบสำรองข้อมูลและกู้คืน (Backup & DR), รวมถึงการตรวจสอบสถานะระบบ (Monitoring) เพื่อให้มั่นใจว่าระบบของลูกค้าทำงานได้อย่างต่อเนื่อง 24x7

2. Cloud Services: ให้บริการคำปรึกษา ออกแบบ และติดตั้งระบบบนโครงสร้างพื้นฐานคลาวด์ (Cloud Infrastructure) ทั้งในรูปแบบ Public, Private และ Hybrid Cloud เพื่อเพิ่มความยืดหยุ่นในการดำเนินธุรกิจ

3. ERP & CRM Application: ให้บริการติดตั้งและดูแลระบบซอฟต์แวร์บริหารจัดการทรัพยากรองค์กร (Enterprise Resource Planning) และระบบบริหารความสัมพันธ์ลูกค้า (Customer Relationship Management) โดยเน้นผลิตภัณฑ์ชั้นนำ เช่น Oracle และ Microsoft

4. Data Analytics: บริการวิเคราะห์ข้อมูลทางธุรกิจ (Business Intelligence) และการจัดการข้อมูลขนาดใหญ่ (Big Data) เพื่อนำมาสรุปผลและช่วยในการตัดสินใจเชิงกลยุทธ์ขององค์กร

5. IT Consulting & Software Development: ให้บริการที่ปรึกษาด้านการจัดการไอที การรักษาความปลอดภัยทางไซเบอร์ (Cybersecurity) และการพัฒนาซอฟต์แวร์ให้ตรงตามความต้องการเฉพาะของลูกค้า

6. IT Distributor: เป็นตัวแทนจำหน่ายหลักและผู้เชี่ยวชาญในการให้บริการโซลูชันจากพาร์ทเนอร์ระดับโลก เช่น Oracle และ Microsoft อย่างเป็นทางการในประเทศไทย

1.5.3 ตำแหน่งงานและลักษณะงานที่ได้รับมอบหมาย

ตำแหน่งงาน: Cloud Engineer

ลักษณะงาน: ทำการ Monitoring ตอนเช้าเพื่อตรวจสอบเครื่องของลูกค้าแต่ละเจ้าและทำ Health Check รายเดือนกับลูกค้าแต่ละเจ้าเพื่อทำ Report โดยการจะทำการเก็บข้อมูลของ Disk, Database, log ต่างๆมาทำข้อมูลใส่ลง Report แล้วทำการส่งต่อให้กับทางพี่เลี้ยง และมีการ Support คนภายในบริษัท เช่น เรื่อง Network, Printer, etc.

1.5.4 ข้อมูลพนักงานที่ปรึกษา

ชื่อ-นามสกุล: นายเจตนิพิฐ พสิษฐ์จังกูร

ตำแหน่ง: Assistant Data Engineer

แผนก: System Management Service

โทรศัพท์: 081-9922026

อีเมล: jetnipit@a-host.co.th

1.6 ระยะเวลาการปฏิบัติงาน

ระยะเวลาของการฝึกสหกิจศึกษาระหว่างวันที่ 10 พฤศจิกายน 2568 – 13 มีนาคม 2569

ตารางที่ 1-1 แผนการปฏิบัติงานสหกิจศึกษา

ขั้นตอนการดำเนินงาน	ระยะเวลาในการดำเนินงาน				
	ปี พ.ศ. 2568 - 2569				
	พ.ค.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาหาข้อมูล Cloud	◄——►				
2. เริ่มทำงาน Service internal	◄——	——	——	——	——►
3. พูดคุยเรื่องหัวข้อ Project		◄——►			

ตารางที่ 1-1 แผนการปฏิบัติงานสหกิจศึกษา (ต่อ)

ขั้นตอนการดำเนินงาน	ระยะเวลาในการดำเนินงาน				
	ปี พ.ศ. 2568 - 2569				
	พ.ค.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
4. ทำ Monitoring ตอนเช้า	←	—	—	—	→
5. ไป Health Check	←	—	—	—	→
6. ทำ Report Health Check	←	—	—	—	→
7. ศึกษาหาข้อมูลของ n8n		↔			
8. ติดตั้ง docker เพื่อใช้งาน n8n		↔			
9. สร้างระบบ Back-end SSH ใน n8n		←	—	→	
10. สร้างระบบจัดการไฟล์ใน n8n			↔		
11. สร้าง Front-end ด้วย React		←	—	—	→
12.สร้าง Task Scheduler ด้วย n8n		↔			
12. นำเสนอความคืบหน้าของงาน ครั้งที่ 1		↔			
13. นำเสนอความคืบหน้าของงาน ครั้งที่ 2			↔		
14. นำเสนอความคืบหน้าของงาน ครั้งที่ 3				↔	
15. ทำเอกสารรายงาน				←	→
16. นำเสนอ Project					↔

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

ในการปฏิบัติงานสหกิจศึกษาตำแหน่ง Cloud Engineer ณ บริษัท เอ-โฮสต์ จำกัด ผู้ปฏิบัติงานได้มุ่งเน้นการศึกษาทฤษฎีและทักษะด้าน Cloud Operations เพื่อเพิ่มประสิทธิภาพในการดูแลระบบของลูกค้า โดยครอบคลุมทั้งการเฝ้าระวังสถานะความพร้อมใช้งานรายวัน (Morning Monitoring) และการวิเคราะห์ดัชนีชี้วัดประสิทธิภาพรายเดือน (Monthly Health Check) เพื่อประเมินการใช้ทรัพยากร อาทิ พื้นที่จัดเก็บข้อมูล (Storage) และความสมบูรณ์ของฐานข้อมูล (Database)

เพื่อให้การดำเนินงานมีความแม่นยำและลดขั้นตอนซ้ำซ้อน ผู้ปฏิบัติงานจึงได้ประยุกต์ใช้เครื่องมือ n8n สำหรับการสร้างระบบอัตโนมัติ (Automation) ซึ่งจะช่วยยกระดับกระบวนการรายงานผลให้รวดเร็วและเป็นระบบมากยิ่งขึ้น โดยมีรายละเอียดหลักการและทฤษฎีที่เกี่ยวข้องตามลำดับหัวข้อดังต่อไปนี้

2.1 นิยามศัพท์ของคำสำคัญ

ในส่วนนี้จะเป็นการกล่าวถึงคำอธิบาย และรายละเอียดของคำศัพท์ที่เกี่ยวข้องกับการทำ Project Automate Daily Monitoring ดังตารางที่ 2-1

ตารางที่ 2-1 นิยามศัพท์ของคำสำคัญ

ลำดับ	คำศัพท์	ความหมาย
1	Monitoring	การเฝ้าระวังระบบ คือกระบวนการเข้าไปตรวจสอบเครื่องเซิร์ฟเวอร์ของลูกค้าอย่างสม่ำเสมอ เพื่อประเมินสถานะการทำงาน
2	Health Check	การตรวจสอบสุขภาพระบบ คือการรวบรวมข้อมูลเชิงลึกจากเครื่องเพื่อนำมาประเมินสถานะความพร้อมใช้งานระยะยาว
3	Manual Process	กระบวนการทำงานที่ดำเนินการด้วยแรงงานคน
4	Routine Task	ภาระงานในลักษณะซ้ำเดิมทุกวัน
5	Human Error	ความเสี่ยงหรือความผิดพลาดที่เกิดจากตัวบุคคล
6	Proactive Management	การนำข้อมูลสถิติไปวิเคราะห์เพื่อป้องกันปัญหาระบบล่มหรือการจัดการเชิงรุก
7	Use Case Diagram	แผนภาพกรณีการใช้งาน เป็นเครื่องมือสำคัญในการวิเคราะห์และออกแบบระบบซอฟต์แวร์

ตารางที่ 2-1 นิยามศัพท์ของคำสำคัญ (ต่อ)

8	Docker Container	เทคโนโลยีการจำลองสภาพแวดล้อมในการทำงานของซอฟต์แวร์ (Virtualization) ในระดับ Operating System เพื่อให้แอปพลิเคชันทำงานได้ภายใต้สภาพแวดล้อมที่คงที่
9	Virtual Machine	ระบบจำลองคอมพิวเตอร์
10	Secure Shell	โปรโตคอลเครือข่ายที่ใช้สำหรับการเข้าถึงและบริหารจัดการเครื่องคอมพิวเตอร์แม่ข่ายจากระยะไกลผ่านช่องทางที่มีความปลอดภัย
11	VPN	เทคโนโลยีการสร้างเครือข่ายเสมือนเพื่อเชื่อมต่อข้อมูลผ่านโครงข่ายสาธารณะที่มีการรักษาความปลอดภัยสูง
12	Tunneling	การสร้างอุโมงค์ข้อมูลสำหรับการเชื่อมต่อ VPN
13	Encryption	การเข้ารหัสข้อมูลตลอดการเชื่อมต่อหรือรับส่งข้อมูลเพื่อป้องกันการดักจับ
14	Low-Code	แนวคิดในการพัฒนาซอฟต์แวร์หรือระบบอัตโนมัติโดยใช้การลากวางองค์ประกอบภาพ (Visual Programming) แทนการเขียนชุดคำสั่ง (Manual Coding) จำนวนมาก
15	Workflow Automation	กระบวนการทำงานอัตโนมัติที่เชื่อมต่อแอปพลิเคชันและบริการต่างๆ เข้าด้วยกัน
16	n8n	เครื่องมือประเภท Low-code Workflow Automation ที่ใช้เชื่อมต่อแอปพลิเคชันต่างๆ เข้าด้วยกันผ่าน Node-based UI
17	Visual Studio Code	โปรแกรมแก้ไขซอร์สโค้ด (Source Code Editor) ที่สนับสนุนการดีบักและการใช้งาน Git
18	GitHub	แพลตฟอร์มจัดเก็บซอร์สโค้ดและบริหารจัดการเวอร์ชันของซอฟต์แวร์ (Version Control)
19	GitHub Runner	เครื่องมือรับคำสั่งจาก GitHub Actions มาประมวลผลหรือรันชุดคำสั่ง
20	VPN Access Manager	ซอฟต์แวร์และโปรแกรมประยุกต์ด้านความปลอดภัยที่ใช้บริหารจัดการและเชื่อมต่อเครือข่ายเสมือนส่วนตัว (VPN)

ตารางที่ 2-1 นิยามศัพท์ของคำสำคัญ (ต่อ)

21	FortiClient	ซอฟต์แวร์และโปรแกรมประยุกต์ด้านความปลอดภัยที่ใช้บริหารจัดการและเชื่อมต่อเครือข่ายเสมือนส่วนตัว (VPN)
22	JavaScript	ภาษาโปรแกรมระดับสูงที่ใช้เพิ่มลูกเล่นและประมวลผลแอปพลิเคชัน
23	React	JavaScript Library ที่ใช้สำหรับสร้างส่วนติดต่อผู้ใช้ (User Interface) แบบ Component-based
24	PostgreSQL	ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System)
25	Node	จุดเชื่อมต่อหรือคำสั่งในระบบ n8n เช่น Webhook Node, Switch Server Node, SSH Command Node
26	Trigger	จุดเริ่มต้นการทำงานของระบบอัตโนมัติ
27	Request	การร้องขอข้อมูลหรือคำสั่ง เช่น POST, GET
28	Response	ข้อมูลการตอบกลับ
29	Respond	ข้อมูลการตอบกลับ
30	JSON	รูปแบบโครงสร้างแฟ้มข้อมูลที่ระบบใช้จัดเก็บและแปลงผลลัพธ์
31	Raw Data	ไฟล์ข้อมูลดิบที่ได้จากเซิร์ฟเวอร์เป้าหมาย
32	Dashboard	หน้าจอหลักกระดานข้อมูลภาพรวม
33	Compare	ฟังก์ชันในการเปรียบเทียบข้อมูลปัจจุบันกับอดีต
34	Managed Services	บริการรับดูแลและบริหารจัดการระบบไอทีแบบครบวงจร (Outsourcing)
35	Cloud Services	บริการคำปรึกษา ออกแบบ และติดตั้งระบบบนโครงสร้างพื้นฐานคลาวด์
36	ERP	ระบบซอฟต์แวร์บริหารจัดการทรัพยากรองค์กร
37	CRM	ระบบบริหารความสัมพันธ์ลูกค้า
38	Data Analytics	การวิเคราะห์ข้อมูลทางธุรกิจ
39	Big Data	การจัดการข้อมูลขนาดใหญ่

2.2 ทฤษฎีและบทความที่เกี่ยวข้อง

2.2.1 การเฝ้าระวังระบบ (Monitoring)

การเฝ้าระวังระบบ คือกระบวนการเข้าไปตรวจสอบเครื่องเซิร์ฟเวอร์ของลูกค้าอย่างสม่ำเสมอ เพื่อประเมินว่าระบบทำงานได้ตามปกติหรือไม่ มีข้อขัดข้องเกิดขึ้น หรือการสำรองข้อมูล (Backup) เสร็จสมบูรณ์หรือเกิดข้อผิดพลาด จากนั้นทำการรวบรวมหลักฐานและสรุปผลเพื่อรายงานให้ทีมงานทราบทุกเช้า

2.2.2 การตรวจสอบสุขภาพระบบ (Health Check)

การตรวจสอบสุขภาพระบบ คือการรวบรวมข้อมูลเชิงลึกจากเครื่องของลูกค้าเพื่อนำมาประเมินสถานะความพร้อมใช้งานระยะยาว โดยครอบคลุมการตรวจสอบพื้นที่จัดเก็บข้อมูล (Disk), ฐานข้อมูล (Database) และไฟล์บันทึก (Log) นำมาสรุปผลเป็นรายงานเพื่อแจ้งให้ลูกค้าทราบถึงแนวโน้มการใช้งาน หรือเสนอแนะการปรับปรุง/ขยายทรัพยากรระบบ

2.2.3 Docker Container

Docker Container คือเทคโนโลยีการจำลองสภาพแวดล้อมในการทำงานของซอฟต์แวร์ (Virtualization) ในระดับ Operating System เพื่อให้แอปพลิเคชันสามารถทำงานได้ภายใต้สภาพแวดล้อมที่คงที่และเหมือนกันในทุกระบบ โดย Docker จะรวมเอา Code, Runtime, System Tools และ Library ที่จำเป็นเข้าไว้ด้วยกันในรูปแบบของ Container ซึ่งมีความเบา (Lightweight) และเริ่มการทำงานได้รวดเร็วกว่า Virtual Machine ทั่วไป ในโครงงานนี้ Docker มีบทบาทสำคัญในการติดตั้งและบริหารจัดการแพลตฟอร์ม n8n เพื่อให้ระบบ Automate Monitoring สามารถทำงานได้อย่างมีเสถียรภาพและเคลื่อนย้าย (Portability) ไปยังระบบคลาวด์ต่างๆ ได้สะดวก

2.2.4 Secure Shell (SSH)

SSH คือโปรโตคอลเครือข่ายที่ใช้สำหรับการเข้าถึงและบริหารจัดการเครื่องคอมพิวเตอร์แม่ข่าย (Server) จากระยะไกลผ่านช่องทางที่มีความปลอดภัย โดยมีการเข้ารหัสข้อมูล (Encryption) ตลอดการเชื่อมต่อ เพื่อป้องกันการดักจับข้อมูลจากผู้ไม่หวังดี ในงานด้าน Cloud Engineer โปรโตคอล SSH ถูกนำมาใช้เป็นช่องทางหลักในการเข้าไปตรวจสอบสถานะเครื่องของลูกค้า (Monitoring) การดึงข้อมูลสถิติ (Health Check) รวมถึงการสั่งการผ่าน Command Line เพื่อรวบรวมไฟล์ Log หรือตรวจสอบสถานะของ Database จากระยะไกลเพื่อนำมาสรุปผลในรายงาน

2.2.5 Low-Code

Low-Code คือแนวคิดในการพัฒนาซอฟต์แวร์หรือระบบการทำงานอัตโนมัติ (Automation) โดยใช้การลากวางองค์ประกอบภาพ (Visual Programming) แทนการเขียนชุดคำสั่ง (Manual Coding) จำนวนมาก ช่วยให้ผู้พัฒนาสามารถออกแบบกระบวนการทำงานหรือ Workflow ได้อย่างรวดเร็วและลดโอกาสการเกิดข้อผิดพลาดในการเขียนโปรแกรม โดยแพลตฟอร์ม n8n ที่นำมาใช้ในโครงงานนี้เป็นเครื่องมือประเภท Low-code ที่ช่วยเชื่อมต่อ API และบริการต่างๆ เข้าด้วยกัน ทำให้การรับ-ส่งข้อมูลสถานะการ Monitoring จาก Server ของลูกค้าไปยังระบบรายงานผลทำได้อย่างมีประสิทธิภาพโดยไม่ต้องพึ่งพาการเขียนโค้ดที่ซับซ้อน

2.2.6 n8n Workflow Automation

n8n คือเครื่องมือประเภท Workflow Automation ในรูปแบบ Open-source ที่ใช้แนวคิด Node-based เพื่อเชื่อมต่อแอปพลิเคชันและบริการต่าง ๆ เข้าด้วยกันผ่าน API โดยจุดเด่นของ n8n คือความยืดหยุ่นในการปรับแต่ง (Self-hosted) และการสนับสนุนการทำงานแบบ Low-code ที่ช่วยให้ผู้พัฒนาสามารถออกแบบลำดับขั้นตอนการทำงานที่ซับซ้อนได้โดยใช้การลากวาง Node มาเชื่อมต่อกัน

2.2.7 VPN

VPN คือเทคโนโลยีการสร้างเครือข่ายเสมือนเพื่อใช้สำหรับการเชื่อมต่อข้อมูลผ่านโครงข่ายสาธารณะที่มีการรักษาความปลอดภัยสูง โดยการสร้างอุโมงค์ข้อมูล (Tunneling) และมีการเข้ารหัสข้อมูล (Encryption) ตลอดการรับส่งข้อมูลเพื่อป้องกันการดักจับข้อมูลจากบุคคลภายนอก ในงานด้าน Cloud Engineer โปรโตคอล VPN ถูกนำมาใช้เป็นด่านแรกในการสร้างช่องทางเชื่อมต่อที่ปลอดภัยจากระยะไกลเข้าสู่ระบบโครงสร้างพื้นฐานไอที (IT Infrastructure) หรือวงเครือข่ายภายใน (Internal Network) ของลูกค้าแต่ละราย เพื่อให้ผู้ปฏิบัติงานสามารถเข้าไปดำเนินการตรวจสอบสถานะเครื่อง (Monitoring) ดึงข้อมูลสถิติ (Health Check) หรือจัดการระบบฐานข้อมูลได้อย่างปลอดภัยตามมาตรฐานความปลอดภัยขององค์กร

2.3 เครื่องมือและเทคโนโลยีที่ใช้ในการปฏิบัติงาน

ในการทำการศึกษาทฤษฎีที่เกี่ยวข้องกับการพัฒนาระบบการทำงานอัตโนมัติ (Automation) ด้วยตนเอง เพื่อประยุกต์ใช้ในการปฏิบัติงานที่ได้รับมอบหมายจากพนักงานพี่เลี้ยง โดยมุ่งเน้นกระบวนการรวบรวมข้อมูลสถานะการทำงานของระบบ (Data Collection) และการบันทึกข้อมูลอย่างเป็นระบบเพื่อใช้ในการตรวจสอบย้อนหลัง ตลอดจนการนำข้อมูลสถิติมาประมวลผลเพื่อแสดงผลในรูปแบบกราฟ (Data Visualization) ให้สอดคล้องกับความต้องการของผู้ใช้งานและการวิเคราะห์ทรัพยากรระบบ ซึ่งรายละเอียดของเครื่องมือและเทคโนโลยีที่ใช้ในการปฏิบัติงานจะแบ่งตามลักษณะการใช้งาน ดังต่อไปนี้

2.3.1 เครื่องมือที่ช่วยในการพัฒนาระบบ

1. n8n

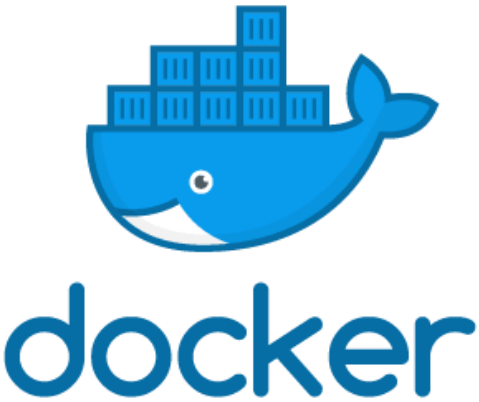
n8n เป็นเครื่องมือประเภท Low-code Workflow Automation ที่ใช้ในการเชื่อมต่อแอปพลิเคชันและบริการต่าง ๆ เข้าด้วยกันผ่าน Node-based UI ซึ่งสนับสนุนการสร้างระบบอัตโนมัติที่ซับซ้อนได้อย่างมีประสิทธิภาพ สามารถกำหนดเงื่อนไขในการประมวลผลข้อมูล (Logic) การเชื่อมต่อผ่านโปรโตคอล SSH เพื่อดึงข้อมูลจาก Server ลูกค้า และการจัดส่งข้อมูลสรุปผลไปยังระบบแจ้งเตือนหรือฐานข้อมูลเพื่อนำไปสร้างรายงานต่อไป



ภาพที่ 2-1 n8n Logo

2. Docker

Docker เป็นแพลตฟอร์มที่ใช้ในการจำลองสภาพแวดล้อม (Containerization) เพื่อให้แอปพลิเคชันและเครื่องมือต่าง ๆ สามารถทำงานได้ในสภาพแวดล้อมที่เสถียรและเป็นอิสระจากระบบปฏิบัติการหลัก โดยมีการนำมาใช้ในการติดตั้งและรัน Service ของ n8n เพื่อให้ง่ายต่อการบริหารจัดการระบบและเพิ่มความสะดวกในการขยายตัวของระบบในอนาคต



ภาพที่ 2-2 Docker Logo

3. Visual Studio Code

Visual Studio Code เป็นโปรแกรมแก้ไขซอร์สโค้ด (Source Code Editor) ที่มีการสนับสนุนสำหรับการดีบัก การควบคุม Git ในตัวและ GitHub การเน้นไวยากรณ์ การเติมโค้ดอัจฉริยะ ตัวอย่าง และ Code Refactoring สามารถปรับแต่งได้หลายอย่าง ให้ผู้ใช้สามารถเปลี่ยนธีม แป้นพิมพ์ลัด การตั้งค่า และติดตั้งส่วนขยายที่เพิ่มฟังก์ชันการทำงานเพิ่มเติม ซึ่งในการปฏิบัติงานได้นำมาใช้เป็นเครื่องมือหลักในการเขียนชุดคำสั่ง JavaScript และการจัดการโครงสร้างโปรเจกต์



ภาพที่ 2-3 Visual Studio Code Logo

4. GitHub

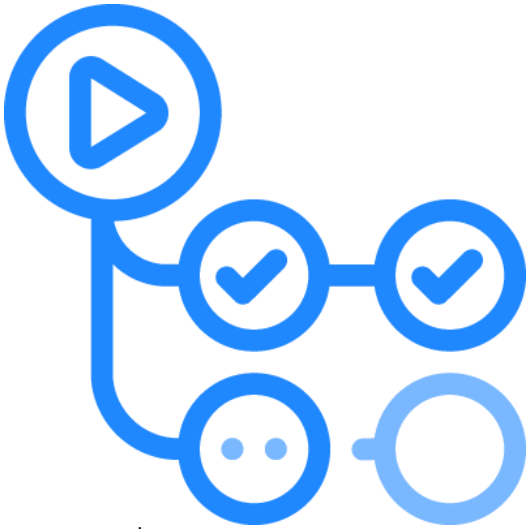
GitHub เป็นแพลตฟอร์มสำหรับการจัดเก็บซอร์สโค้ดและบริหารจัดการเวอร์ชันของซอฟต์แวร์ (Version Control) โดยใช้ระบบ Git ช่วยให้ผู้ปฏิบัติงานสามารถบันทึกประวัติการแก้ไขโค้ด สำรองข้อมูลโปรเจกต์ และทำงานร่วมกับพนักงานพี่เลี้ยงได้อย่างเป็นระบบ รวมถึงยังใช้ในการติดตามความคืบหน้าของงานและการจัดการประเด็นปัญหา (Issue Tracking) ที่เกิดขึ้นระหว่างการพัฒนา



ภาพที่ 2-4 GitHub Logo

5. GitHub Runner

GitHub Runner คือเครื่องมือหรือบริการที่ทำหน้าที่รับคำสั่งจาก GitHub Actions มาประมวลผลหรือรันชุดคำสั่ง (Jobs) ตามที่กำหนดไว้ในไฟล์ Workflow ในการปฏิบัติงานตำแหน่ง Cloud Engineer ได้มีการนำ GitHub Runner มาใช้เป็นตัวกลางในการสั่งรันสคริปต์อัตโนมัติ เพื่อเชื่อมต่อเข้ากับระบบโครงสร้างพื้นฐานบนคลาวด์หรือเครื่องเซิร์ฟเวอร์ของลูกค้า โดยตัว Runner จะคอยจัดการสภาพแวดล้อมให้พร้อมสำหรับการทำงาน และส่งผลลัพธ์กลับไปยังระบบบริหารจัดการเวอร์ชัน (Version Control) เพื่อให้การตรวจสอบและอัปเดตระบบทำได้อย่างต่อเนื่องและเป็นอัตโนมัติ



ภาพที่ 2-5 GitHub Runner Logo

6. VPN Access Manager

VPN Access Manager เป็นซอฟต์แวร์ที่ใช้สำหรับบริหารจัดการและเชื่อมต่อเครือข่ายเสมือนส่วนตัว (VPN) โดยเน้นการใช้งานร่วมกับโปรโตคอลมาตรฐานต่าง ๆ ในการปฏิบัติงานนี้ ผู้ปฏิบัติงานได้นำมาใช้เพื่อสร้างช่องทางเชื่อมต่อที่ปลอดภัยไปยังระบบเครือข่ายของลูกค้าบางรายที่ต้องมีการกำหนดค่าทางเทคนิคเฉพาะตัว ช่วยให้สามารถเข้าถึงทรัพยากรภายในของลูกค้าเพื่อดำเนินการตรวจสอบสถานะเครื่อง (Monitoring) ได้อย่างถูกต้องตามมาตรฐานความปลอดภัย



ภาพที่ 2-6 VPN Access Manager Logo

7. FortiClient

FortiClient เป็นโปรแกรมประยุกต์ด้านความปลอดภัยแบบครบวงจร ที่มีความสามารถในการเชื่อมต่อ SSL VPN และ IPsec VPN รวมถึงมีระบบป้องกัน Endpoint Security ในตัว ในการปฏิบัติงานตำแหน่ง Cloud Engineer ผู้ปฏิบัติงานได้ใช้ FortiClient เป็นเครื่องมือหลักในการเชื่อมต่อเข้าสู่ระบบ IT Infrastructure ของลูกค้า เพื่อเข้าไปรวบรวมข้อมูลสถานะการทำงานและ Log ต่าง ๆ สำหรับการทำ Morning Monitoring และการเก็บสถิติ Health Check รายเดือนได้อย่างมีประสิทธิภาพและมีความปลอดภัยสูงสุด



ภาพที่ 2-7 FortiClient Logo

2.3.2 เทคโนโลยีและภาษาที่ใช้ในการพัฒนาระบบ

1. JavaScript

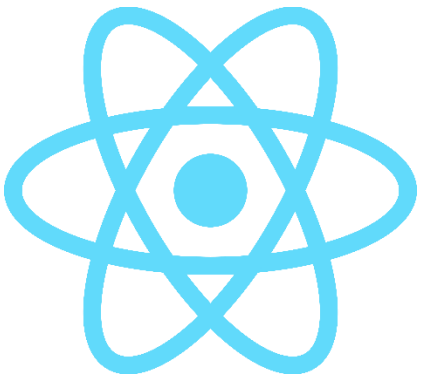
JavaScript เป็นภาษาโปรแกรมระดับสูงที่ใช้ในการเพิ่มลูกเล่นและความสามารถในการประมวลผลให้กับแอปพลิเคชัน ในโครงงานนี้ได้นำ JavaScript มาใช้ร่วมกับ n8n ในรูปแบบของ Node "Function" เพื่อเขียน Logic ที่ซับซ้อนในการจัดการข้อมูล (Data Manipulation) เช่น การคัดแยกข้อมูล Error จาก Log ของลูกค้า หรือการแปลงรูปแบบวันที่และเวลาให้สอดคล้องกับรายงานที่ต้องนำเสนอ



ภาพที่ 2-8 JavaScript Logo

2. React

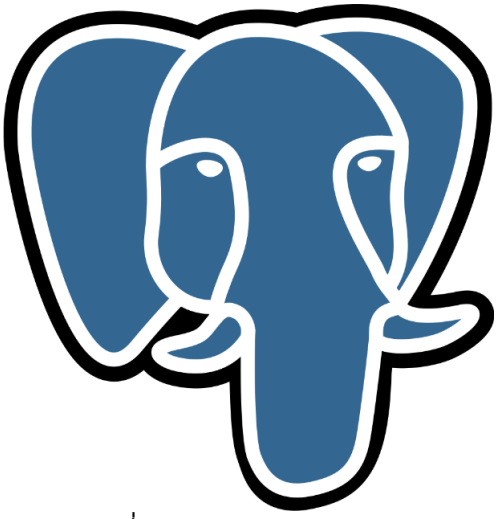
React เป็น JavaScript Library ที่ใช้สำหรับสร้างส่วนติดต่อผู้ใช้ (User Interface) ในรูปแบบ Component-based ซึ่งเน้นการประมวลผลที่รวดเร็วและยืดหยุ่น โดยได้นำมาใช้ในการพัฒนาหน้าจอแสดงผล (Dashboard) เพื่อดึงข้อมูลสถิติที่ได้จากการทำ Monitoring และ Health Check มาแสดงผลในรูปแบบกราฟหรือตารางที่ดูง่าย ช่วยให้พนักงานพี่เลี้ยงสามารถมองเห็นภาพรวมสถานะของระบบลูกค้าได้ในทันที



ภาพที่ 2-9 React Logo

3. PostgreSQL

PostgreSQL เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) ที่มีความเสถียรและรองรับการจัดการข้อมูลขนาดใหญ่ได้อย่างมีประสิทธิภาพ ในการปฏิบัติงานได้นำ PostgreSQL มาใช้เป็นที่จัดเก็บข้อมูลหลัก (Centralized Database) สำหรับบันทึกสถานะการทำ Monitoring รายวันและข้อมูลการใช้ทรัพยากรรายเดือน เพื่อใช้เป็นแหล่งข้อมูลในการวิเคราะห์แนวโน้มและจัดทำรายงาน Health Check ส่งให้ลูกค้า



ภาพที่ 2-10 PostgreSQL Logo

2.3.3 เครื่องมือที่ช่วยอำนวยความสะดวกในการดำเนินงานเอกสาร

1. Microsoft Word

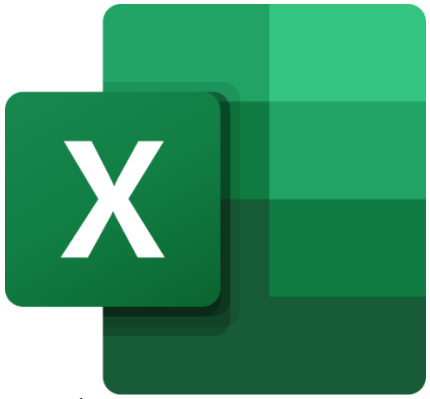
Microsoft Word เป็นโปรแกรมประมวลผลคำ (Word Processing) ที่ใช้ในการจัดทำเอกสารและรายงานต่าง ๆ โดยผู้ปฏิบัติงานได้นำมาใช้ในการจัดทำเอกสารคู่มือการปฏิบัติงาน (Operation Manual) รวมถึงการจัดทำเล่มรายงานสรุปผลการดำเนินงานสหกิจศึกษา และเอกสารประกอบการวิเคราะห์ระบบเพื่อให้มีความเป็นระเบียบและเป็นไปตามมาตรฐานขององค์กร



ภาพที่ 2-11 Microsoft Word Logo

2. Microsoft Excel

Microsoft Excel เป็นโปรแกรมประเภทตารางคำนวณ (Spreadsheet) ที่มีความสามารถในการจัดการข้อมูลตัวเลข การสร้างสูตรคำนวณ และการสร้างแผนภูมิ ในการปฏิบัติงานนี้ ผู้ปฏิบัติงานได้นำ Microsoft Excel มาใช้เป็นเครื่องมือหลักในการจัดเก็บข้อมูล (Data Storage) ที่ได้จากการทำ Morning Monitoring ในแต่ละวัน เพื่อรวบรวมสถิติการทำงานของเครื่องเซิร์ฟเวอร์ลูกค้าอย่างเป็นระบบ ช่วยให้สามารถตรวจสอบข้อมูลย้อนหลังและนำข้อมูลมาวิเคราะห์เป้าหมายเพื่อใช้ในการทำรายงาน Health Check ได้อย่างแม่นยำ



ภาพที่ 2-12 Microsoft Excel Logo

3. Microsoft PowerPoint

Microsoft PowerPoint เป็นโปรแกรมประยุกต์ด้านการนำเสนอ (Presentation Software) ที่มีความสามารถในการจัดทำสื่อประสมทั้งในรูปแบบข้อความ กราฟ ภาพนิ่ง และตารางข้อมูล ในการปฏิบัติงานนี้ ผู้ปฏิบัติงานได้นำ Microsoft PowerPoint มาใช้เป็นเครื่องมือหลักในการรวบรวมสรุปผลการทำ Health Check รายเดือน เพื่อจัดทำเล่มรายงาน (Report) ส่งให้แก่พนักงานพี่เลี้ยงและลูกค้า โดยเน้นการนำข้อมูลสถิติที่ประมวลผลได้มานำเสนอในรูปแบบแผนภูมิต่าง ๆ เพื่อให้ง่ายต่อการวิเคราะห์สถานะการใช้งานทรัพยากรระบบ



ภาพที่ 2-13 Microsoft PowerPoint Logo

4. Microsoft Outlook

Microsoft Outlook เป็นระบบบริหารจัดการอีเมลและการสื่อสารภายในองค์กรที่ใช้เป็นช่องทางหลักในการรับ-ส่งข้อมูลระหว่างผู้ปฏิบัติงาน พนักงานพี่เลี้ยง และลูกค้า โดยมีการนำมาใช้ในการนัดหมายการทำ Health Check และการจัดส่งสรุปผลการรายงานผ่านอีเมลอย่างเป็นทางการ เพื่อให้เกิดบันทึกหลักฐานการปฏิบัติงานที่ชัดเจน



ภาพที่ 2-14 Microsoft Outlook Logo

5. Line

LINE เป็นแอปพลิเคชันสำหรับการสื่อสารที่รวดเร็วและมีประสิทธิภาพ ในโครงงานนี้ได้นำ LINE มาใช้ร่วมกับระบบ Automation ผ่าน LINE Notify หรือ LINE Messaging API เพื่อทำหน้าที่เป็นระบบการแจ้งเตือน (Notification System) เมื่อระบบตรวจพบความผิดปกติ (Error) จากการทำ Morning Monitoring ทำให้ผู้ปฏิบัติงานสามารถรับทราบปัญหาและเข้าแก้ไขสถานการณ์ได้ทันท่วงที รวมถึงใช้ในการสื่อสารประสานงานภายในทีมปฏิบัติงาน



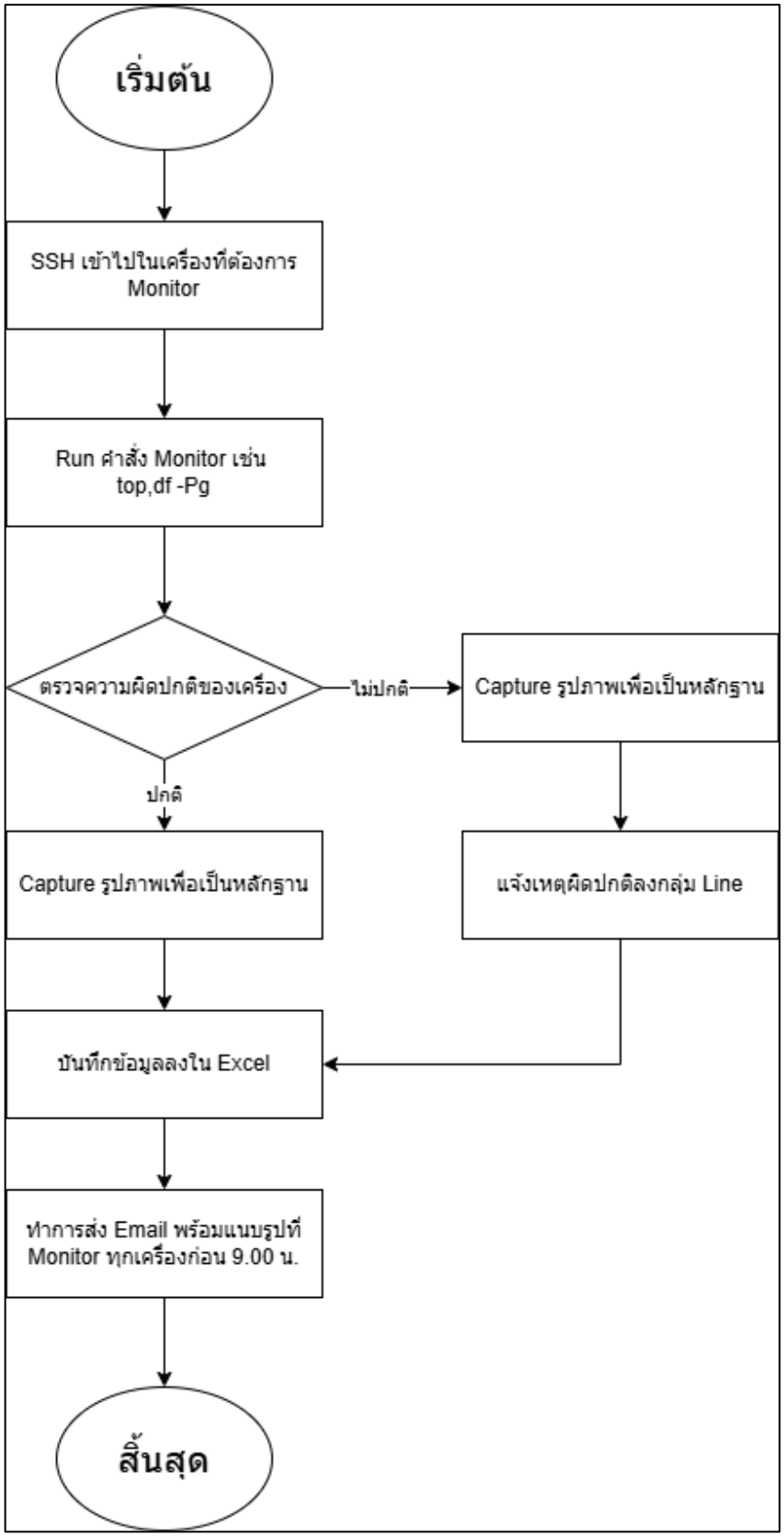
ภาพที่ 2-15 Line Logo

บทที่ 3
รายละเอียดของการปฏิบัติงาน

ในส่วนนี้จะอธิบายถึงกระบวนการและขั้นตอนการดำเนินงานโดยละเอียด ในระหว่างการปฏิบัติงานสหกิจศึกษาตำแหน่ง Cloud Engineer ซึ่งได้รับมอบหมายให้พัฒนาโครงการ Automate Daily Monitoring อันเป็นระบบตรวจสอบสถานะการทำงานของโครงสร้างพื้นฐานและซอฟต์แวร์แบบอัตโนมัติ เพื่อช่วยลดภาระงานของพนักงานและเพิ่มความแม่นยำในการติดตามสถานะระบบรายวัน โดยมีลำดับขั้นตอนดังนี้

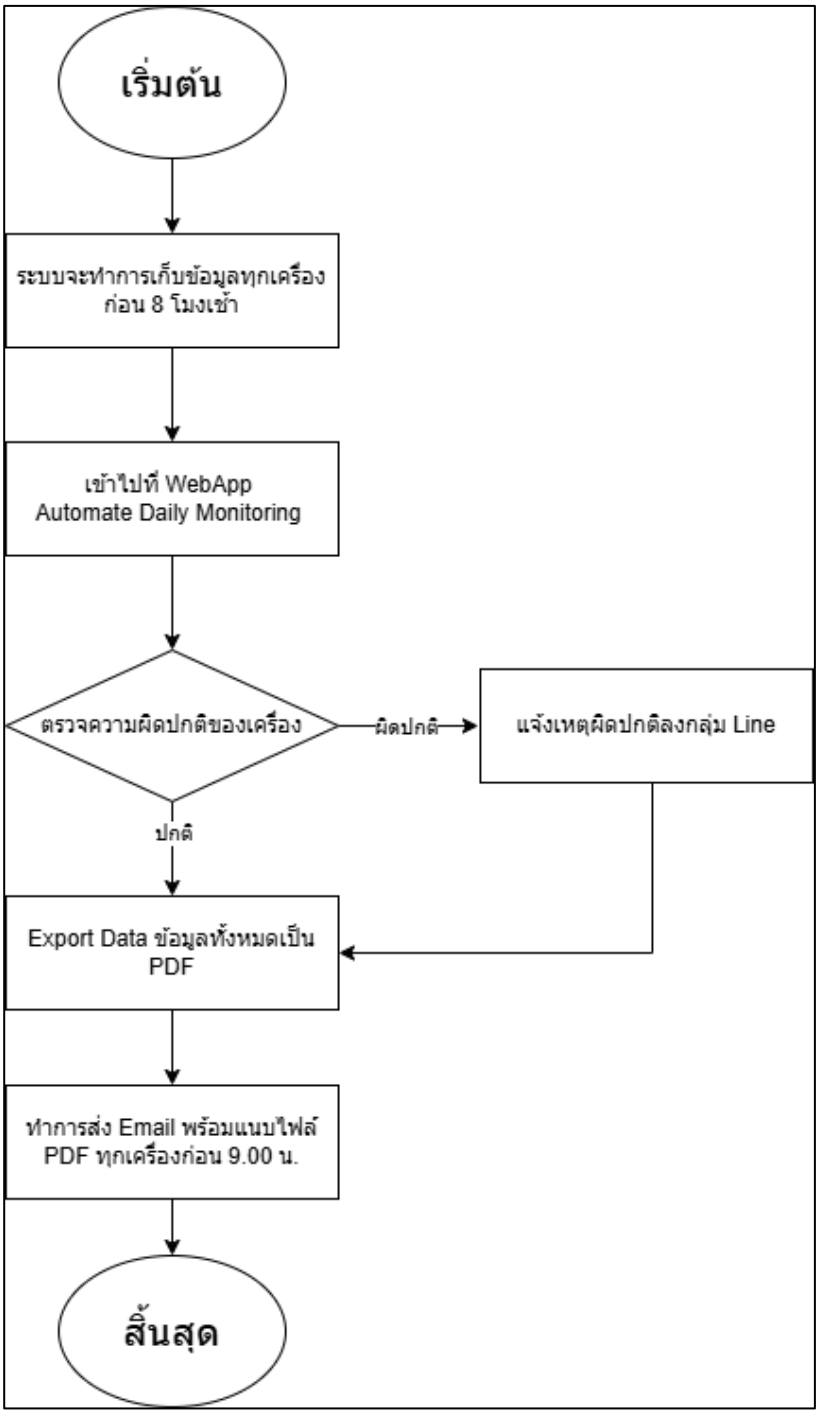
3.1 วิเคราะห์และออกแบบระบบ Automate Daily Monitoring

การวิเคราะห์ระบบสารสนเทศของโครงการนี้ เป็นการศึกษากระบวนการตรวจสอบสถานะระบบเดิมที่ต้องอาศัยแรงงานคน (Manual) เพื่อนำมาออกแบบใหม่ให้เป็นระบบอัตโนมัติ โดยระบบต้องสามารถเชื่อมต่อไปยังเซิร์ฟเวอร์หรือบริการต่างๆ ที่ทำงานอยู่ (เช่น ภายใน Docker Containers) เพื่อตรวจสอบสถานะ (Health Check) รวบรวมข้อมูล และส่งต่อมาแสดงผลบนหน้า Web Application ได้อย่างถูกต้อง



ภาพที่ 3-1 Flowchart ขั้นตอนการทำ Daily Monitoring

จากภาพที่ 3-1 จะเห็นว่าขั้นตอนการตรวจสอบประกอบด้วยการทำซ้ำ (Loop) ในเซิร์ฟเวอร์แต่ละเครื่อง ซึ่งต้องใช้เวลาดำเนินการนานและมีความเสี่ยงที่ข้อมูลจะตกหล่นหรือผิดพลาด คณะผู้จัดทำจึงได้ออกแบบสถาปัตยกรรมระบบ Automate Daily Monitoring ขึ้นใหม่ เพื่อรวบกระบวนการทั้งหมดให้ทำงานอัตโนมัติ ซึ่งสามารถแสดงขั้นตอนการทำงานใหม่ได้ดังภาพที่ 3-2



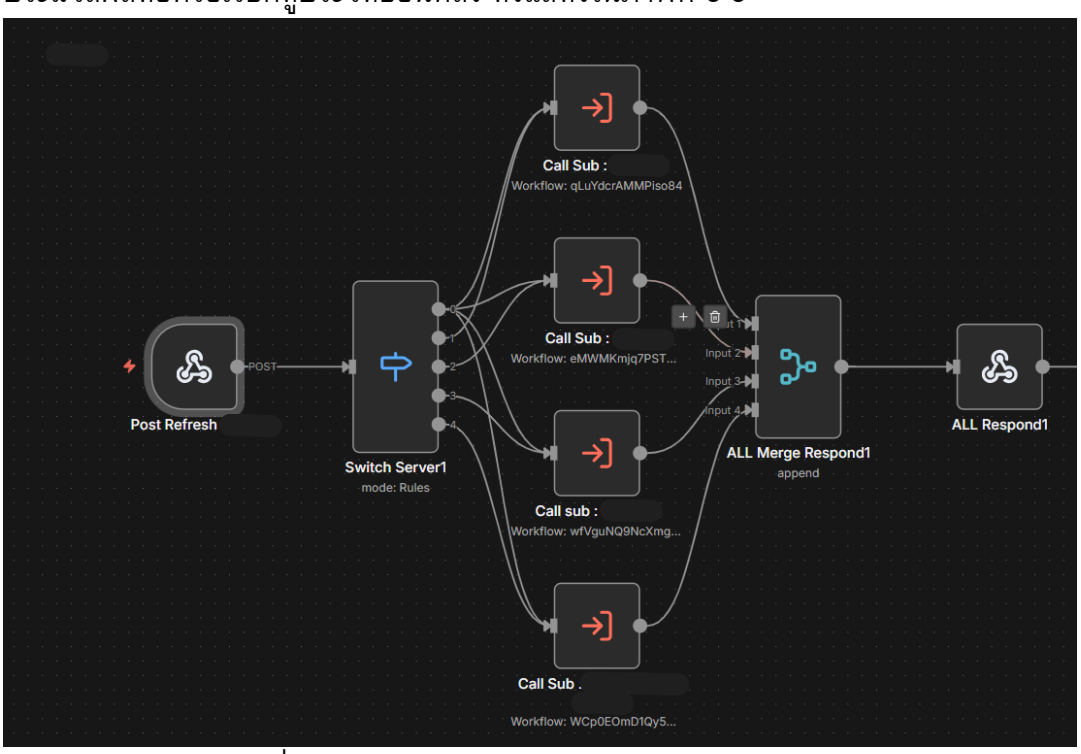
ภาพที่ 3-2 Flowchart ขั้นตอนการทำ Daily Monitoring ด้วยระบบ Automate

3.2 ขั้นตอนการออกแบบ Back-end ของ Automate Daily Monitoring

ระบบ Back-end ของ Automate Daily Monitoring ถูกออกแบบขึ้นโดยมุ่งเน้นการนำขั้นตอนที่ต้องดำเนินการซ้ำ (Routine Task) ในการตรวจสอบระบบประจำวันมาเปลี่ยนเป็นกระบวนการอัตโนมัติ (Automation) ทั้งนี้ ได้เลือกใช้แพลตฟอร์ม n8n เป็นเครื่องมือหลัก เนื่องจากรองรับการเชื่อมต่อผ่านโปรโตคอล SSH ซึ่งช่วยให้สามารถดึงข้อมูลสถานะการทำงานของเซิร์ฟเวอร์ได้โดยตรง ลดความซับซ้อนในการเขียนโค้ดลงได้อย่างมาก โดยโฟลว์การทำงาน (Workflow) ของระบบ Back-end แบ่งออกเป็นส่วนหลัก ๆ ดังนี้

3.2.1 Refresh Data

กระบวนการ Refresh Data ถือเป็นหัวใจหลักของระบบ ทำหน้าที่รับคำสั่งผ่าน API เพื่อเชื่อมต่อ SSH เข้าไปยังเซิร์ฟเวอร์เป้าหมาย (สามารถเลือกสั่งการเฉพาะเจาะจงบางเครื่อง หรือสั่งการทุกเครื่องพร้อมกันได้) จากนั้นระบบจะทำการรวบรวมข้อมูลสถานะประจำวันและจัดเก็บในรูปแบบไฟล์ JSON และ Text เพื่อให้ง่ายต่อการนำข้อมูลไปประมวลผลต่อหรือเรียกดูประวัติย้อนหลัง ดังแสดงในภาพที่ 3-3

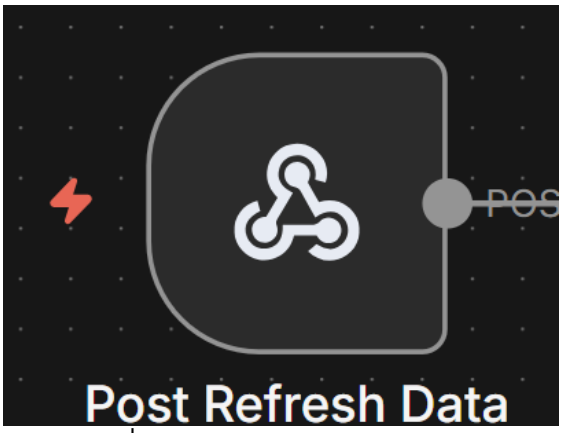


ภาพที่ 3-3 ตัวอย่าง Flowchart Backend Refresh Data

จากภาพจะเห็นว่าระบบเริ่มต้นด้วย Webhook Node ซึ่งทำหน้าที่รับ Request แบบ POST เพื่อพิจารณาเงื่อนไขว่าต้องการอัปเดตข้อมูลของเครื่องใด จากนั้นระบบจะใช้ Switch Node เพื่อแยกสายการทำงาน และเรียกใช้งาน Sub-Workflow ให้ดำเนินการเชื่อมต่อ SSH ข้อมูลการตอบกลับ (Response) ทั้งหมดจะถูกรวบรวมไว้เพื่อใช้ในการแจ้งเตือนสถานะการทำงานหรือข้อผิดพลาดที่เกิดขึ้น โดยมีรายละเอียดการทำงานของแต่ละ Node ดังนี้

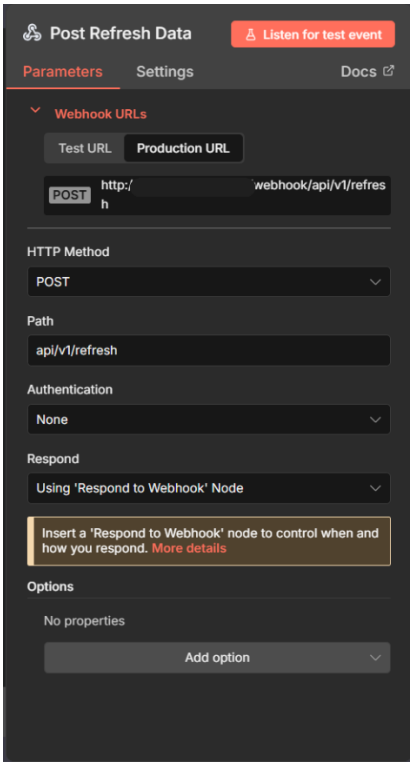
3.2.1.1 Post Refresh Data Node

ทำหน้าที่เป็น Webhook รับข้อมูล Request แบบ POST เพื่อเป็นจุดเริ่มต้นการทำงาน (Trigger) ของระบบอัตโนมัติในการสั่งอัปเดตข้อมูล



ภาพที่ 3-4 Post Refresh Data Node

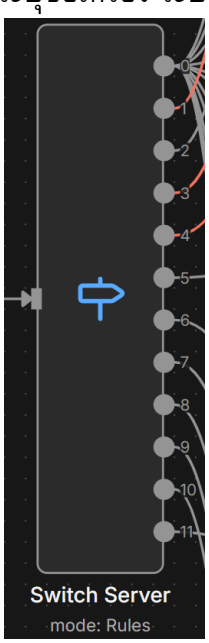
โดยมีการกำหนดค่า Parameter ของ Node โดยเลือก Method เป็น POST และกำหนด Path สำหรับการเรียกใช้งาน ดังภาพที่ 3-5



ภาพที่ 3-5 Config Post Refresh Data Node

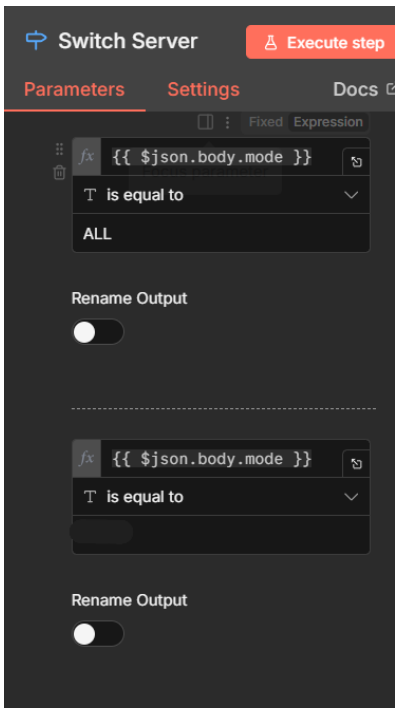
3.2.1.2 Switch Server Node

ทำหน้าที่ตรวจสอบเงื่อนไขและกำหนดเส้นทางการทำงาน (Routing) ของ Node ถัดไป โดยอ้างอิงจากข้อมูล Input ที่ได้รับจาก Webhook เช่น หากระบุค่าเป็น "All" จะเป็นการสั่งอัปเดตทุกเซิร์ฟเวอร์ หรือหากระบุชื่อเครื่อง ระบบก็จะทำงานเฉพาะเซิร์ฟเวอร์นั้น ๆ



ภาพที่ 3-6 Switch Server Node

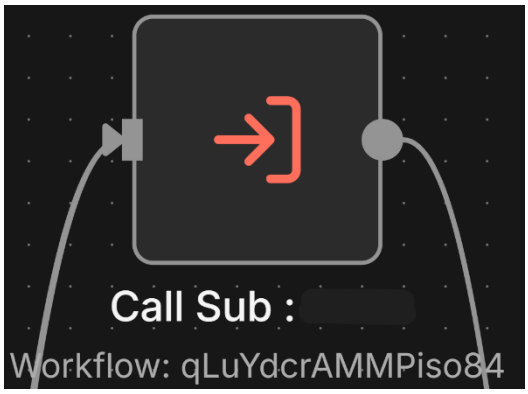
โดยมีการกำหนดเงื่อนไข (Condition) ใช้รูปแบบ "is equal to" เพื่อตรวจสอบ Input ที่เข้ามาว่าตรงกับเงื่อนไขใด ดังภาพที่ 3-7



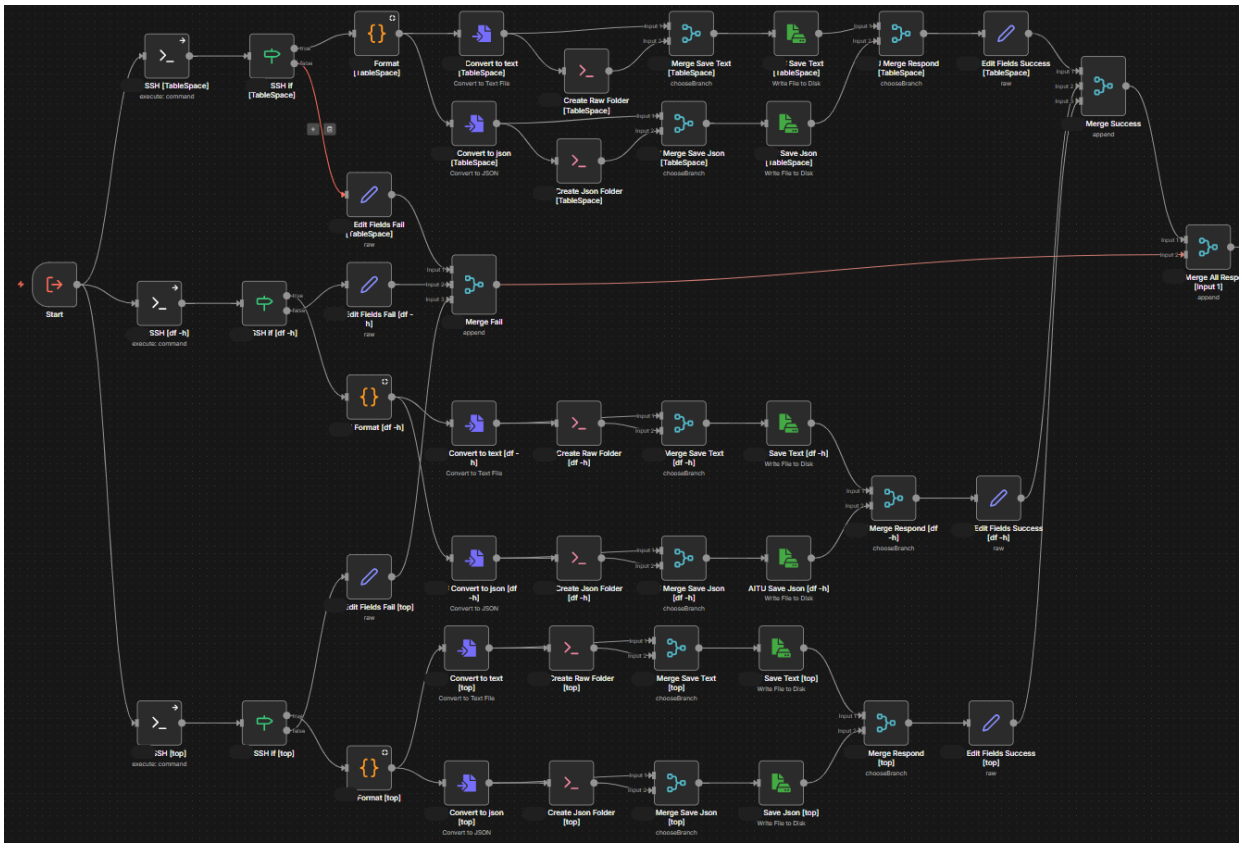
ภาพที่ 3-7 Config Switch Server Node

3.2.1.3 Sub Workflow SSH Node

เป็น Workflow ย่อยของกระบวนการ Refresh Data ทำหน้าที่ดำเนินการเชื่อมต่อ SSH และประมวลผลคำสั่ง ภายใน Sub-Workflow นี้จะทำการแปลงข้อมูล (Parse) ผลลัพธ์ให้อยู่ในรูปแบบ JSON และจัดเก็บลงในระบบเซิร์ฟเวอร์ของ n8n โดยมีการสร้างโฟลเดอร์แยกตามชื่อเครื่อง ปี เดือน วัน และประเภทไฟล์ (Text/JSON) อย่างเป็นระบบ พร้อมทั้งส่งสถานะการทำงานกลับไปยัง Workflow หลักเพื่อทำการแจ้งเตือน



ภาพที่ 3-8 Sub Workflow SSH Node



ภาพที่ 3-9 ภายใน Sub Workflow SSH Node

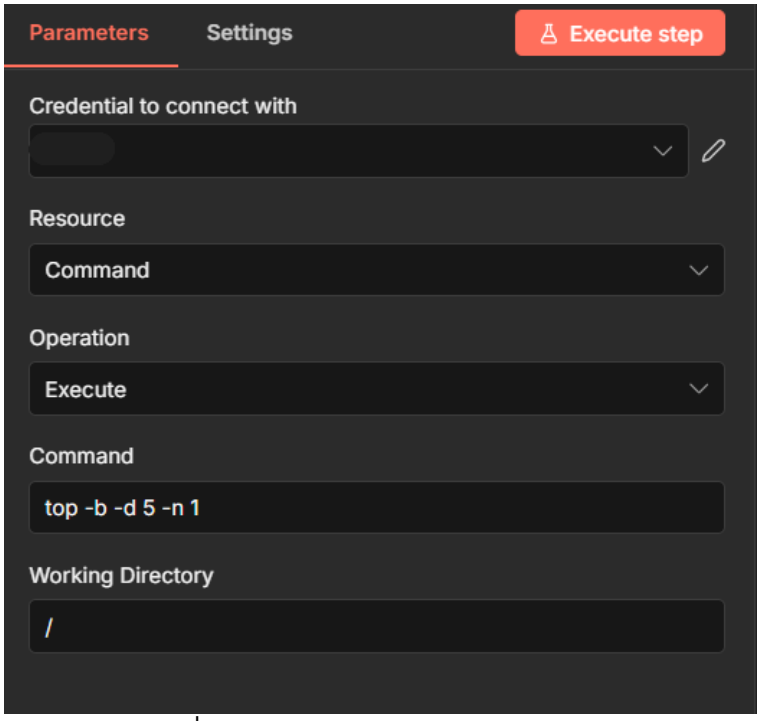
3.2.1.3.1 SSH Command Node

ทำหน้าที่เชื่อมต่อ SSH เข้าไปยังเซิร์ฟเวอร์เป้าหมายผ่าน Credential ที่กำหนด และรันคำสั่ง (Command) เพื่อดึงข้อมูลสถานะระบบกลับมาในรูปแบบข้อความ (Text) โดยแต่ละเครื่องอาจใช้คำสั่งที่แตกต่างกัน



ภาพที่ 3-10 SSH Command top Node

โดยกำหนดค่าให้รันคำสั่งผ่าน Resource และป้อนคำสั่งที่ต้องการลงในช่อง Command ดังภาพที่ 3-11



ภาพที่ 3-11 Config SSH Command Node

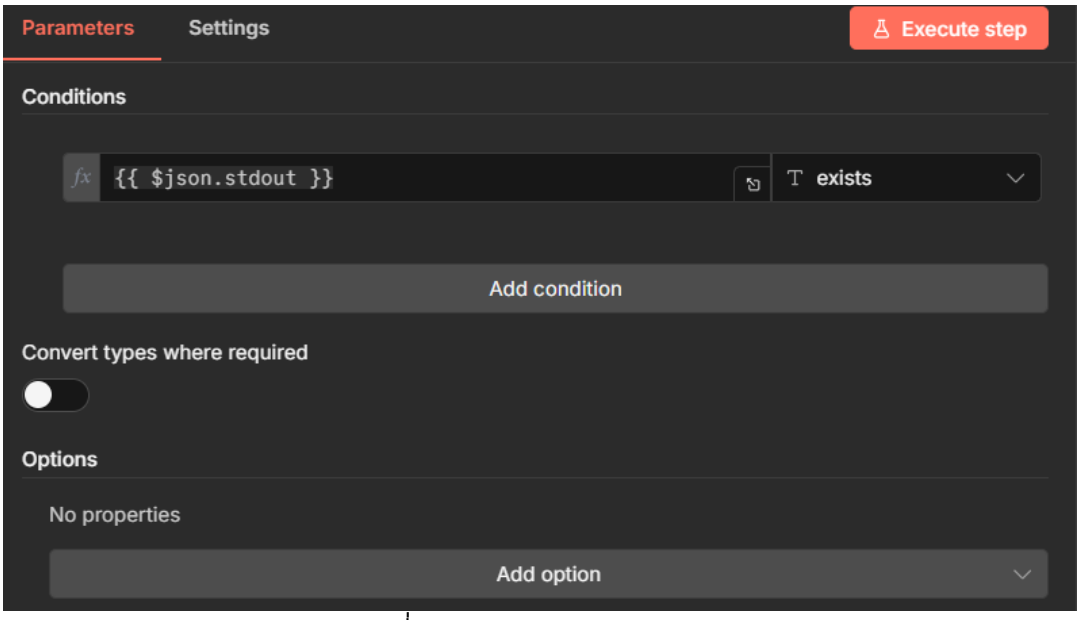
3.2.1.3.2 IF SSH Node

ทำหน้าที่ตรวจสอบว่า SSH Command Node มีผลลัพธ์ (Output) ส่งกลับมาหรือไม่ หากไม่มีผลลัพธ์ระบบจะข้ามไปที่ Edit Fields Fail Node เพื่อแจ้งเตือนข้อผิดพลาดในการเชื่อมต่อ แต่หากมีผลลัพธ์ ระบบจะส่งข้อมูลต่อไปยัง Parse Data Node



ภาพที่ 3-12 IF SSH Node

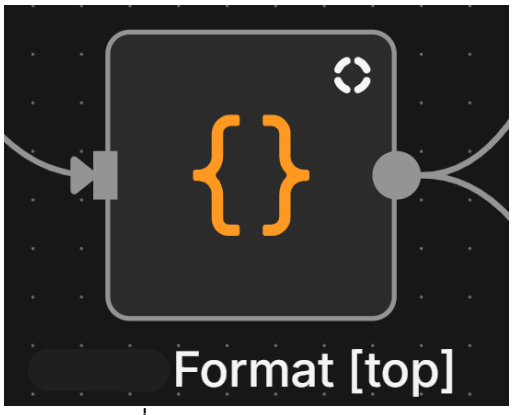
โดยใช้เงื่อนไข "exists" เพื่อตรวจสอบความมีอยู่ของผลลัพธ์ที่ได้จาก Node ก่อนหน้า ดังภาพที่ 3-13



ภาพที่ 3-13 Config IF SSH Node

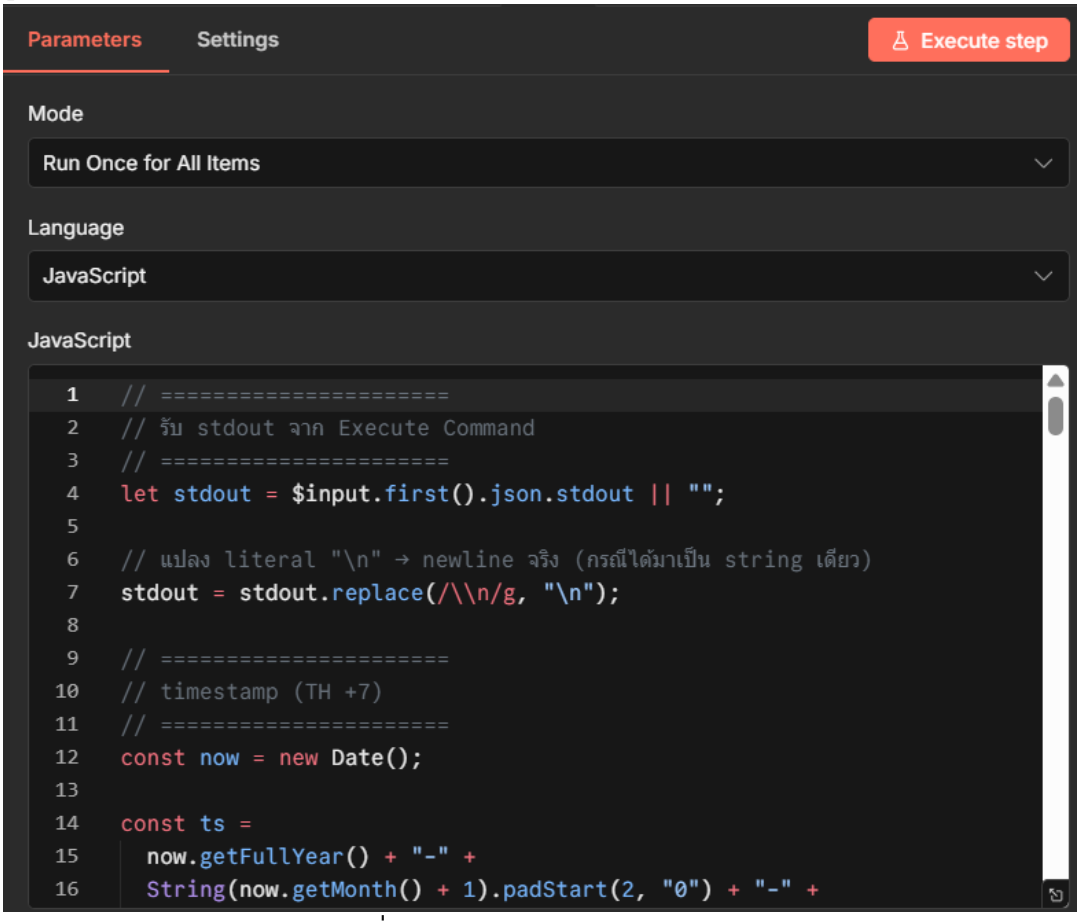
3.2.1.3.3 Parse Data Node

ทำหน้าที่แปลงผลลัพธ์ที่เป็นข้อความ (Text) ให้กลายเป็นรูปแบบ JSON เพื่อให้ง่ายต่อการนำข้อมูลไปใช้งานต่อ



ภาพที่ 3-14 Parse Data Node

โดยกำหนดภาษาที่ใช้ประมวลผลเป็น JavaScript และเขียนสคริปต์ให้สอดคล้องกับรูปแบบข้อความที่ได้รับจากแต่ละคำสั่ง ดังภาพที่ 3-15



ภาพที่ 3-15 Config Parse Data Node

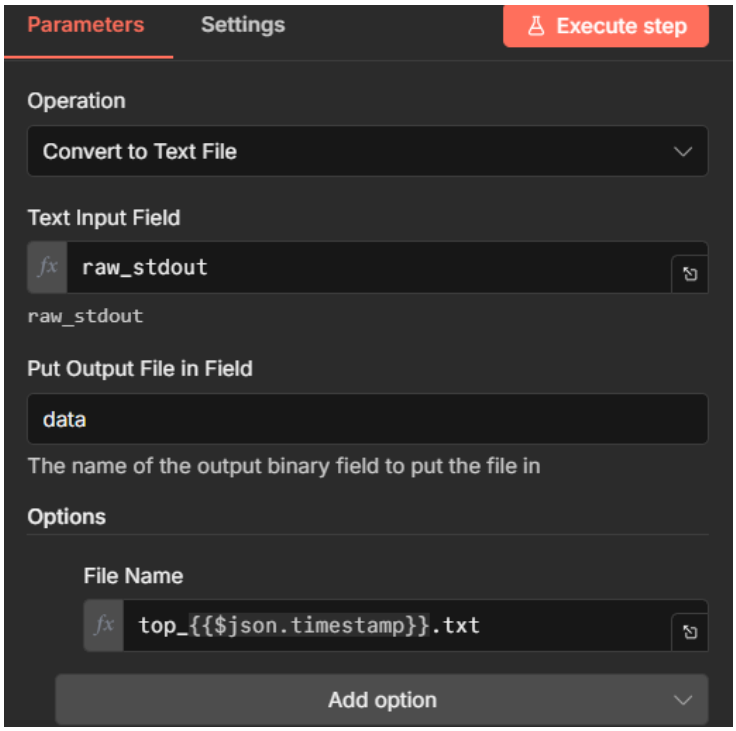
3.2.1.3.4 Convert Data Node

ทำหน้าที่แปลงชุดข้อมูลให้อยู่ในรูปแบบไฟล์ที่พร้อมบันทึก ทั้งสกุลไฟล์ .txt และ .Json



ภาพที่ 3-16 Convert Data Node

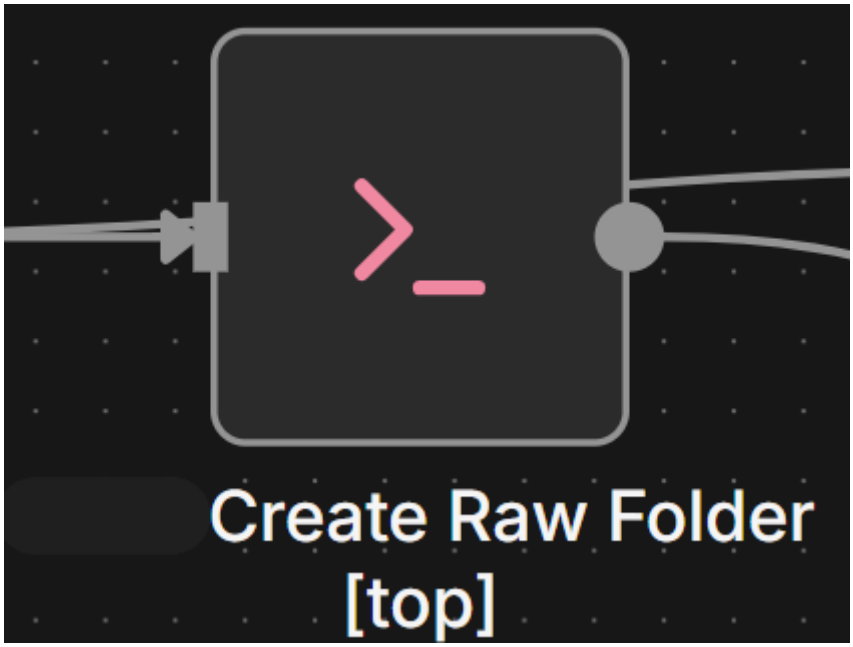
โดยมีการกำหนดให้เลือก Covert to Text และ Json ส่วนของ Input จะใส่ตัวแปรที่ต่างกันในที่นี้คือ Raw และ Json และกำหนดชื่อไฟล์โดยใช้ชื่อ Command_ตัวแปรของเวลาและวันที่.txt ในส่วนของ Json จะเป็นในรูปแบบ Command_ตัวแปรของเวลาและวันที่.Json ดังภาพที่ 3-17



ภาพที่ 3-17 Config Convert Data Node

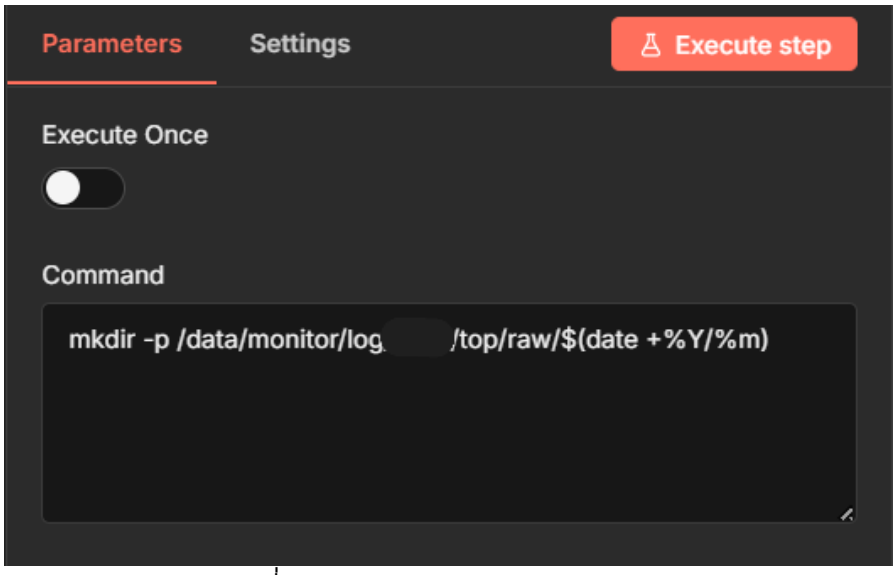
3.2.1.3.5 Create Folder Node

ทำหน้าที่สร้างโฟลเดอร์สำหรับจัดเก็บไฟล์โดยอัตโนมัติ



ภาพที่ 3-18 Create Folder Node

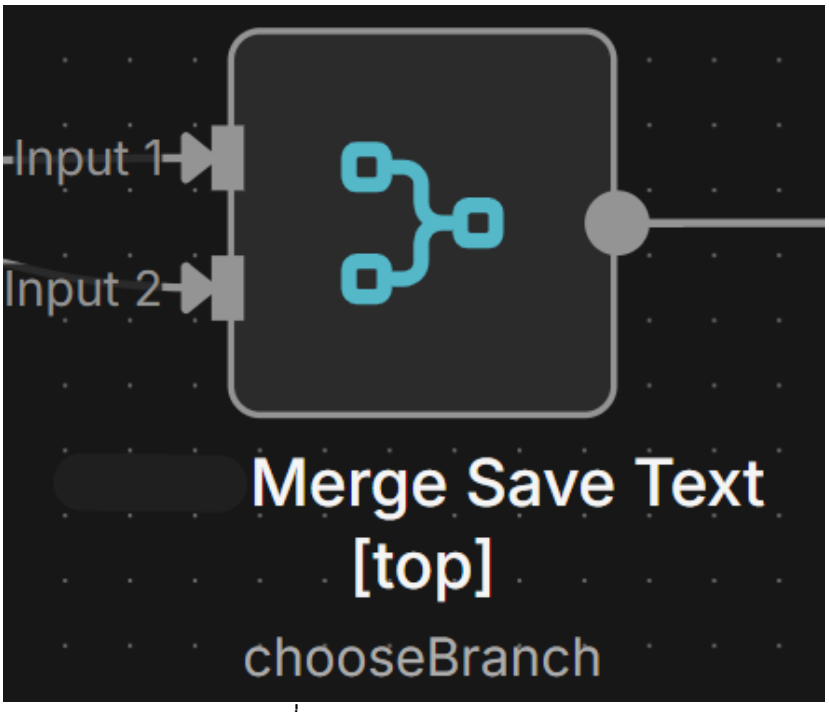
โดยใช้คำสั่งสร้างไดเรกทอรี (mkdir) และจัดโครงสร้าง Path ตาม: ชื่อเครื่อง > คำสั่ง > ประเภทไฟล์ > ปี > เดือน > วัน เพื่อความเป็นระเบียบ ดังภาพที่ 3-19



ภาพที่ 3-19 Config Create Folder Node

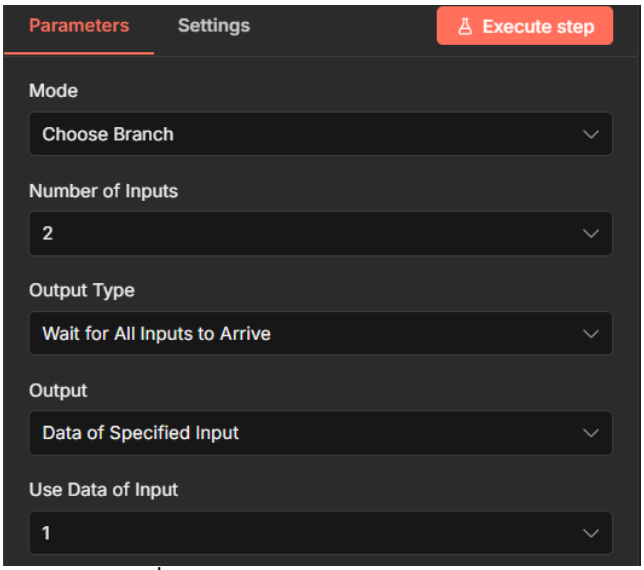
3.2.1.3.6 Merge Save Node

ทำหน้าที่รอและรวบรวมข้อมูล (Wait for all inputs to arrive) จาก Node การประมวลผลและการสร้างโฟลเดอร์ เพื่อป้องกันปัญหาข้อมูลสูญหายก่อนเข้าสู่ขั้นตอนการบันทึกไฟล์



ภาพที่ 3-20 Merge Save Node

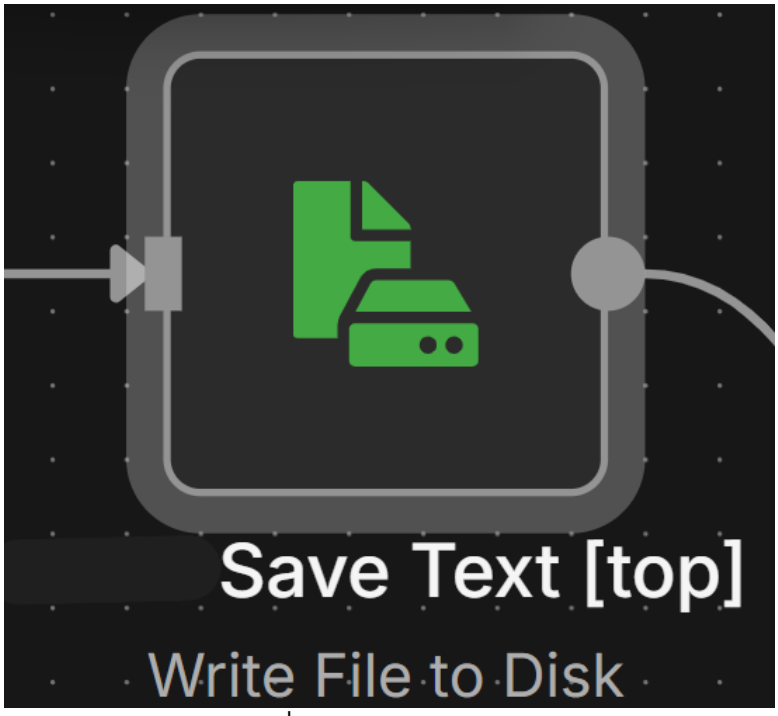
โดยมีการกำหนด Mode Choose Branch , Input 2 โดยให้ Output Type โดยเลือก Wait for All inputs to Arrive เพื่อให้จำทำงานหลัง 2 Node ที่ Input เข้ามาทำงานเสร็จสิ้นแล้ว และกำหนดให้ Output เป็น Data of Specified Input เพื่อกำหนด Output เป็น Node 1 ที่มี Data เพื่อเข้าไปใน Save File Node ดังภาพ 3-21



ภาพที่ 3-21 Config Merge Save Node

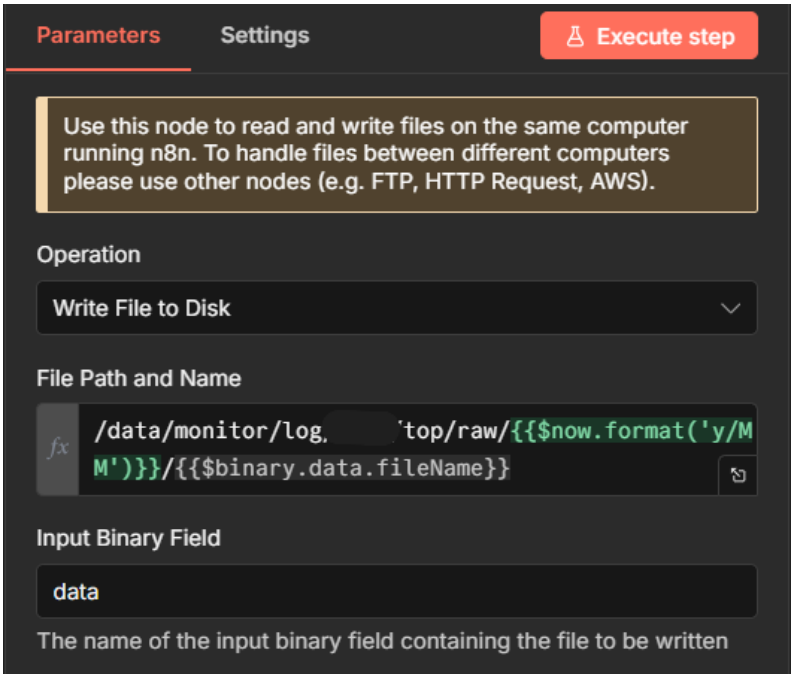
3.2.1.3.7 Save File Node

ทำหน้าที่เขียนไฟล์ข้อมูล (Write File to Disk) บันทึกลงบนพื้นที่จัดเก็บของเซิร์ฟเวอร์ n8n ตาม Path ที่ได้สร้างไว้



ภาพที่ 3-22 Save File Node

โดยมีการกำหนด Operation Write File to Disk เพื่อให้เขียนไฟล์ลงในเครื่อง และกำหนด File Path and Name ตามข้อมูลก่อนหน้านี้ ดังภาพ 3-23



ภาพที่ 3-23 Config Save File Node

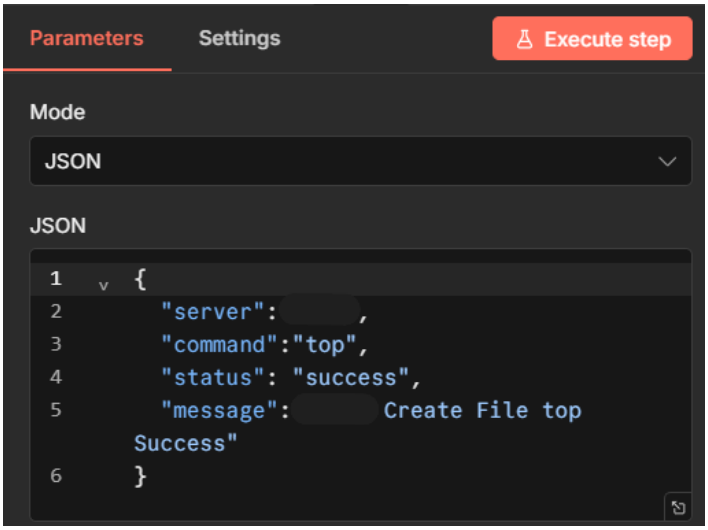
3.2.1.3.8 Edit Fields Status Node

ทำหน้าที่สร้างตัวแปรสถานะการทำงาน โดยกำหนดข้อความเป็น "Success" เมื่อบันทึกข้อมูลสำเร็จ หรือ "Fail" หากเกิดข้อผิดพลาดในการรัน SSH



ภาพที่ 3-24 Edit Fields Status Node

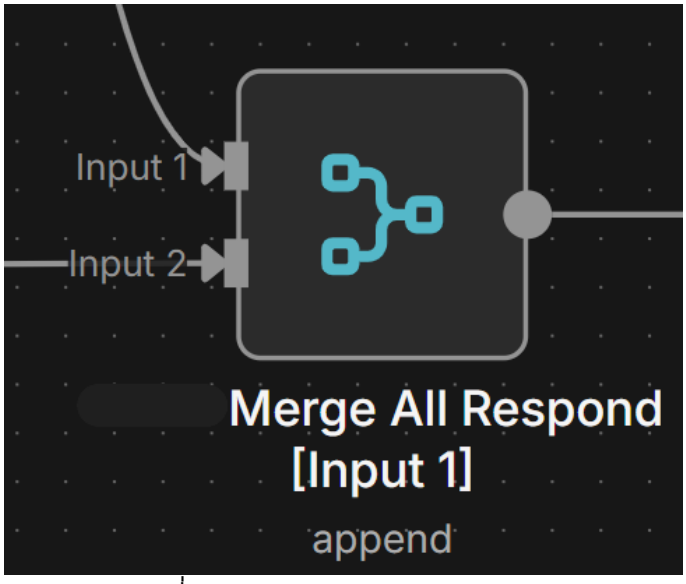
โดยมีการกำหนด Mode เป็น Json และเขียนแสดง Server Command Status และ Message โดยจะแตกต่างกันไปของ Success และ Fail ดังภาพ 3-25



ภาพที่ 3-25 Config Edit Fields Status Node

3.2.1.3.9 Merge All Respond Node

ทำหน้าที่รวบรวมข้อความสถานะการทำงานทั้งหมดให้เป็นชุดข้อมูลเดียว ก่อนที่จะส่งกลับ (Response) ออกไปยัง Workflow หลัก



ภาพที่ 3-26 Merge All Respond Node

3.2.1.4 All Respond Node

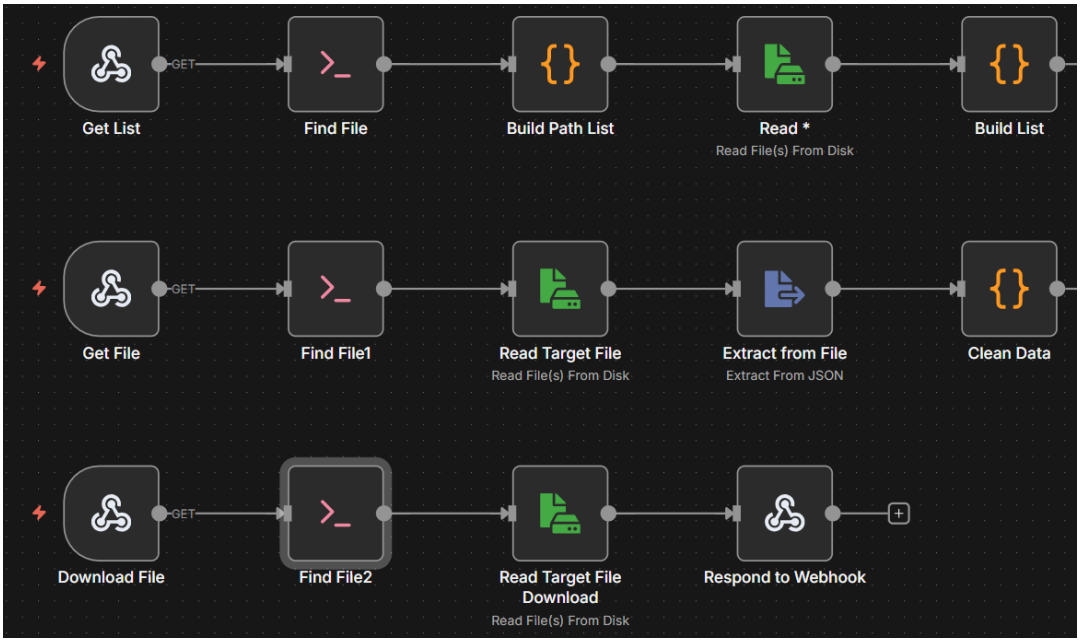
ทำหน้าที่รวบรวมการตอบกลับ (Response) ของทุกเซิร์ฟเวอร์เพื่อแสดงผลลัพธ์ภาพรวมหลังจากที่ระบบได้รับการเรียกใช้งาน ทำให้ผู้ปฏิบัติงานทราบถึงสถานะการทำงานทั้งหมดเพื่อใช้ประกอบการแจ้งเตือน



ภาพที่ 3-27 All Respond Node

3.2.2 File Log Monitoring

File Log Monitoring เป็นชุด Workflow ที่ทำหน้าที่เสมือน API ในการจัดการไฟล์ข้อมูลที่ระบบจัดเก็บไว้ โดยแบ่งการทำงานออกเป็น 3 ส่วนหลัก ได้แก่ การเรียกดูรายชื่อไฟล์ทั้งหมด (List File), การขอดูข้อมูลไฟล์ (Get File) และการดาวน์โหลดไฟล์ (Download File)



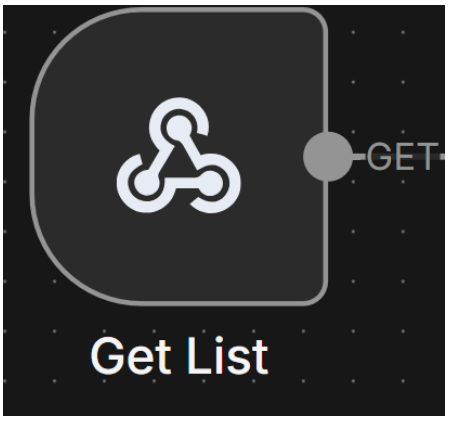
ภาพที่ 3-28 Flowchart Backend File Log Monitoring

3.2.2.1 List File

เป็นการค้นหารายชื่อไฟล์ทั้งหมดในระบบ เพื่อให้ผู้ใช้งานสามารถเลือกดูข้อมูลจากไฟล์เป้าหมายที่ต้องการได้

3.2.2.1.1 Get List Node

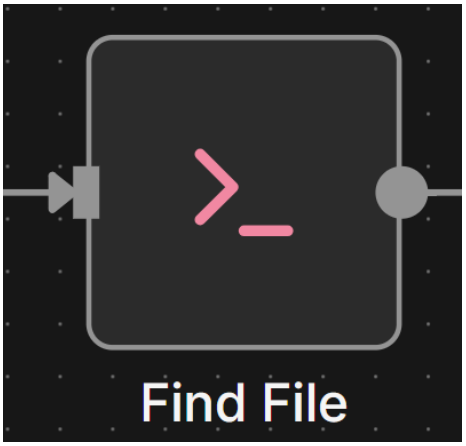
ทำหน้าที่เป็น Webhook รับ Request แบบ GET เพื่อเริ่มต้นกระบวนการร้องขอรายชื่อไฟล์



ภาพที่ 3-29 Get List Node

3.2.2.1.2 Find File Node

รับพารามิเตอร์ API และไปรันคำสั่ง find ในระบบปฏิบัติการเพื่อค้นหาไฟล์ใน Directory



ภาพที่ 3-30 Find File Node

3.2.2.1.3 Build Path List Node

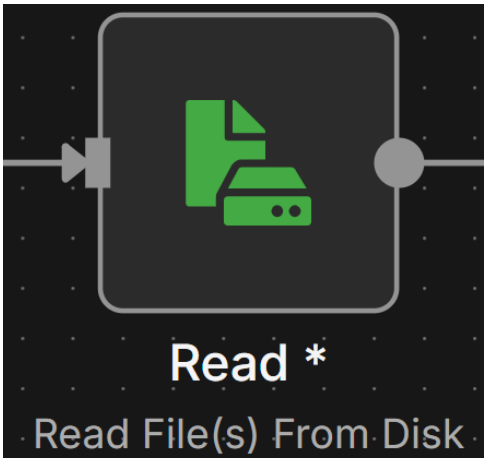
นำผลลัพธ์ข้อความที่ได้จากการค้นหามาแยกส่วนประกอบของ Path (เช่น ปี, เดือน, ชื่อไฟล์) เพื่อจัดโครงสร้างให้อยู่ในรูปแบบ JSON



ภาพที่ 3-31 Build Path List Node

3.2.2.1.3 Read All Node

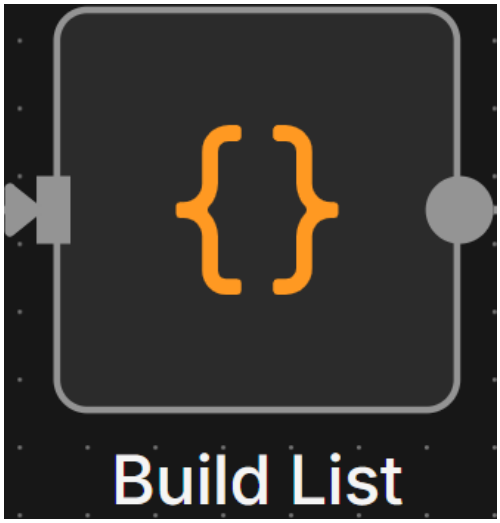
ทำหน้าที่อ่านข้อมูลไฟล์ทั้งหมดจาก Path ที่ระบบค้นพบในขั้นตอนก่อนหน้า



ภาพที่ 3-32 Read All Node

3.2.2.1.3 Build List Node

จัดเตรียมข้อมูลขั้นสุดท้าย โดยดึงเฉพาะข้อมูลชื่อไฟล์มารวมกันเป็นอาร์เรย์ (Array) ในรูปแบบ JSON เพื่อตอบกลับไปยังผู้ใช้งาน



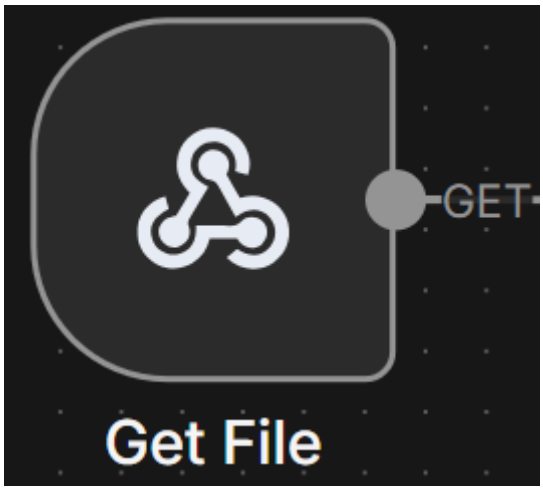
ภาพที่ 3-33 Build List Node

3.2.2.2 Get File

เป็นการนำชื่อไฟล์เป้าหมายมาค้นหา เพื่อดึงข้อมูลรายละเอียดภายในไฟล์นั้นออกมาแสดงผล

3.2.2.2.1 Get File Node

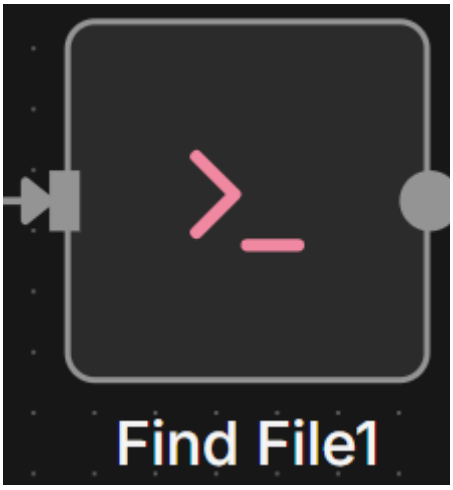
ทำหน้าที่รับ Request แบบ GET สำหรับการร้องขอดูรายละเอียดข้อมูลภายในไฟล์ที่ระบุ



ภาพที่ 3-34 Get File Node

3.2.2.2.2 Find File Node

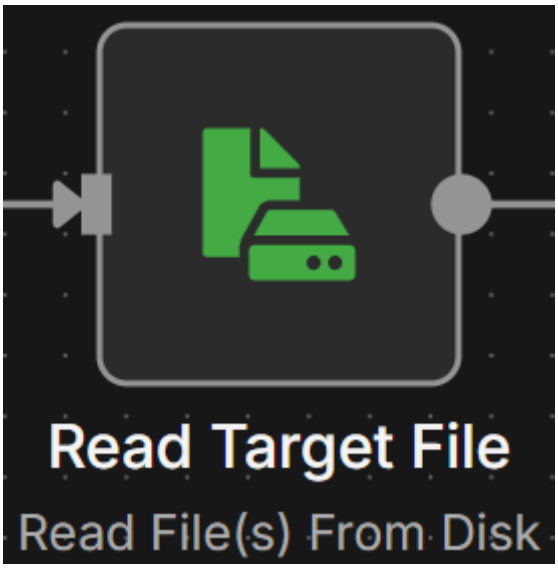
รันคำสั่ง find โดยระบุชื่อไฟล์แบบเจาะจง เพื่อค้นหา Path ที่แน่นอนของไฟล์เป้าหมาย



ภาพที่ 3-35 Find File Node

3.2.2.2.3 Read Target File Node

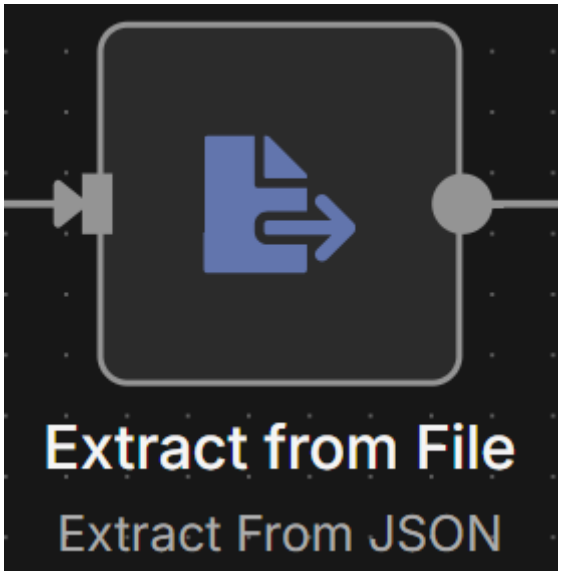
ทำการอ่านข้อมูลของไฟล์เป้าหมายจากฮาร์ดดิสก์ และนำข้อมูลมาพักไว้เพื่อรอการประมวลผล



ภาพที่ 3-36 Read Target File Node

3.2.2.2.3 Extract from File Node

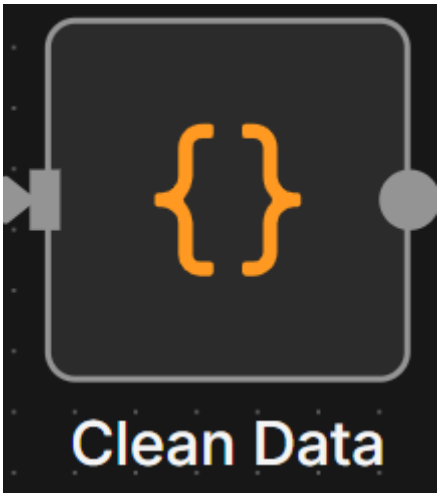
แปลงข้อมูล JSON ที่อ่านได้จากไฟล์ให้อยู่ในรูปแบบ Binary เพื่อรองรับการตอบกลับ (Response)



ภาพที่ 3-37 Extract from File Node

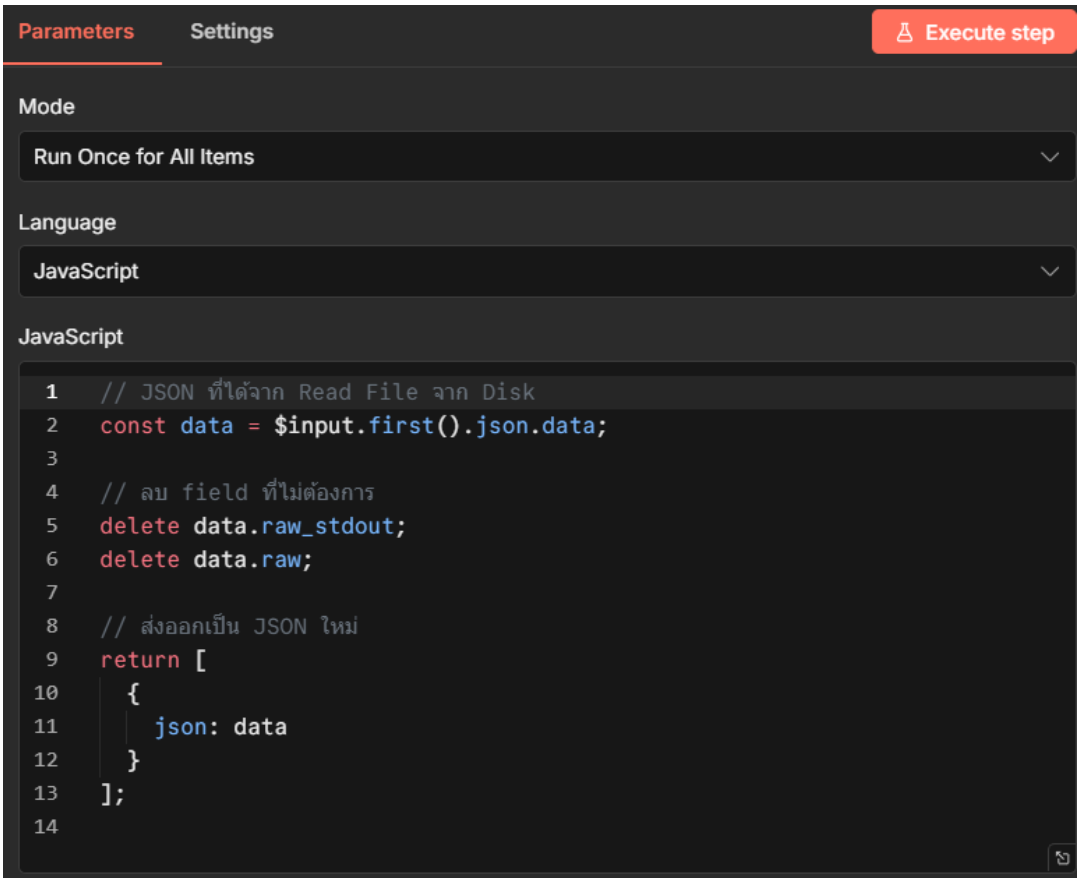
3.2.2.2.3 Clean Data Node

ทำหน้าที่กรองข้อมูล โดยลบฟิลด์ที่ไม่ได้ใช้งาน (เช่น ฟิลด์ Raw) ออกไป เพื่อให้ได้ชุดข้อมูลที่สะอาดและง่ายต่อการใช้งาน



ภาพที่ 3-38 Clean Data Node

โดยภายในจะเป็นการลบฟิลด์ของ raw ออกไปเพื่อง่ายต่อการใช้งาน



ภาพที่ 3-39 Config Clean Data Node

3.2.2.3 Download File

ระบบที่ช่วยอำนวยความสะดวกให้ผู้ใช้งานสามารถดาวน์โหลดไฟล์ Log ออกมาไว้ในเครื่องตนเองได้โดยตรง

3.2.2.2.1 Download File Node

ทำหน้าที่รับ Request สำหรับการร้องขอดาวน์โหลดไฟล์



ภาพที่ 3-40 Download File Node

3.2.2.2.2 Find File Node

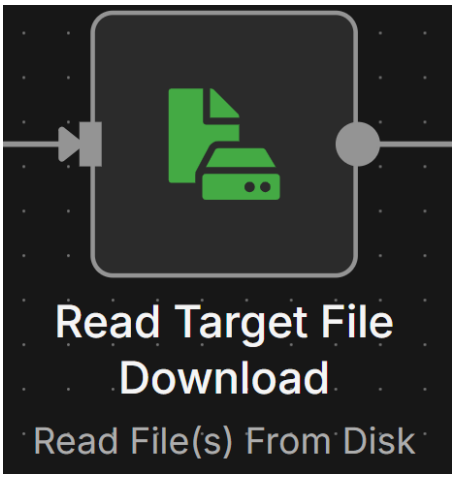
รันคำสั่ง find เพื่อหา Path ที่แน่นอนของไฟล์เป้าหมายที่จะดาวน์โหลด



ภาพที่ 3-41 Find File Node

3.2.2.2.3 Read Target File Node

ทำการอ่านไฟล์เป้าหมายจากระบบเพื่อเตรียมพร้อมสำหรับการส่งออก



ภาพที่ 3-42 Read Target File Node

3.2.2.2.3 Respond to Webhook Node

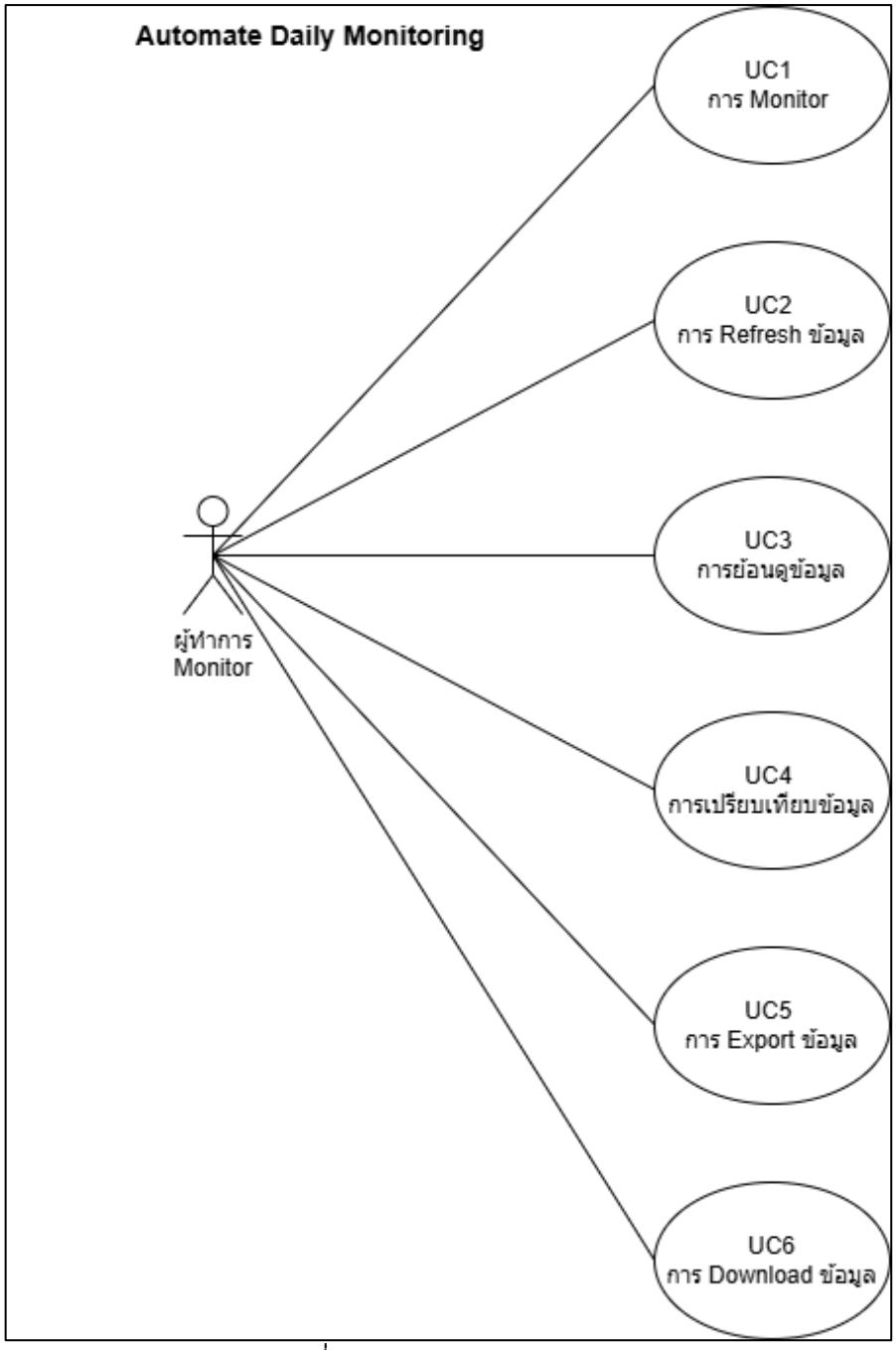
ทำหน้าที่ตอบกลับ HTTP Request พิเศษ โดยส่งข้อมูลไฟล์ในรูปแบบ Binary พร้อมทั้งกำหนด HTTP Header แบบ Content-Disposition เป็น attachment เพื่อสั่งการให้เบราว์เซอร์ของผู้ใช้งานทำการดาวน์โหลดไฟล์ลงเครื่อง แทนที่จะนำไปแสดงผลบนหน้าเว็บ



ภาพที่ 3-43 Respond to Webhook Node

3.2.3 Use Case Diagram

การออกแบบและพัฒนาระบบซอฟต์แวร์ที่มีความซับซ้อนมักจะเกี่ยวข้องกับการทำความเข้าใจความต้องการของผู้ใช้และการวางแผนการทำงานของระบบให้สอดคล้องกับความต้องการเหล่านั้นหนึ่งในเครื่องมือที่มีประสิทธิภาพในการบรรลุเป้าหมายนี้คือแผนภาพกรณีการใช้งาน Use Case Diagram ซึ่งเป็นส่วนหนึ่งของกระบวนการวิเคราะห์และออกแบบเชิงวัตถุ (OOAD) แผนภาพกรณีการใช้งานเป็นเครื่องมือสำคัญในการวิเคราะห์และออกแบบระบบซอฟต์แวร์ ซึ่งช่วยให้ผู้ปฏิบัติงานเข้าใจภาพรวมของระบบและการทำงานได้อย่างชัดเจนและเป็นระเบียบ ดังภาพที่ 3-44



ภาพที่ 3-44 Use Case Diagram

3.2.4 Use Case Description

ในการพัฒนาระบบซอฟต์แวร์ที่มีความซับซ้อนและต้องการความถูกต้องและครบถ้วนของการทำงาน การสร้าง Use Case Description เป็นขั้นตอนสำคัญที่ช่วยให้ผู้ปฏิบัติงานเข้าใจรายละเอียดของแต่ละ กรณีการใช้งาน (Use Case) อย่างลึกซึ้ง มีการแบ่งประเด็นเป็นหัวข้อต่าง ๆ อย่างชัดเจน โดยหัวข้อที่ควรอธิบายดังนี้

1. Use Case ID หมายถึง รหัสของ Use Case
2. Use Case Name หมายถึง ชื่อของ Use Case
3. Created By หมายถึง ใส่วันที่ ที่สร้าง Use Case
4. Date Created หมายถึง เหตุผลที่ต้องทำ Use Case Format ของวันที่ MM/DD/YYYY
5. Actors หมายถึง ผู้ใช้ที่เกี่ยวข้องกับ Use Case
6. Description หมายถึง บรรยายคำอธิบายสั้น ๆ ของ Use Case ถึงเหตุผลการทำรวมถึง
7. ผลลัพธ์ของการทำ Use Case นี้
8. Trigger หมายถึง อธิบายเหตุการที่ทำให้มีการใช้งาน Use Case หรืออาจเป็นกระบวนการก่อนหน้าที่ทำให้เกิดการเรียก Use Case ก็ได้
9. Preconditions หมายถึง รายการของเงื่อนไขเบื้องต้น ก่อนเข้าทำ Use Case นี้
10. Postconditions หมายถึงขั้นตอนที่ไม่เป็นไปตาม Main Flow
11. Normal Flow หมายถึง ใส่รายละเอียดของการทำงานระหว่างผู้ใช้ (User) และระบบ (System) ในการทำงานปกติของ Use Case นี้ โดยเป็นการโต้ตอบระหว่างผู้ใช้ และระบบเพื่อทำให้เป็นไปตามเป้าหมายของ Use Case
12. Alternative Flows หมายถึง อธิบายรายละเอียดของขั้นตอนที่แยกออกมาจาก Flow อื่นโดยที่จะต้องมีจุดที่แยกออกมา
13. Exceptions หมายถึง ขั้นตอนที่ไม่เป็นไปตาม Main Flow
14. Includes หมายถึง รายการ Use Case อื่นที่ถูกเรียกใช้โดย Use Case นี้
15. Frequency of Use หมายถึงความถี่ การใช้งาน Use Case

1.การ Monitor (UC1)

เป็นฟังก์ชันหลักของระบบโดยมีความสำคัญในการดูข้อมูลของแต่ละวันว่าเกิดปัญหาใดเพื่อทำการแก้และให้ระบบสามารถทำงานได้ตลอดไม่มีปัญหาดังตารางที่ 3-1

ตารางที่ 3-1 UC1 การ Monitor

Use Case ID:	UC1		
Use Case Name:	การ Monitor		
Created By:	ธาม แก้วเกษ	Last Updated By:	ธาม แก้วเกษ
Date Created:	04/03/2026	**Last Revision Date:**	04/03/2026
Actors:	ผู้ทำการ Monitor		
Description:	Use Case นี้อธิบายกระบวนการการ Monitor เพื่อดูข้อมูลของแต่ละวันว่าเกิดปัญหาใดเพื่อทำการแก้และให้ระบบสามารถทำงานได้ตลอดไม่มีปัญหาโดยจะแสดงข้อมูล Top, Disk และ Tablespace หรือข้อมูลอื่นๆ ขึ้นอยู่กับเครื่องนั้นๆ		
Trigger:	ผู้ใช้งานเปิดหน้าเว็บไซต์ของระบบ Automate Daily Monitoring		
Preconditions: (Input)	1. ระบบจะทำการดึงข้อมูลล่าสุดของเครื่องทุกเครื่อง		
Postconditions: (Output)	1. ระบบจะแสดงผลข้อมูลล่าสุดของทุกเครื่อง		
Normal Flow: (หน้าจอ Monitor)	1. ระบบ แสดงช่องใส่ข้อมูล ดังนี้ 1.1. หน้า Home แสดงข้อมูลทุกเครื่องตามลำดับ 1.2. หน้าข้อมูลเครื่องแต่ละเครื่อง 2. ตาราง Top 3. ตาราง Disk 4. ตาราง Tablespace 5. ตารางอื่นๆขึ้นอยู่กับเครื่องนั้นๆ		

ตารางที่ 3-1 UC 1 การ Monitor (ต่อ)

[Alternative Flow 1 – ไม่สามารถดึงข้อมูลจากเครื่องเป้าหมายได้**]**	1a. ในขั้นตอนที่ 1 ของ Normal Flow หากระบบไม่สามารถดึงข้อมูลจากเครื่องเซิร์ฟเวอร์บางเครื่องได้ (เช่น ไม่มีข้อมูล Tablespace) 1.ระบบแสดงสถานะตารางเครื่องนั้นเป็น " No data available "
[**Alternative Flow** 2 – พบทรัพยากรระบบมีการใช้งานเกินเกณฑ์ (**Threshold)]**	2a. ในขั้นตอนที่ 2, 3 หรือ 4 ของ Normal Flow หากข้อมูลในตาราง Top, Disk หรือ Tablespace มีค่าสูงเกินเกณฑ์ที่กำหนดไว้ (เช่น Disk Space > 90%) 1.ระบบแสดงผลข้อมูลในแถวนั้นด้วยการไฮไลต์สี (เช่น สีแดง หรือ สีเหลือง) เพื่อให้สังเกตได้ง่าย
Exceptions**:**	1. หากระบบฐานข้อมูลหลักของตัว Automate Daily Monitoring เองล่ม ทำให้ไม่สามารถดึงข้อมูลใดๆ มาแสดงผลได้เลย ระบบจะแสดงหน้าจอ Error และหยุดการทำงาน
Includes**:**	ตรวจสอบข้อมูลเพื่อนำไปแสดงผลข้อมูลในแถวนั้นด้วยการไฮไลต์สี
Frequency of Use**:**	ใช้งานทุกวัน (Daily) หรือทำงานตลอดเวลาในลักษณะขึ้นอยู่กับผู้ทำการ Monitor
Special Requirements:	UI Responsiveness: การแสดงผลข้อมูลของเครื่องเซิร์ฟเวอร์หลายๆ เครื่องพร้อมกันในหน้า Home จะต้องไม่ทำให้หน้าเว็บค้างหรือกระตุกขณะที่ระบบกำลังดึงข้อมูลใหม่มาอัปเดตในแต่ละส่วนประกอบของหน้าจอ
Notes and Issues:	ความเร็วในการดึงและแสดงผลข้อมูลขึ้นอยู่กับความเสถียรของ Network ระหว่างตัวระบบ Monitoring และเครื่อง Back-end

2.การ Refresh ข้อมูล (UC2)

เป็นฟังก์ชันในการ Refresh ข้อมูลให้ล่าสุดเสมอโดยมีแบบเลือกเฉพาะเครื่องหรือทุกเครื่องดังตารางที่ 3-2

ตารางที่ 3-2 UC 2 การ Refresh ข้อมูล

Use Case ID:	UC2		
Use Case Name:	การ Refresh ข้อมูล		
Created By:	ธาม แก้วเกษ	Last Updated By:	ธาม แก้วเกษ
Date Created:	04/03/2026	**Last Revision Date:**	04/03/2026
Actors:	ผู้ทำการ Monitor		
Description:	Use Case นี้อธิบายกระบวนการที่ผู้ใช้งานสั่งอัปเดตข้อมูลสถานะของเครื่องเซิร์ฟเวอร์ให้เป็นข้อมูลปัจจุบันทันทีโดยสามารถเลือกได้ว่าจะอัปเดตทุกเครื่องพร้อมกันในหน้า Home หรือเจาะจงอัปเดตเฉพาะเครื่องที่กำลังดูข้อมูลอยู่		
Trigger:	ผู้ใช้งานคลิกปุ่ม "Refresh All" (อัปเดตทั้งหมด) หรือปุ่ม "Refresh" (อัปเดตเฉพาะเครื่อง) บนหน้าจอแสดงผล		
Preconditions: (Input)	1.ผู้ใช้งานอยู่ในหน้า Home หรือหน้าข้อมูลเฉพาะเครื่อง 2.ระบบสามารถเชื่อมต่อและส่งคำสั่งไปยังเครื่องเป้าหมายได้		
Postconditions: (Output)	1.ข้อมูลในตารางสถานะต่าง ๆ (Top, Disk, Tablespace ฯลฯ) บนหน้าจอ ถูกอัปเดตเป็นข้อมูลชุดใหม่ล่าสุด ณ วินาทีนั้น		

ตารางที่ 3-2 UC 2 การ Refresh ข้อมูล (ต่อ)

Normal Flow:	1. ผู้ใช้งานเลือกทำรายการ Refresh ข้อมูล โดยมี 2 ทางเลือก 1.1. (Refresh All) ผู้ใช้งานคลิกปุ่ม "Refresh All" ที่หน้า Home 1.2. (Refresh Specific) ผู้ใช้งานคลิกปุ่ม "Refresh" ที่หน้าของเครื่องนั้นๆ 2. ระบบแสดงสถานะกำลังโหลด (เช่น ไอคอนหมุน Loading/Spinner) บนหน้าจอเพื่อแจ้งให้ผู้ใช้ทราบว่าระบบกำลังทำงาน 3. ระบบ (Frontend) ส่งคำสั่งไปยัง Backend เพื่อดึงข้อมูลสถานะล่าสุดจากเครื่องเป้าหมายตามที่ผู้ใช้เลือก (ทุกเครื่อง หรือ 1 เครื่อง) 4. Backend ดึงข้อมูลสำเร็จและส่งข้อมูลกลับมาที่ระบบ 5. ระบบนำข้อมูลใหม่มาแทนที่ข้อมูลเดิมในตารางต่างๆ บนหน้าจอทันที
[Alternative Flow 1 – การ Refresh All มีบางเครื่องไม่ตอบสนอง**]**	1a. ในขั้นตอนที่ 4 ของ Normal Flow (กรณี Refresh All) หากมีเครื่องเซิร์ฟเวอร์บางเครื่องไม่สามารถตอบสนองได้ทันเวลา (Timeout) หรือเชื่อมต่อไม่ได้ 1. ระบบอัปเดตข้อมูลของเครื่องที่ดึงข้อมูลสำเร็จตามปกติ 2. สำหรับเครื่องที่ล้มเหลว ระบบจะแสดงสถานะเป็น "Offline" หรือไอคอนแจ้งเตือน Error บริเวณข้อมูลของเครื่องนั้น 3. ระบบแจ้งเตือนว่า "ดึงข้อมูลสำเร็จบางส่วน ไม่สามารถเชื่อมต่อเครื่อง (ชื่อเครื่อง) ได้"
Exceptions**:**	1. หากการเชื่อมต่อระหว่างหน้าเว็บกับ Backend หลักล่ม จะยกเลิกการ Refresh ทันที
Includes**:**	ตรวจสอบข้อมูลเพื่อนำไปแสดงผลข้อมูลในแถวนั้นด้วยการไฮไลต์สี
Frequency of Use**:**	บ่อยครั้ง (เมื่อผู้ใช้งานต้องการเห็นข้อมูลที่อัปเดตทันที นอกเหนือจากรอบ Auto-Refresh)

ตารางที่ 3-2 UC 2 การ Refresh ข้อมูล (ต่อ)

Special Requirements:	Loading Feedback: ระหว่างที่กดปุ่ม Refresh ระบบต้องมี UI แสดงให้ผู้ใช้เห็นชัดเจนว่ากำลังทำงานอยู่ เพื่อป้องกันการกดปุ่มซ้ำหลายครั้ง (Prevent Double Click)
Notes and Issues:	การใช้ "Refresh All" หากมีเครื่องเป้าหมายจำนวนมาก (เช่น 50-100 เครื่อง) อาจทำให้ใช้เวลารอข้อมูลนานกว่าปกติ

3. การย้อนดูข้อมูล (UC3)

เป็นฟังก์ชันสำหรับการเรียกดูข้อมูลสถิติหรือสถานะของเซิร์ฟเวอร์ย้อนหลังตามวันที่ได้จัดเก็บไว้ เพื่อนำมาวิเคราะห์และตรวจสอบการทำงานในอดีตดังตารางที่ 3-3

ตารางที่ 3-3 UC 3 การย้อนดูข้อมูล

Use Case ID:	UC3		
Use Case Name:	การย้อนดูข้อมูล		
Created By:	ธาม แก้วเกษ	Last Updated By:	ธาม แก้วเกษ
Date Created:	04/03/2026	**Last Revision Date:**	04/03/2026
Actors:	ผู้ทำการ Monitor		
Description:	Use Case นี้อธิบายกระบวนการที่ผู้ใช้งานต้องการเรียกดูข้อมูลสถานะเซิร์ฟเวอร์ย้อนหลังตามวันที่ได้จัดเก็บประวัติไว้ในฐานข้อมูลหรือ File System เพื่อนำมาวิเคราะห์ โดยสามารถย้อนดูข้อมูลของ Top, Disk และ Tablespace หรือข้อมูลอื่นๆ ขึ้นอยู่กับเครื่องนั้นๆ		
Trigger:	ผู้ใช้งานเลือกวันที่บนเครื่องมือค้นหาย้อนหลัง (Date Picker) หรือระบุช่วงเวลาที่ต้องการ		
Preconditions: (Input)	มีไฟล์ประวัติการ Monitor (เช่น JSON หรือ Text) ของวันที่ต้องการจัดเก็บอยู่ในระบบ		
Postconditions: (Output)	ระบบแสดงข้อมูลสถานะการใช้งานระบบตามวันที่ที่ผู้ใช้ระบุบนหน้าจอ		

ตารางที่ 3-3 UC 3 การย้อนดูข้อมูล (ต่อ)

Normal Flow:	1. ผู้ใช้งานคลิกเลือกปฏิทินเพื่อระบุวันที่ต้องการย้อนดูข้อมูล 2. ระบบส่งคำขอ (Query) ไปยังระบบหลังบ้านเพื่อดึงข้อมูลของวันนั้น 3. ระบบได้รับชุดข้อมูลประวัติ และนำมาปรับปรุงตารางแสดงผลบนหน้าจอ
[Alternative Flow 1 – การ Refresh All มีบางเครื่องไม่ตอบสนอง]	1a. หากผู้ใช้งานเลือกวันที่ไม่มีการบันทึกข้อมูลไว้ 1. ระบบจะเคลียร์ตารางแสดงผลและแสดงข้อความแจ้งเตือน "ไม่พบข้อมูลของวันที่ระบุ"
Exceptions:	1. หากโครงสร้างไฟล์ประวัติเสียหาย หรืออ่านค่าไม่ได้ ระบบจะแสดงข้อความ Error "ไม่สามารถอ่านไฟล์ข้อมูลได้"
Includes:	-
Frequency of Use:	ปานกลาง (มักใช้งานเมื่อต้องการทำรายงานย้อนหลัง หรือตรวจสอบสาเหตุของปัญหาระบบล่ม)
Special Requirements:	-
Notes and Issues:	ข้อมูลย้อนหลังจะอิงจากการจัดเก็บของระบบ Backend ซึ่งหากมีการลบ Log ทิ้งตามรอบ Retention Policy อาจจะไม่สามารถย้อนดูได้

4. การเปรียบเทียบข้อมูล (UC4)

เป็นฟังก์ชันที่ใช้ในการนำข้อมูลสถิติของเซิร์ฟเวอร์ใน 2 ช่วงเวลามาแสดงผลเปรียบเทียบกัน การใช้งานทรัพยากรดังตารางที่ 3-4

ตารางที่ 3-4 UC 4 การเปรียบเทียบข้อมูล

Use Case ID:	UC4		
Use Case Name:	การเปรียบเทียบข้อมูล		
Created By:	ธาม แก้วเกษ	Last Updated By:	ธาม แก้วเกษ
Date Created:	04/03/2026	**Last Revision Date:**	04/03/2026
Actors:	ผู้ทำการ Monitor		
Description:	Use Case นี้อธิบายกระบวนการนำข้อมูลสถิติหรือสถานะของเซิร์ฟเวอร์ 2 ช่วงเวลา มาแสดงผลเปรียบเทียบกัน เพื่อวิเคราะห์แนวโน้มการใช้ทรัพยากรที่เพิ่มขึ้นหรือลดลง โดยเปรียบเทียบข้อมูลได้เฉพาะ Disk และ Tablespace		
Trigger:	ผู้ใช้งานเลือกฟังก์ชันเปรียบเทียบข้อมูล และกำหนดตัวแปร (เช่น วันที่ ก. เทียบกับ วันที่ ข.)		
Preconditions: (Input)	มีข้อมูลของช่วงเวลาหรือเซิร์ฟเวอร์ที่ต้องการเปรียบเทียบอยู่ในระบบอย่างสมบูรณ์		
Postconditions: (Output)	ระบบแสดงตารางผลลัพธ์ที่แสดงการเปรียบเทียบข้อมูลให้เห็นความแตกต่างชัดเจน		
Normal Flow:	1. ผู้ใช้งานเข้าสู่เมนู/แท็บ การเปรียบเทียบข้อมูล 2. ผู้ใช้งานเลือกวันที่เป้าหมายที่ 1 และ 2 3. ระบบดึงข้อมูลทั้ง 2 ส่วนมาทำการคำนวณหาความต่าง (Diff)		

ตารางที่ 3-4 UC 4 การเปรียบเทียบข้อมูล (ต่อ)

Exceptions:	1. หากผู้ใช้เลือกช่วงเวลาที่มีแค่ข้อมูลเดียวระบบจะไม่สามารถเปรียบเทียบได้
Includes:	-
Frequency of Use:	ต่ำถึงปานกลาง
Special Requirements:	สามารถดูการแสดงผลแบบกราฟจะต้องดูเข้าใจง่าย มีคำอธิบายแกน (Legend) และสีที่แยกความแตกต่างได้ชัดเจน
Notes and Issues:	-

6.การ Export ข้อมูล (UC5)

เป็นฟังก์ชันสำหรับการแปลงข้อมูลสถานะที่แสดงผลอยู่บนหน้าเว็บให้ออกมาเป็นไฟล์เอกสาร (PDF) เพื่อใช้สรุปผลและจัดส่งรายงานให้กับทีมงานหรือลูกค้าดังตารางที่ 3-5

ตารางที่ 3-5 UC 5 การ Export ข้อมูล

Use Case ID:	UC5		
Use Case Name:	การ Export ข้อมูล		
Created By:	ธาม แก้วเกษ	Last Updated By:	ธาม แก้วเกษ
Date Created:	04/03/2026	**Last Revision Date:**	04/03/2026
Actors:	ผู้ทำการ Monitor		
Description:	Use Case นี้อธิบายขั้นตอนการแปลงข้อมูลสถานะที่แสดงผลอยู่บนหน้าเว็บแอปพลิเคชัน ให้ออกมาเป็นไฟล์เอกสาร (PDF) เพื่อใช้สรุปผลและใช้แนบส่งอีเมลให้ลูกค้าหรือทีมงาน		

ตารางที่ 3-5 UC 5 การ Export ข้อมูล (ต่อ)

Trigger:	ผู้ใช้งานคลิกปุ่ม "Export to PDF" บนหน้าจอ
Preconditions: (Input)	หน้าเว็บโหลดข้อมูลทั้งหมดที่ต้องการ Export เสร็จสมบูรณ์แล้ว ไม่มีการประมวลผลค้างอยู่
Postconditions: (Output)	ผู้ใช้งานได้รับไฟล์เอกสารรายงานบันทึกลงในเครื่องคอมพิวเตอร์
Normal Flow:	1. ผู้ใช้งานตรวจสอบความเรียบร้อยของข้อมูลบนหน้าจอ 2. ผู้ใช้งานคลิกปุ่ม "Export to PDF" 3. ระบบแสดงสถานะ "กำลังประมวลผลสร้างไฟล์รายงาน..." 4. ระบบจัดเรียง Layout หน้าจอใหม่ให้เหมาะสมกับสัดส่วนหน้ากระดาษ A4 5. ระบบแปลงหน้าจอเป็นไฟล์นามสกุล .pdf และกระตุ้นให้ Browser ดาวน์โหลดไฟล์
Exceptions:	1. หากมีปริมาณข้อมูลตารางมากเกินขีดจำกัดหน่วยความจำของบราวเซอร์ อาจทำให้สร้างไฟล์ PDF ล้มเหลว
Includes:	-
Frequency of Use:	เป็นประจำทุกวัน (ภายหลังจากการทำ Monitoring และตรวจเช็คเสร็จสิ้น)
Special Requirements:	เอกสารที่ถูก Export ออกมาจะต้องไม่ผิดเพี้ยนจากที่แสดงบนหน้าจอ ตัวหนังสืออ่านง่าย และรองรับฟอนต์ภาษาไทยได้สมบูรณ์
Notes and Issues:	ควรมีการแทรกวันที่และเวลาที่ทำการดึงข้อมูลไว้ที่หัวกระดาษเอกสารอัตโนมัติ

6. การ Download ข้อมูล (UC6)

เป็นฟังก์ชันที่ให้ผู้ใช้งานสามารถดาวน์โหลดไฟล์ข้อมูลดิบ (Raw Data) หรือไฟล์ JSON ที่ระบบจัดเก็บไว้บนเซิร์ฟเวอร์ เพื่อนำไปวิเคราะห์ในเชิงลึกหรือใช้งานกับสคริปต์ภายนอกดังตารางที่ 3-6

ตารางที่ 3-6 UC 6 การ Download ข้อมูล

Use Case ID:	UC6		
Use Case Name:	การ Download ข้อมูล		
Created By:	ธาม แก้วเกษ	Last Updated By:	ธาม แก้วเกษ
Date Created:	04/03/2026	**Last Revision Date:**	04/03/2026
Actors:	ผู้ทำการ Monitor		
Description:	Use Case นี้อธิบายการทำงานที่ผู้ใช้งานต้องการดาวน์โหลดไฟล์ข้อมูลดิบ (Raw Data) หรือไฟล์ JSON ที่ระบบ Backend ได้รวบรวมและจัดเก็บไว้บนเซิร์ฟเวอร์ เพื่อนำไปวิเคราะห์เชิงลึก หรือทำ Script ภายนอก โดยสามารถดาวน์โหลดได้ทุกข้อมูลที่แสดงที่แต่เครื่อง		
Trigger:	ผู้ใช้งานคลิกปุ่ม "Download" ที่รายการไฟล์ประวัติ หรือบนแท็บ File Log		
Preconditions: (Input)	ต้องมีไฟล์ Log (.txt) หรือไฟล์ .Json จัดเก็บอยู่ในเครื่องเซิร์ฟเวอร์ที่รันระบบ Backend อยู่		
Postconditions: (Output)	เบราว์เซอร์ทำการดาวน์โหลดไฟล์ดิบ (Text หรือ JSON) ลงสู่เครื่องคอมพิวเตอร์ของผู้ใช้		
Normal Flow:	1. ผู้ใช้งานเข้าสู่ส่วนของระบบที่แสดงรายการไฟล์ Log 2. ผู้ใช้งานคลิกปุ่ม Download ที่ท้ายชื่อไฟล์ที่ต้องการ 3. Frontend ส่ง Request แจ้งชื่อและพาธของไฟล์ไปยัง Backend 4. Backend ทำการค้นหาไฟล์ (Find File) และอ่านข้อมูลไฟล์นั้น (Read File) 5. Backend จัดเตรียม Response กลับมาในรูปแบบเอกสารแนบ (Attachment) 6. ไฟล์ถูกดาวน์โหลดลงเครื่องผู้ใช้งาน		

ตารางที่ 3-6 UC 6 การ Download ข้อมูล (ต่อ)

Exceptions:	1. หากไฟล์มีขนาดใหญ่มากจนเกินกำหนด (File Size Limit) อาจจะทำให้การดาวน์โหลดถูกตัดการเชื่อมต่อระหว่างกลางได้
Includes:	-
Frequency of Use:	นานๆ ครั้ง (เฉพาะในกรณีที่มีปัญหาเชิงลึก และทีมงานต้องการดูไฟล์ผลลัพธ์ Command ของจริง)
Special Requirements:	-
Notes and Issues:	ใช้การตั้งค่า Content-Disposition: attachment ในส่วนของ Header ตอนตอบกลับ (Respond to Webhook) เพื่อบังคับดาวน์โหลด

3.3 ขั้นตอนการออกแบบ Front-end ของ Automate Daily Monitoring

ในส่วนนี้จะนำเสนอการออกแบบหน้าจอผู้ใช้งาน (User Interface) ของระบบโดยเน้นที่การสร้างประสบการณ์ใช้งานที่สะดวก และเป็นมิตรกับผู้ใช้ (User Friendly) โดยองค์ประกอบสำคัญภายในหน้าจอจะประกอบด้วยแถบเมนูหลักที่ช่วยให้ผู้ใช้งานสามารถเข้าถึงฟังก์ชันต่าง ๆ ของระบบได้อย่างง่ายดาย การจัดวางปุ่ม และข้อความที่ชัดเจนเพื่อลดความซับซ้อนในการใช้งานอีกทั้งยังมีการใช้สีและตัวอักษรที่เหมาะสมเพื่อให้การอ่าน และการโต้ตอบสะดวกมากยิ่งขึ้น โดยมีรายละเอียดดังต่อไปนี้

1. หน้าจอหลัก (Home) โดยแสดงดังภาพที่ 3-45



ภาพที่ 3-45 ภาพหน้าจอหลัก (Home)

จากภาพที่ 3-51 เป็นการออกแบบหน้าจอการเข้าหลัก ซึ่งเป็นหน้าจอแรกที่ผู้ใช้งานจะต้องเจอ โดยหน้าจอที่ออกแบบมีองค์ประกอบต่าง ๆ ดังนี้

1.ปุ่ม Refresh All

2.ปุ่ม Export PDF

3.ข้อมูลที่แสดงในแต่ละ Server โดย หน้า Home จะเป็นหน้าที่รวบรวมข้อมูลทุก Server ไว้ในหน้าเดียวกันเพื่อง่ายต่อการดูข้อมูลโดยมีข้อมูลดังต่อไปนี้

3.1 ข้อมูลตารางจะแสดงตารางของ CPU และ Ram มีองค์ประกอบต่าง ๆ ดังนี้

1.ปุ่ม Refresh

2.ปุ่ม Range

3.ปุ่ม Day Range

4.ปุ่ม Month

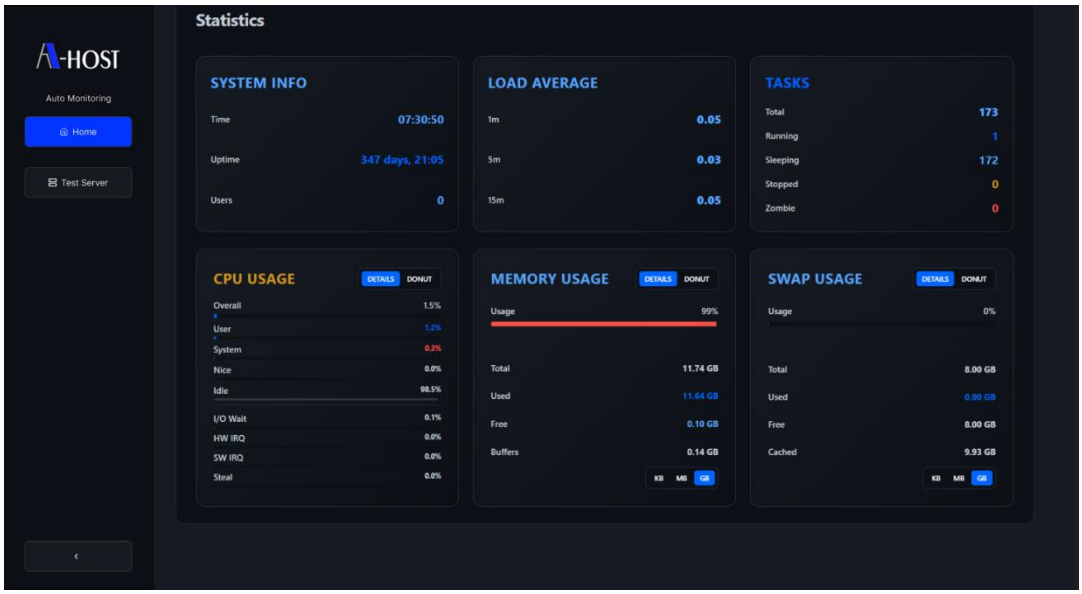
5.ปุ่ม Year

6.กราฟแสดง CPU TREND

7.กราฟแสดง RAM TREND

8.ปุ่ม Download กราฟ

3.2 ข้อมูล Statistics แสดงข้อมูลการใช้งาน CPU Ram และ อื่นๆดังภาพที่ 3-46



ภาพที่ 3-46 ภาพข้อมูล Statistics

โดยข้อมูล Statistics มีองค์ประกอบต่าง ๆ ดังนี้

1.ข้อมูล SYSTEM INFO

2.ค่า LOAD AVERAGE

3.ค่า TASKS

4.ค่า CPU USAGE

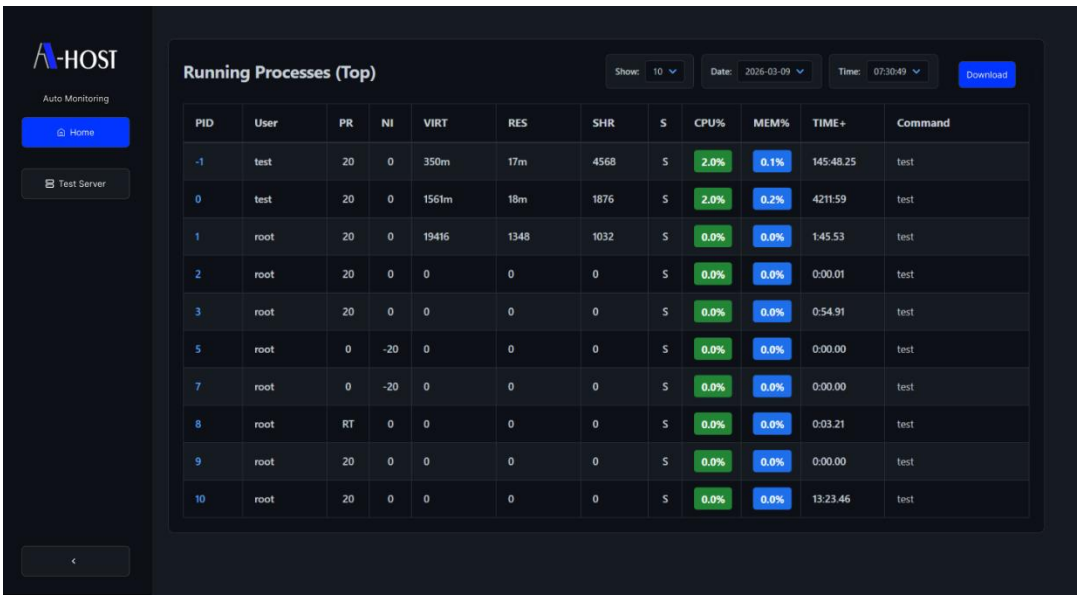
5.ค่า MEMORY USAGE

6.ค่า SWAP USAGE

7.ปุ่มแสดงข้อมูล Details และ Donut

8.ปุ่มเปลี่ยนการแสดงข้อมูล KB, MB และ GB

3.3 ข้อมูล Running Processes (Top) แสดงข้อมูลการใช้งาน CPU โดยจะแสดงเป็นตาราง ของ Processes ทั้งหมด และมีปุ่ม Download ดังภาพที่ 3-47

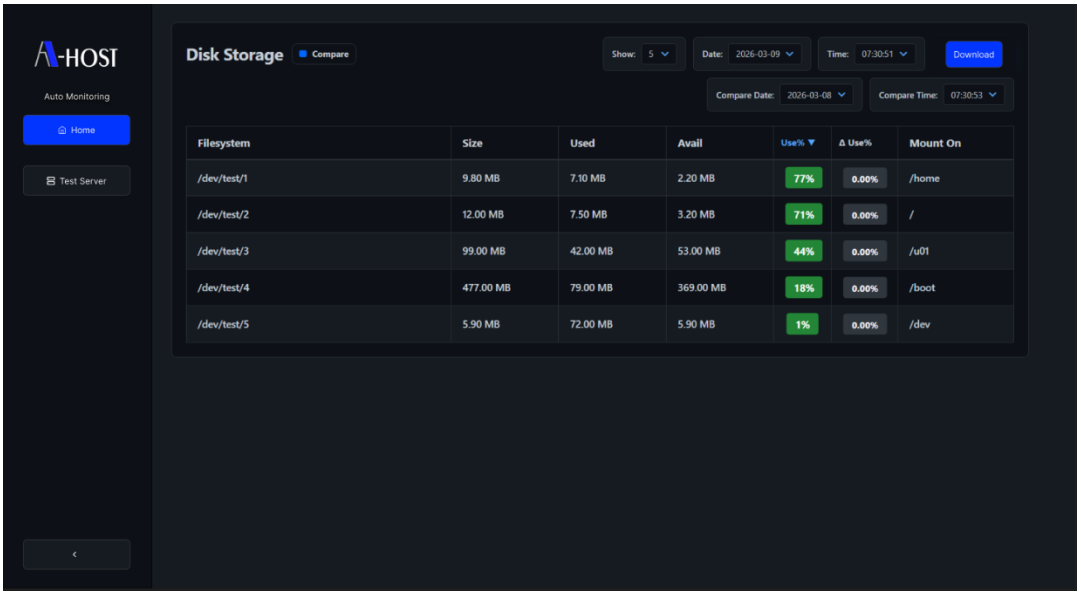


ภาพที่ 3-47 ภาพข้อมูล Running Processes (Top)

โดยข้อมูล Running Processes (Top) มีองค์ประกอบต่าง ๆ ดังนี้

1.ปุ่มแสดงจำนวนข้อมูล
2.ปุ่มเลือกวันที่
3.ปุ่มเลือกเวลา
4.ปุ่ม Download
5.ข้อมูล PID
6.ข้อมูล USER
7.ค่า PR
8.ค่า NI
9.ค่า VIRT
10.ค่า RES
11.ค่า SHR
12.ข้อมูล S
13.ค่า CPU%
14.ค่า MEM%
15.ค่า TIME+
16.ข้อมูล Command

3.4 ข้อมูล Disk Storage แสดงข้อมูลการใช้งาน Disk โดยจะแสดงเป็นตารางของ Disk ทั้งหมด มีปุ่ม Download และ Compare สำหรับใช้เปรียบเทียบข้อมูลก่อนหน้า ดังภาพที่ 3-48



ภาพที่ 3-48 ภาพข้อมูล Disk Storage

โดยข้อมูล Disk Storage มีองค์ประกอบต่าง ๆ ดังนี้

1.ปุ่มแสดงจำนวนข้อมูล
2.ปุ่มเลือกวันที่
3.ปุ่มเลือกเวลา
4.ปุ่ม Download
5.ปุ่ม Compare
6.ปุ่มเลือกวันที่ Compare
7.ปุ่มเลือกเวลา Compare
8.ข้อมูล Filesystem
9.ค่า Size
10.ค่า Used
11.ค่า Avail
12.ค่า Use%
13.ค่า Use% Compare
14.ข้อมูล Mount On

3.5 ข้อมูลเฉพาะ Server เนื่องจากมีข้อมูลบาง Server ที่หน้าตาไม่เหมือน Server อื่นจะจัดแสดงในรูปแบบของ ตารางและมีองค์ประกอบต่างๆ ดังนี้

1.ปุ่มแสดงจำนวนข้อมูล

2.ปุ่มเลือกวันที่

3.ปุ่มเลือกเวลา

4.ปุ่ม Download

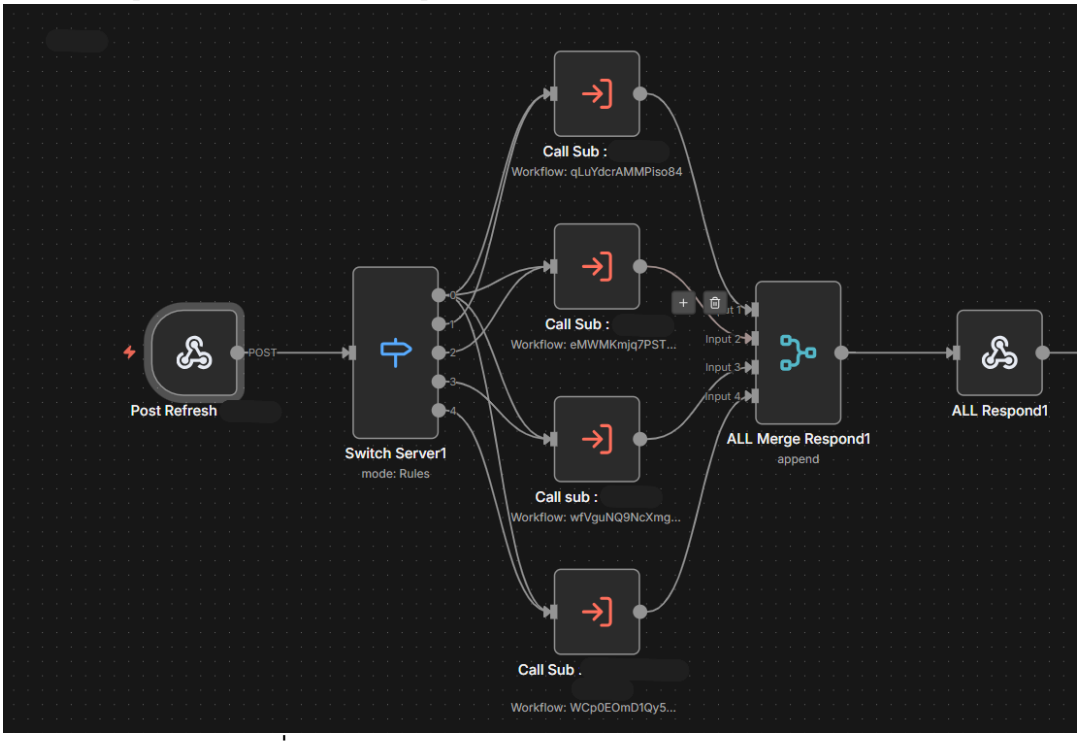
5.ข้อมูลเฉพาะของ Server นั้นๆ

บทที่ 4
ผลการดำเนินงานสหกิจศึกษา

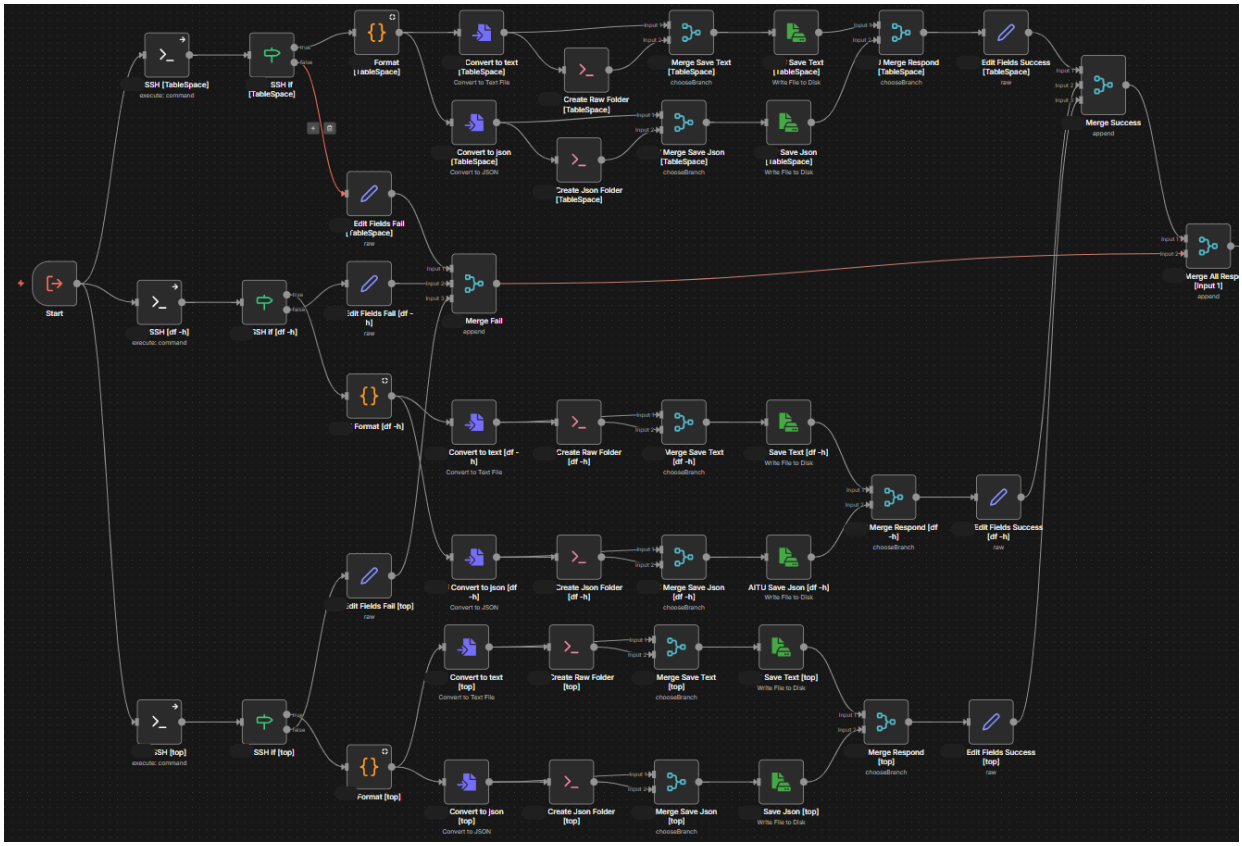
จากการปฏิบัติงานสหกิจศึกษา ณ บริษัท เอ-โฮสต์ จำกัด (A-HOST Co., Ltd.) ในตำแหน่ง Cloud Engineer ตั้งแต่วันที่ 10 พฤศจิกายน พ.ศ. 2568 ถึงวันที่ 13 มีนาคม พ.ศ. 2569 ระยะเวลา 17 สัปดาห์ ผู้จัดทำได้รับมอบหมายให้พัฒนาโครงงาน "การออกแบบและพัฒนาเว็บแอปพลิเคชัน Automate Daily Monitoring" ซึ่งผลการดำเนินงานได้เสร็จสิ้นสมบูรณ์ตามวัตถุประสงค์ที่วางไว้ โดยสามารถสรุปผลการดำเนินงานได้ดังนี้

4.1 ผลการพัฒนาระบบส่วน Back-end

ระบบในส่วนของการจัดการข้อมูลเบื้องหลังสามารถทำงานได้โดยอัตโนมัติตามกระบวนการที่ออกแบบไว้ โดยระบบสามารถทำการดึงข้อมูลสถานะการใช้ทรัพยากรต่างๆ เช่น CPU, RAM และ Disk Storage ผ่านการเชื่อมต่อด้วยคำสั่ง SSH เข้าไปยังเครื่องเซิร์ฟเวอร์เป้าหมายของลูกค้าโดยอัตโนมัติในช่วงเช้าของทุกวัน จากนั้นระบบจะทำการคัดกรองข้อมูล แปลงข้อมูล (Parse Data) ให้อยู่ในรูปแบบ JSON และจัดเก็บลงฐานข้อมูลได้อย่างถูกต้องแม่นยำ ขั้นตอนนี้ช่วยลดภาระงานที่ต้องใช้แรงงานคนเข้าไปรันคำสั่งและคัดลอกข้อมูลทีละเครื่องได้อย่างสมบูรณ์โดยมีผลลัพธ์ ดังภาพที่ 4-1 และ 4-2



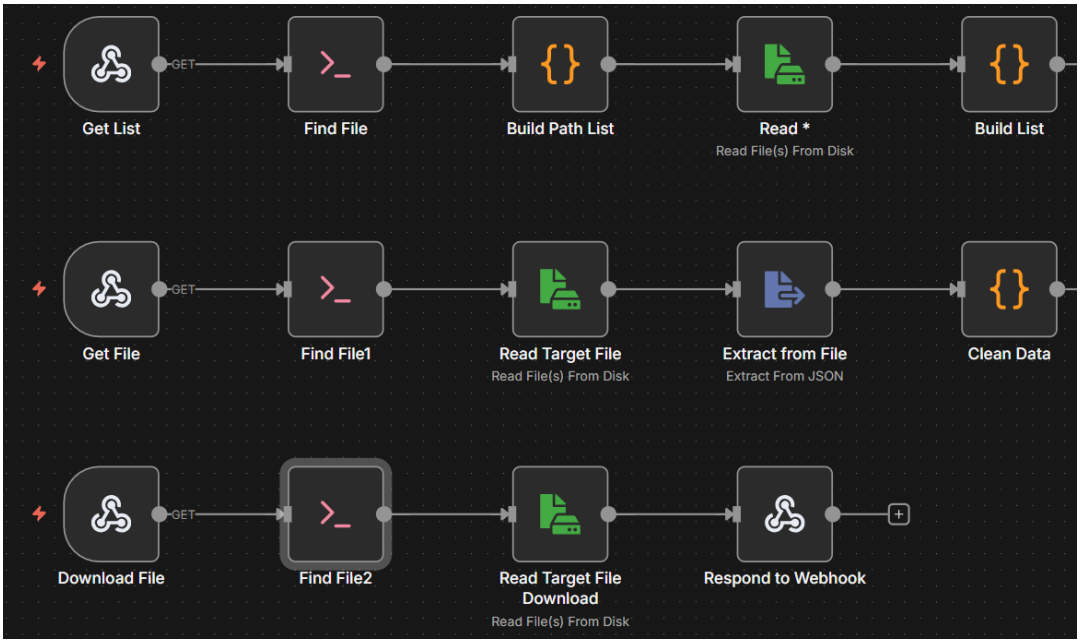
ภาพที่ 4-1 ตัวอย่าง Flowchart Backend Refresh Data



ภาพที่ 4-2 ภายใน Sub Workflow SSH Node

นอกเหนือจากการดึงข้อมูลสถานะทรัพยากรของเครื่องเซิร์ฟเวอร์แบบอัตโนมัติแล้ว ระบบยังได้มีการพัฒนา API สำหรับการเรียกดูข้อมูลไฟล์ (File Management) โดยเฉพาะไฟล์บันทึกการทำงาน (Log Files) จากเครื่องเป้าหมาย

ระบบมีการสร้าง Workflow ผ่าน n8n เพื่อทำหน้าที่เป็น API Endpoint สำหรับรับคำสั่งจากหน้าเว็บแอปพลิเคชัน เมื่อผู้ใช้งานต้องการตรวจสอบไฟล์ ระบบจะทำการดึงข้อมูลไฟล์ Log จากเครื่องเซิร์ฟเวอร์เป้าหมาย และส่งกลับมาแสดงผลให้ผู้ใช้งานสามารถอ่านและตรวจสอบความผิดปกติได้โดยตรงผ่านหน้าจอ Web Application ทันที ฟังก์ชันนี้ช่วยเพิ่มความสะดวกอย่างมาก เนื่องจากผู้ดูแลระบบไม่จำเป็นต้องใช้คำสั่ง SSH เข้าไปค้นหาและเปิดอ่านไฟล์ Log ในแต่ละเครื่องด้วยตนเอง ทำให้การวิเคราะห์และแก้ไขปัญหา (Troubleshooting) เป็นไปได้อย่างรวดเร็วและมีประสิทธิภาพมากขึ้นโดยมีผลลัพธ์ ดังภาพที่ 4-3



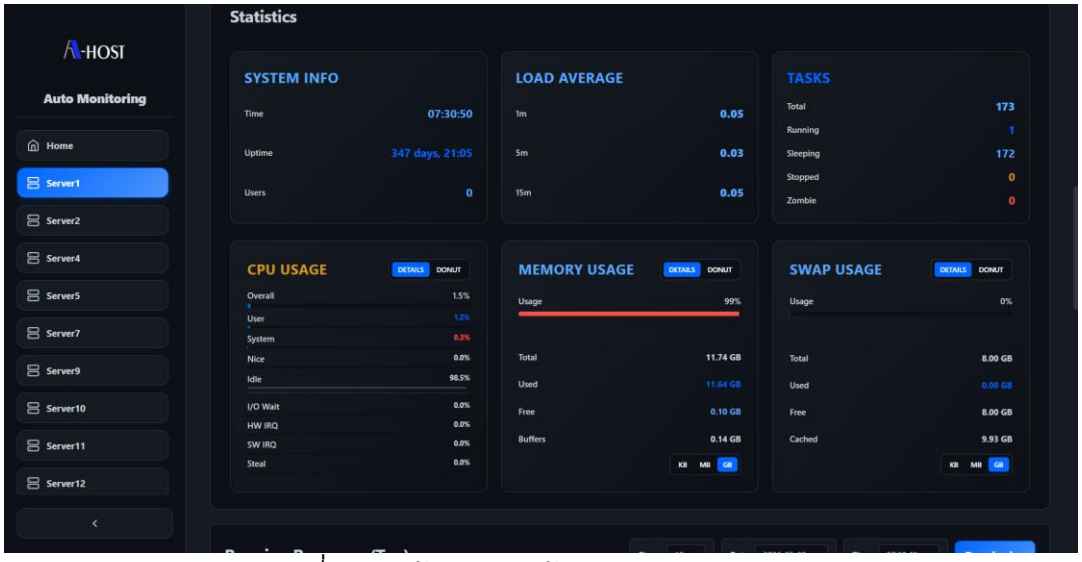
ภาพที่ 4-3 Flowchart Backend File Log Monitoring

4.2 ผลการพัฒนาระบบส่วน Front-end

ในการพัฒนาระบบส่วนติดต่อผู้ใช้งาน (User Interface) ได้มีการเลือกใช้ไลบรารี React ในการพัฒนาเป็น Web Application โดยมีจุดประสงค์หลักเพื่อนำข้อมูลที่ถูกประมวลผลจากส่วน Back-end มาแสดงผลให้ผู้ใช้งานสามารถทำความเข้าใจได้ง่ายและรวดเร็วที่สุด โดยสามารถแบ่งผลการพัฒนาออกเป็นส่วนสำคัญ ดังนี้

4.2.1 การแสดงผลกระดานข้อมูลภาพรวม

ระบบได้รับการออกแบบให้มีหน้าจอหลัก (Dashboard) ที่รวบรวมข้อมูลสถานะของเซิร์ฟเวอร์ทั้งหมดมาไว้ในหน้าเดียว โดยมีการนำระบบสัญลักษณ์สี (Health Indicators) มาใช้เพื่อบ่งบอกถึงสถานะการทำงานและความผิดปกติของระบบ หากเซิร์ฟเวอร์ใดมีการใช้ทรัพยากรเกินเกณฑ์มาตรฐานที่กำหนดไว้ ระบบจะแสดงแถบสีแจ้งเตือนอย่างชัดเจน ทำให้ผู้ดูแลระบบสามารถกวาดสายตาเพื่อประเมินสถานการณ์ภาพรวม (Morning Check) ได้ภายในเวลาไม่กี่นาทีโดยมีผลลัพธ์ ดังภาพที่ 4-4

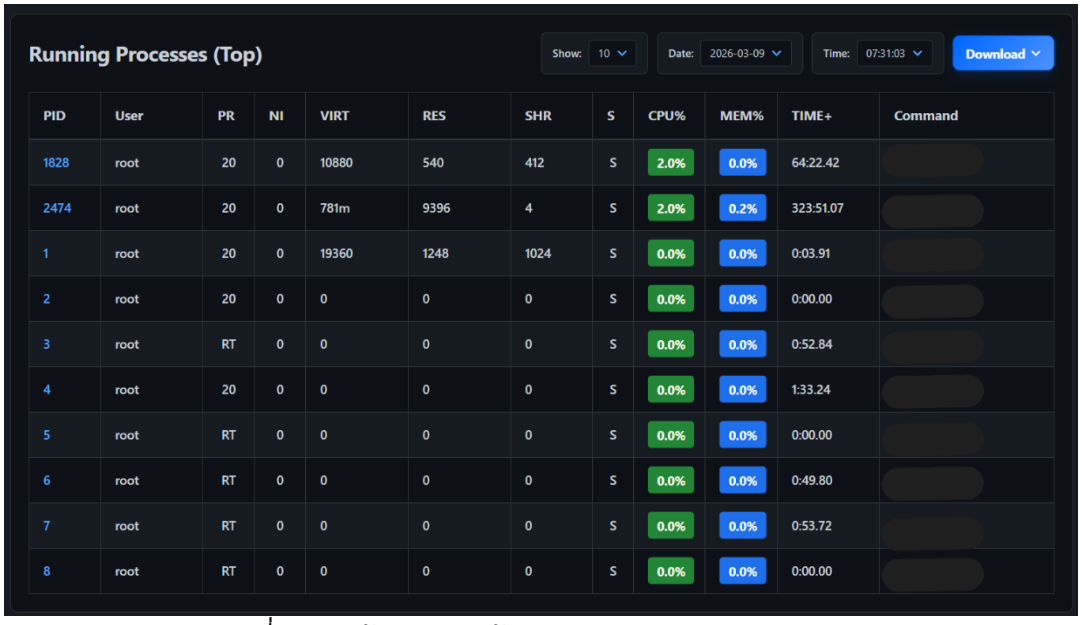


ภาพที่ 4-4 หน้าจอแสดงข้อมูลภาพรวม Statistics

4.2.2 การแสดงผลข้อมูลเชิงลึกและการเปรียบเทียบ

ผู้ใช้งานสามารถกดดูรายละเอียดเชิงลึกของแต่ละเซิร์ฟเวอร์ได้ โดยระบบจะแสดงผลข้อมูลแยกเป็นหมวดหมู่ที่สำคัญต่อการตรวจสอบ ได้แก่

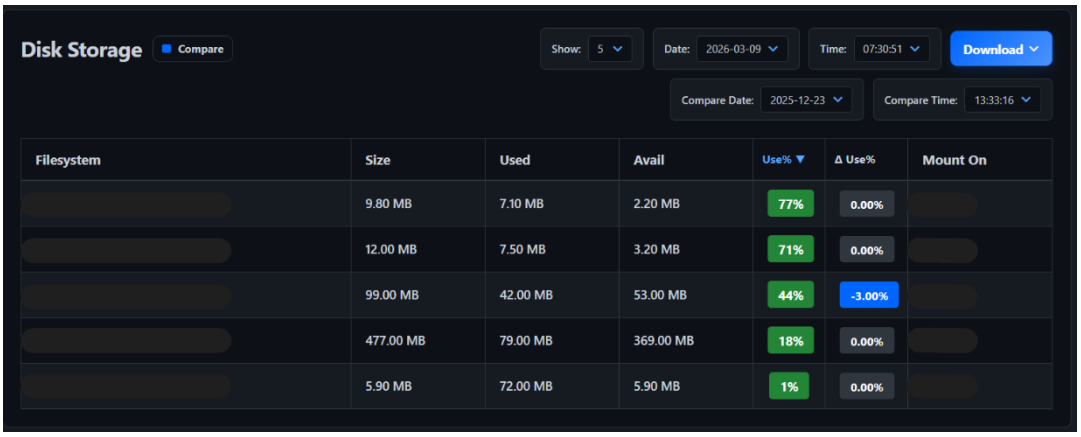
1.Running Processes (Top): แสดงรายการโปรเซสที่กำลังทำงานและกินทรัพยากรของเครื่อง ณ เวลานั้นๆโดยมีผลลัพธ์ ดังภาพที่ 4-5



ภาพที่ 4-5 หน้าจอแสดงข้อมูล Running Processes (Top)

2.ข้อมูลพื้นที่จัดเก็บ (Disk Storage): แสดงรายละเอียดของ Filesystem, ขนาดพื้นที่ทั้งหมด (Size), พื้นที่ที่ใช้งานไป (Used) และพื้นที่คงเหลือ (Avail) ในรูปแบบตารางที่อ่านค่าได้ง่าย

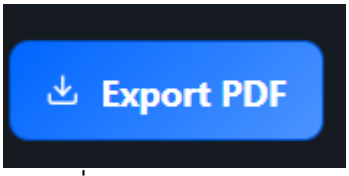
ฟังก์ชันเปรียบเทียบข้อมูล (Compare): ระบบได้เพิ่มความสามารถในการเลือกวันและเวลา เพื่อนำข้อมูลสถานะของเซิร์ฟเวอร์ในอดีตมาเปรียบเทียบกับข้อมูลปัจจุบัน (Compare Feature) ฟังก์ชันนี้มีประโยชน์อย่างมากในการวิเคราะห์แนวโน้มการเติบโตของข้อมูล หรือค้นหาสาเหตุของความผิดปกติที่เกิดขึ้นแบบก้าวกระโดด



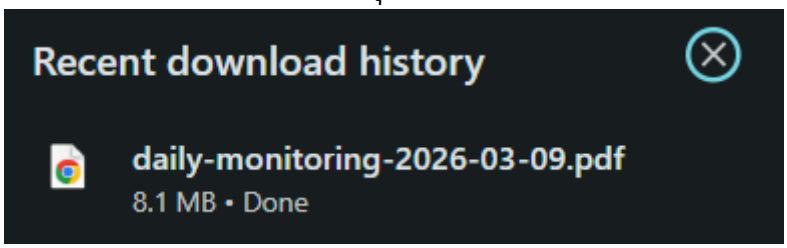
ภาพที่ 4-6 หน้าจอแสดงข้อมูล Disk Storage และทำการ Compare

3.การส่งออกรายงานอัตโนมัติ

เพื่อตอบสนองวัตถุประสงค์ในการลดขั้นตอนการทำงานซ้ำซ้อน ระบบ Front-end ได้เตรียมฟังก์ชันสำหรับดาวน์โหลดรายงาน (Export PDF) โดยผู้ใช้งานสามารถกดเพียงปุ่มเดียวเพื่อส่งออกข้อมูลสถานะที่กำลังแสดงผลอยู่ ให้อยู่ในรูปแบบไฟล์เอกสารที่พร้อมสำหรับการนำส่งให้ลูกค้าหรือหัวหน้างานได้ทันที ช่วยลดภาระการคัดลอกข้อมูลลงตารางแบบเดิมได้อย่างสมบูรณ์



ภาพที่ 4-7 ปุ่ม Export PDF

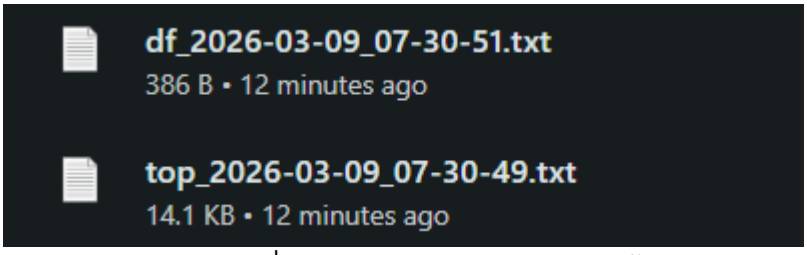


ภาพที่ 4-8 ผลลัพธ์การ Export PDF

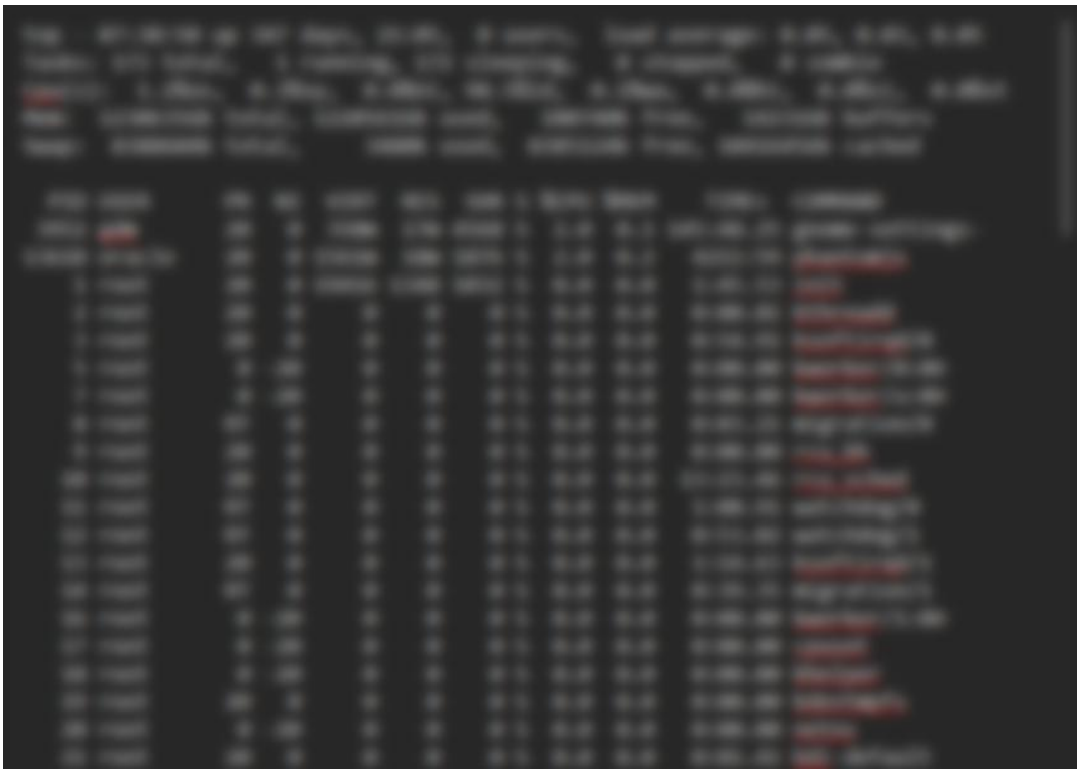


ภาพที่ 4-9 ผลลัพธ์ไฟล์ PDF

4.การดาวน์โหลดไฟล์บันทึกการทำงาน นอกเหนือจากการแสดงผลข้อมูลสถิติและสถานะของเซิร์ฟเวอร์แล้ว ระบบส่วน Front-end ยังรองรับการจัดการไฟล์บันทึกการทำงาน (Log Files) โดยมีฟังก์ชันให้ผู้ใช้งานสามารถดาวน์โหลดไฟล์ Log ที่ถูกดึงมาจากเซิร์ฟเวอร์เป้าหมายผ่าน API ได้โดยตรงจากหน้าเว็บแอปพลิเคชัน ฟังก์ชันนี้ถูกออกแบบมาเพื่ออำนวยความสะดวกในกรณีที่ผู้ดูแลระบบต้องการนำไฟล์ข้อมูลดิบ (Raw Data) ไปวิเคราะห์เชิงลึก (Deep-dive Analysis) ด้วยเครื่องมือเฉพาะทางอื่น ๆ หรือต้องการจัดเก็บไฟล์ไว้เป็นประวัติอ้างอิง (Archive) นอกระบบ รวมถึงสามารถนำไฟล์ที่ดาวน์โหลดไปส่งต่อให้ทีมงานวิศวกรที่เกี่ยวข้อง เพื่อนำไปใช้ประกอบการตรวจสอบและแก้ไขปัญหาของระบบได้อย่างรวดเร็ว



ภาพที่ 4-10 ผลลัพธ์การ Download ไฟล์



ภาพที่ 4-11 ผลลัพธ์ไฟล์ที่ทำการ Download

บทที่ 5
สรุปและวิจารณ์ผลการปฏิบัติงานสหกิจศึกษา

จากการปฏิบัติงานสหกิจศึกษา ณ บริษัท เอ-โฮสต์ จำกัด (A-HOST Co., Ltd.) ในตำแหน่ง Cloud Engineer แผนก System Management Service (SMS) ตั้งแต่วันที่ 10 พฤศจิกายน พ.ศ. 2568 ถึงวันที่ 13 มีนาคม พ.ศ. 2569 ระยะเวลา 17 สัปดาห์ ผู้จัดทำได้รับมอบหมายให้พัฒนาโครงงาน "การออกแบบและพัฒนาเว็บแอปพลิเคชัน Automate Daily Monitoring" ซึ่งผลการดำเนินงานได้เสร็จสิ้นสมบูรณ์ตามวัตถุประสงค์ที่วางไว้ โดยสามารถสรุปผลการดำเนินงาน ประโยชน์ที่ได้รับ ปัญหาและอุปสรรค ตลอดจนข้อเสนอแนะได้ดังนี้

5.1 สรุปผลการดำเนินงาน

จากการพัฒนาโครงงาน ผู้จัดทำได้ทำการออกแบบและพัฒนาระบบที่ช่วยลดขั้นตอนการทำงานแบบแมนนวล (Manual Process) ในการตรวจสอบสถานะเซิร์ฟเวอร์ประจำวัน โดยระบบแบ่งการทำงานออกเป็น 2 ส่วนหลัก ได้แก่

1.ส่วน Back-end

พัฒนาด้วย n8n ทำหน้าที่เชื่อมต่อผ่านคำสั่ง SSH เพื่อดึงข้อมูลสถานะทรัพยากรเครื่อง (CPU, RAM, Disk Storage) และข้อมูลไฟล์บันทึกการทำงาน (Log Files) นำมาแปลงโครงสร้างข้อมูลให้อยู่ในรูปแบบ JSON อย่างอัตโนมัติ

2.ส่วน Front-end

พัฒนาด้วยไลบรารี React ทำหน้าที่เป็น Web Application สำหรับแสดงผลข้อมูลในรูปแบบ Centralized Dashboard ที่สามารถดูสถานะภาพรวม เปรียบเทียบข้อมูลย้อนหลัง (Compare) ดาวน์โหลดไฟล์ Log และส่งออกรายงานเป็นไฟล์เอกสารได้

ผลลัพธ์จากการดำเนินงานพบว่า ระบบสามารถใช้งานได้จริงและช่วยลดระยะเวลาในการตรวจสอบสถานะเซิร์ฟเวอร์ช่วงเช้า (Morning Check) จากเดิมที่ต้องใช้เวลา 1-2 ชั่วโมง เหลือเพียงประมาณ 5 นาที อีกทั้งยังช่วยลดข้อผิดพลาดที่เกิดจากการบันทึกข้อมูลด้วยมนุษย์ (Human Error) ทำให้การทำงานของแผนกมีความรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น

5.2 ประโยชน์ที่ได้รับจากการปฏิบัติงานสหกิจศึกษา

5.2.1 ประโยชน์ต่อตนเอง

1.ด้านทักษะทางวิชาชีพ (Hard Skills)

ได้เรียนรู้และลงมือปฏิบัติจริงในการใช้เครื่องมือต่างๆ เช่น การสร้าง Workflow อัตโนมัติด้วย n8n, การพัฒนาหน้าเว็บด้วย React, การสร้าง API สำหรับการจัดการไฟล์ รวมถึงการใช้งานคำสั่ง Linux เบื้องต้นในการจัดการเซิร์ฟเวอร์

2.ด้านกระบวนการคิดและแก้ปัญหา

ได้ฝึกทักษะการวิเคราะห์ปัญหาที่เกิดขึ้นจริงในระบบเครือข่ายและเซิร์ฟเวอร์ รวมถึงตรรกะในการคัดกรองข้อมูล (Data Parsing) จากไฟล์ Log ที่มีความซับซ้อน

3.ด้านทักษะทางสังคมและการทำงาน

ได้เรียนรู้ระบบการทำงานขององค์กร การทำงานร่วมกับทีมงานวิศวกร การสื่อสาร การรับผิดชอบต่องานที่ได้รับมอบหมาย และการปรับตัวเข้ากับสภาพแวดล้อมในการทำงานจริง

5.2.2 ประโยชน์ต่อสถานประกอบการ

1.บริษัทฯ ได้รับระบบ Automate Daily Monitoring

ที่ช่วยลดภาระงานซ้ำซ้อน (Routine Task) ของทีมงาน ทำให้บุคลากรมีเวลาในการมุ่งเน้นงานวิเคราะห์หรือแก้ไขปัญหาที่มีความซับซ้อนและสร้างมูลค่าเพิ่มได้มากขึ้น

2.เพิ่มความรวดเร็วและความแม่นยำ

ในการจัดทำรายงานสถานะเซิร์ฟเวอร์เพื่อนำส่งลูกค้า รวมถึงมีระบบที่ช่วยให้สามารถรับมือกับปัญหาเชิงรุกได้อย่างทันท่วงที

5.3 ปัญหาและอุปสรรคในการปฏิบัติงาน

ในการปฏิบัติงานและพัฒนาโครงงาน ผู้จัดทำพบปัญหาและอุปสรรคดังนี้

1.ความซับซ้อนของรูปแบบข้อมูล (Data Format)

ข้อมูลดิบที่ดึงมาจากเซิร์ฟเวอร์มีรูปแบบที่ไม่แน่นอน ทำให้ต้องใช้เวลาในการศึกษาและเขียนสคริปต์ใน n8n เพื่อกรองและแปลงข้อมูลให้อยู่ในรูปแบบโครงสร้าง JSON ที่สามารถนำไปใช้งานต่อได้อย่างถูกต้อง

2.ข้อจำกัดด้านความรู้เบื้องต้นในเครื่องมือใหม่

ในช่วงแรกของการปฏิบัติงาน ผู้จัดทำยังไม่มีความคุ้นเคยกับเครื่องมืออย่าง n8n ทำให้ต้องใช้เวลาในการศึกษาค้นคว้าเพิ่มเติมเกี่ยวกับการตั้งค่า Node ต่างๆ และการจัดการ API

3.การเชื่อมต่อและสิทธิการเข้าถึงเซิร์ฟเวอร์

การดึงข้อมูล Log File ต้องอาศัยการจัดการสิทธิ (Permissions) ที่ถูกต้อง และต้องจัดการกับข้อผิดพลาดกรณีที่เครือข่ายเกิดความล่าช้า (Timeout) ระหว่างที่ระบบอัตโนมัติกำลังทำงาน

5.4 ข้อเสนอแนะ

5.4.1 ข้อเสนอแนะในการนำระบบไปพัฒนาต่อ

1.การเพิ่มช่องทางการแจ้งเตือน

ควรมีการพัฒนาเชื่อมต่อ API เพื่อส่งการแจ้งเตือน (Alert Notification) ผ่านช่องทางแอปพลิเคชันอื่น ๆ เช่น LINE Notify หรือ Email ทันทีเมื่อระบบตรวจพบว่า CPU หรือ Disk มีการทำงานเกินค่ามาตรฐานที่กำหนดไว้

2.การรองรับคำสั่งที่ครอบคลุมขึ้น

ควรพัฒนาให้ Workflow ใน n8n สามารถตรวจสอบและรันคำสั่งที่เหมาะสมกับระบบปฏิบัติการ (OS) ของเซิร์ฟเวอร์เป้าหมายที่มีความหลากหลายได้อัตโนมัติมากยิ่งขึ้น

5.4.2 ข้อเสนอแนะสำหรับนักศึกษารุ่นต่อไป

นักศึกษาที่สนใจปฏิบัติงานในตำแหน่ง Cloud Engineer ควรเตรียมความพร้อมด้านพื้นฐานระบบปฏิบัติการ Linux, การจัดการเครือข่ายเบื้องต้น, โครงสร้างข้อมูลแบบ JSON และตรรกะการเขียนโปรแกรม เนื่องจากเป็นทักษะพื้นฐานสำคัญที่ต้องใช้ในการแก้ปัญหาและการพัฒนาระบบอัตโนมัติในระดับองค์กร

บรรณานุกรม

Docker Inc. (2025). Docker Documentation: Containerization [ออนไลน์].
เข้าถึงข้อมูลวันที่ 22 พฤศจิกายน 2568. จาก https://docs.docker.com/

GitHub. (2025). GitHub Actions and Runner Documentation [ออนไลน์].
เข้าถึงข้อมูลวันที่ 5 ธันวาคม 2568. จาก https://docs.github.com/en/actions

Meta Platforms. (2025). React: The library for web and native user interfaces [ออนไลน์].
เข้าถึงข้อมูลวันที่ 20 พฤศจิกายน 2568. จาก https://react.dev/

Microsoft. (2025). Visual Studio Code Documentation [ออนไลน์].
เข้าถึงข้อมูลวันที่ 10 ธันวาคม 2568. จาก https://code.visualstudio.com/docs

Mozilla. (2025). JavaScript Guide - MDN Web Docs [ออนไลน์].
เข้าถึงข้อมูลวันที่ 1 ธันวาคม 2568. จาก https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide

n8n. (2025). n8n Documentation: Workflow Automation [ออนไลน์].
เข้าถึงข้อมูลวันที่ 15 พฤศจิกายน 2568. จาก https://docs.n8n.io/

PostgreSQL Global Development Group. (2025). PostgreSQL Documentation [ออนไลน์].
เข้าถึงข้อมูลวันที่ 25 พฤศจิกายน 2568. จาก https://www.postgresql.org/docs/

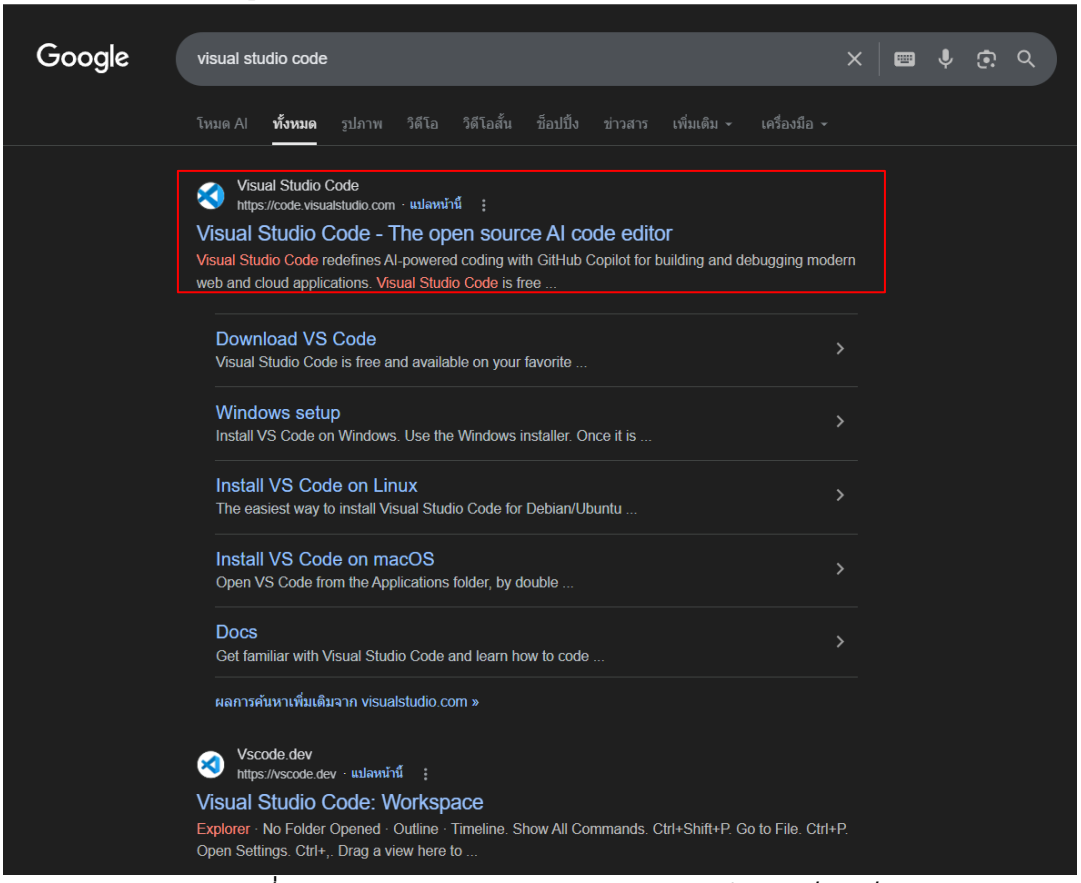
ภาคผนวก

ภาคผนวก ก
ติดตั้งโปรแกรมและส่วนขยายที่เกี่ยวข้อง

ติดตั้งโปรแกรมและส่วนขยายที่เกี่ยวข้อง

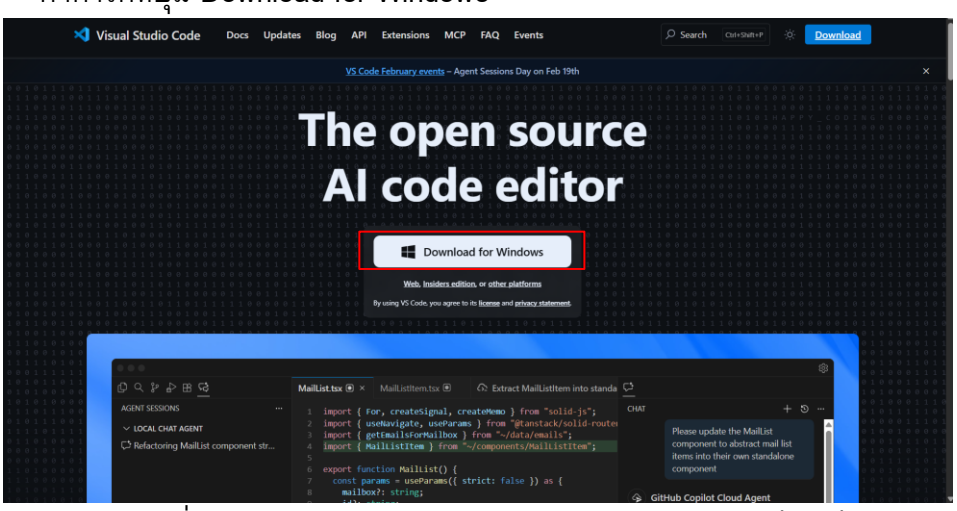
1. Visual Studio Code

เริ่มต้นด้วยการดาวน์โหลดสื่อการติดตั้ง Visual Studio Code โดยการเปิดเว็บเบราว์เซอร์ที่ เราใช้งานอยู่ เช่น Chrome แล้วค้นหาคำว่า visual studio code จากนั้นให้คลิกเข้าไป ที่ชื่อตามรูปภาพ แสดงให้เห็นดังภาพที่ ก-1



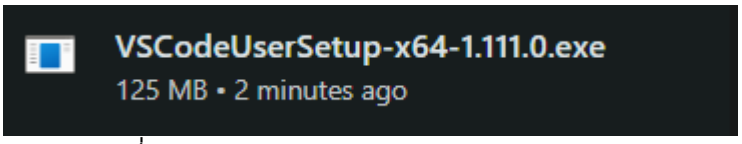
ภาพที่ ก-1 ค้นหา Visual Studio Code บนเว็บบราว์เซอร์

ทำการกดปุ่ม Download for Windows



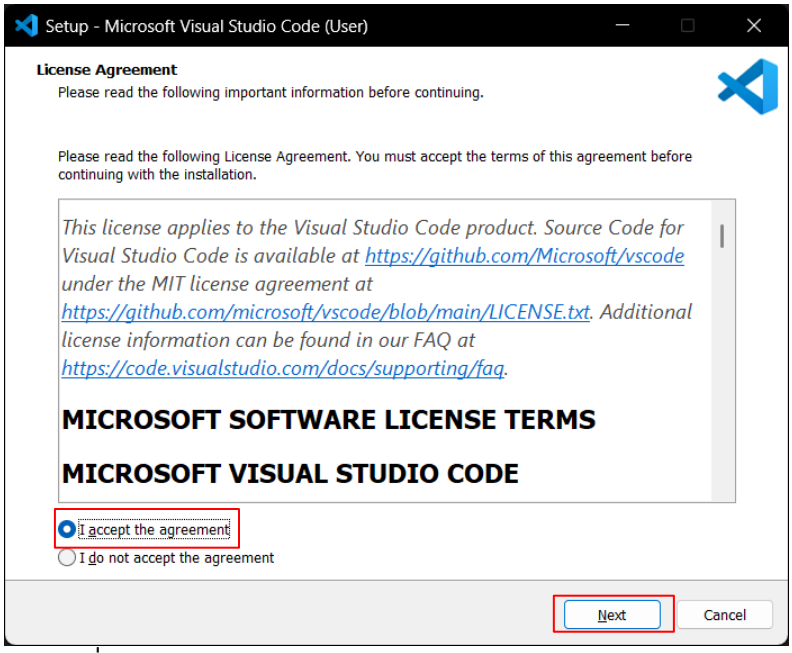
ภาพที่ ก-2 Download Visual Studio Code บนเว็บบราว์เซอร์

รอ Download จนเสร็จสิ้นและทำการเปิดโปรแกรม



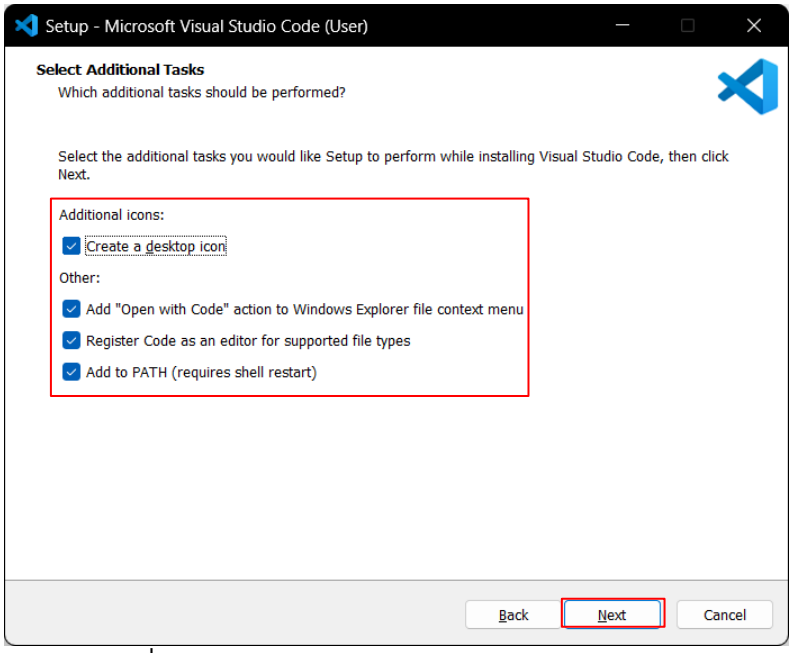
ภาพที่ ก-3 ผลการ Download Visual Studio Code

กด Accept agreement และกด Next



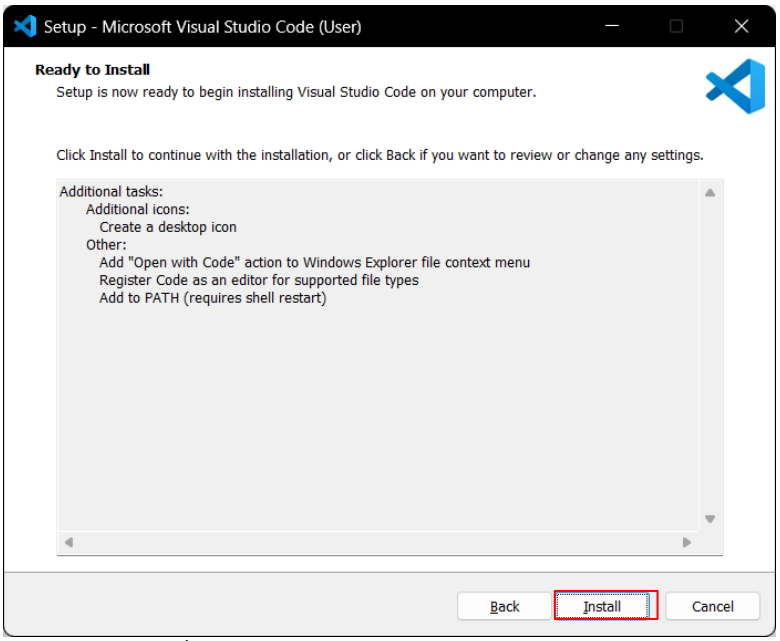
ภาพที่ ก-4 ผลการกด Accept agreement Visual Studio Code

กด add tasks ทั้งหมด และกด Next



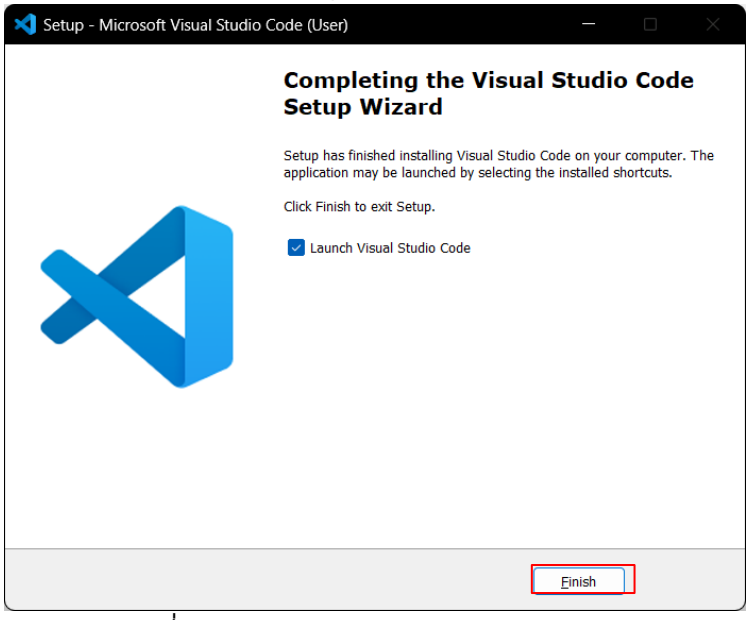
ภาพที่ ก-5 ผลการกด add tasks Visual Studio Code

ทำการกด Install เพื่อติดตั้ง



ภาพที่ ก-6 การกด install Visual Studio Code

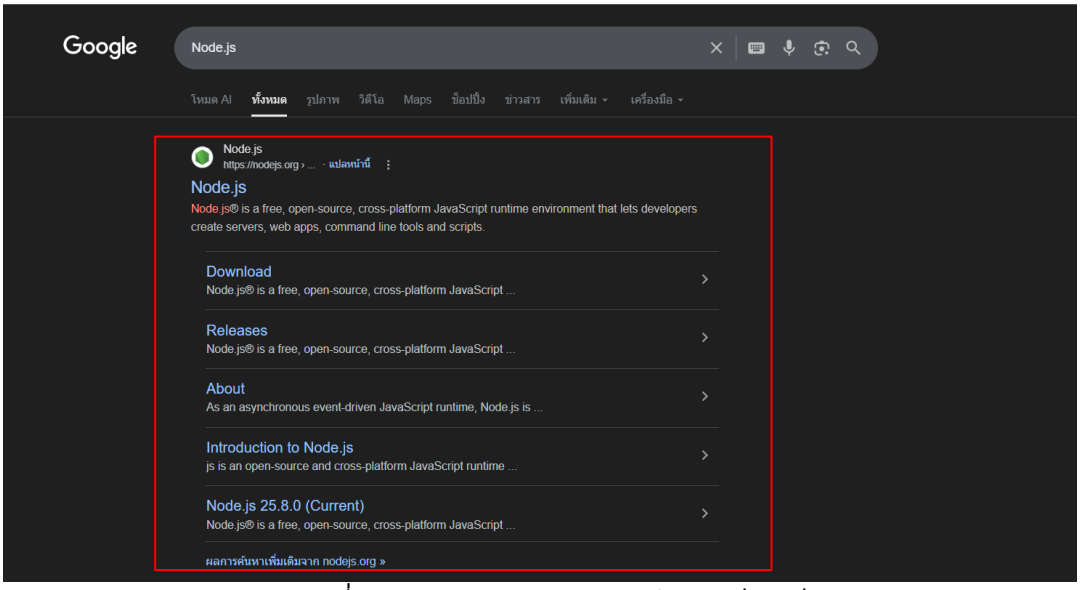
รอการติดตั้งจนสำเร็จระบบจะขึ้นปุ่ม Finish มาให้กด



ภาพที่ ก-7 Finish Install Visual Studio Code

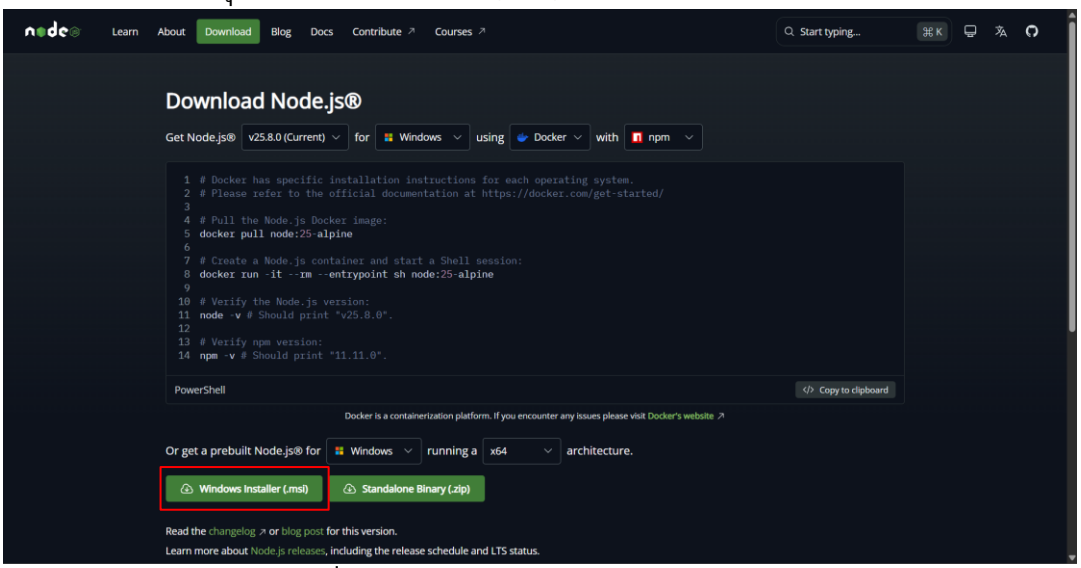
2. Node.js

เริ่มต้นด้วยการดาวน์โหลดสื่อการติดตั้ง Node.js โดยการเปิดเว็บเบราว์เซอร์ที่ เราใช้งานอยู่ เช่น Chrome แล้วค้นหาคำว่า Node.js จากนั้นให้คลิกเข้าไป ที่ชื่อตามรูปภาพแสดงให้เห็นดังภาพที่ ก-8



ภาพที่ ก-8 ค้นหา Node.js บนเว็บบราว์เซอร์

ทำการกดปุ่ม Windows Installer (.msi) เพื่อทำการ Download



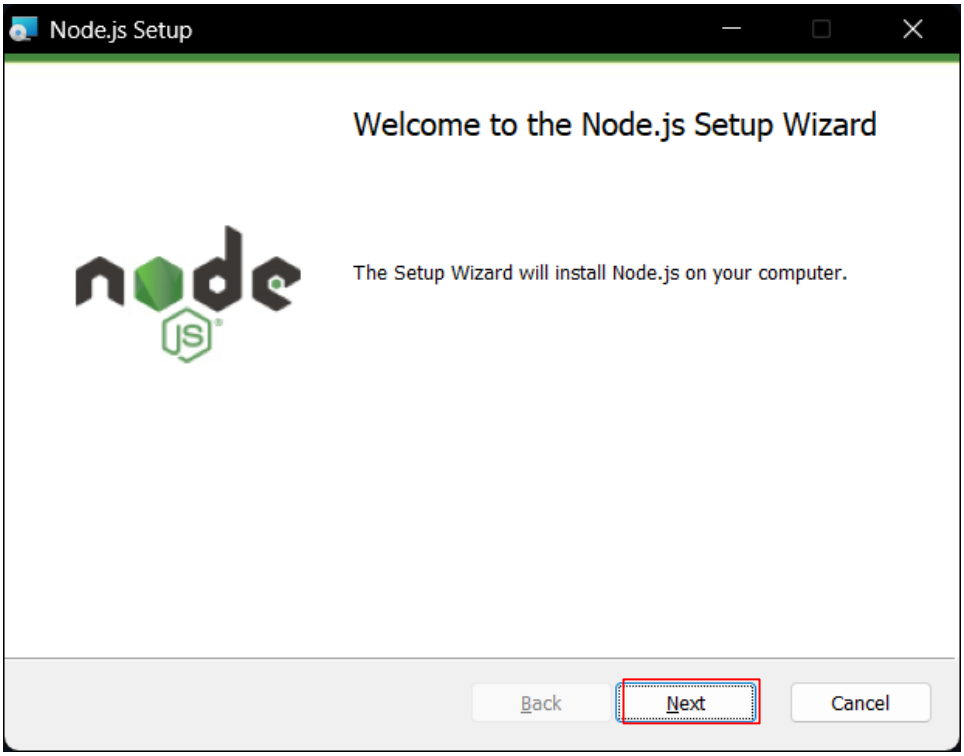
ภาพที่ ก-9 Node.js Windows Installer (.msi)

รอ Download จนเสร็จสิ้นและทำการเปิดโปรแกรม



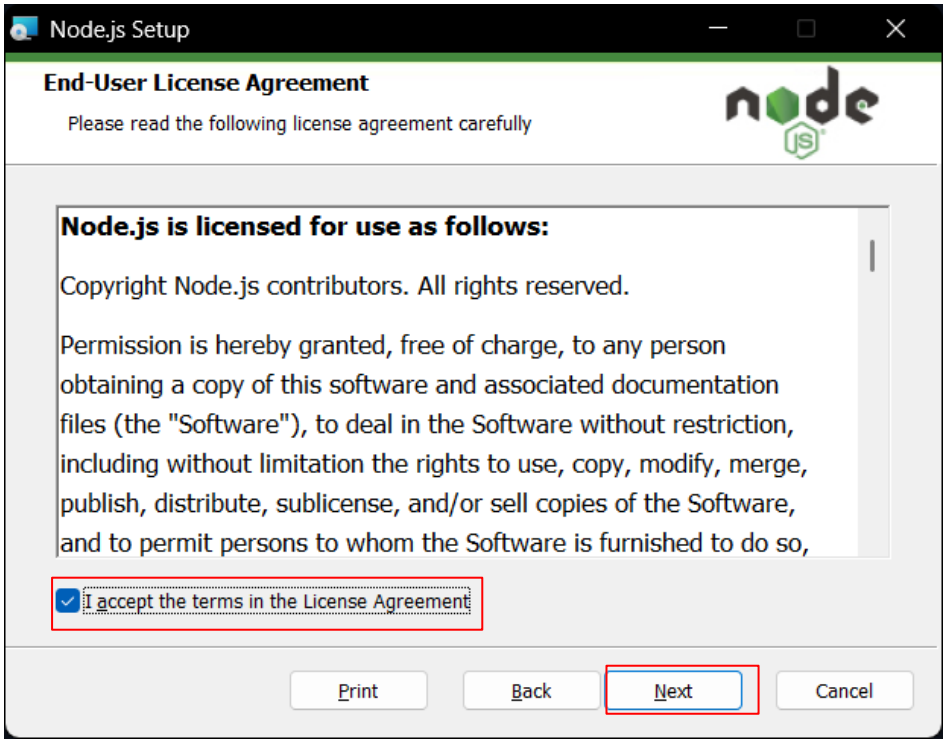
ภาพที่ ก-10 ผลการ Download Node.js

ทำการกด Next เพื่อติดตั้ง



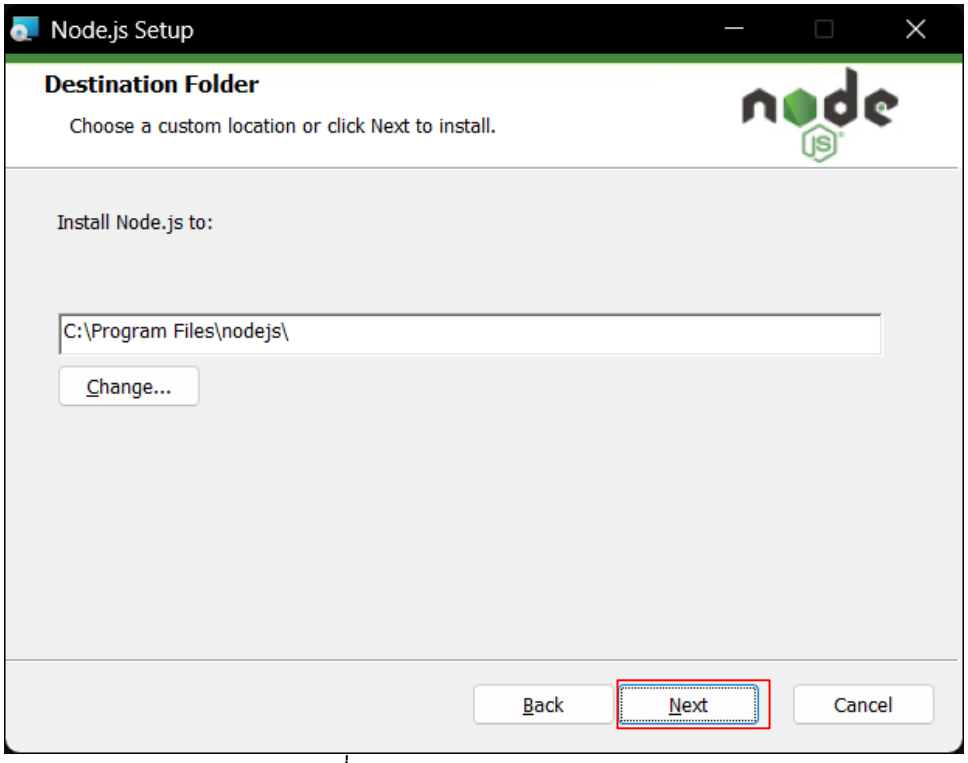
ภาพที่ ก-11 Node.js Setup Wizard Next

กด Accept agreement และกด Next



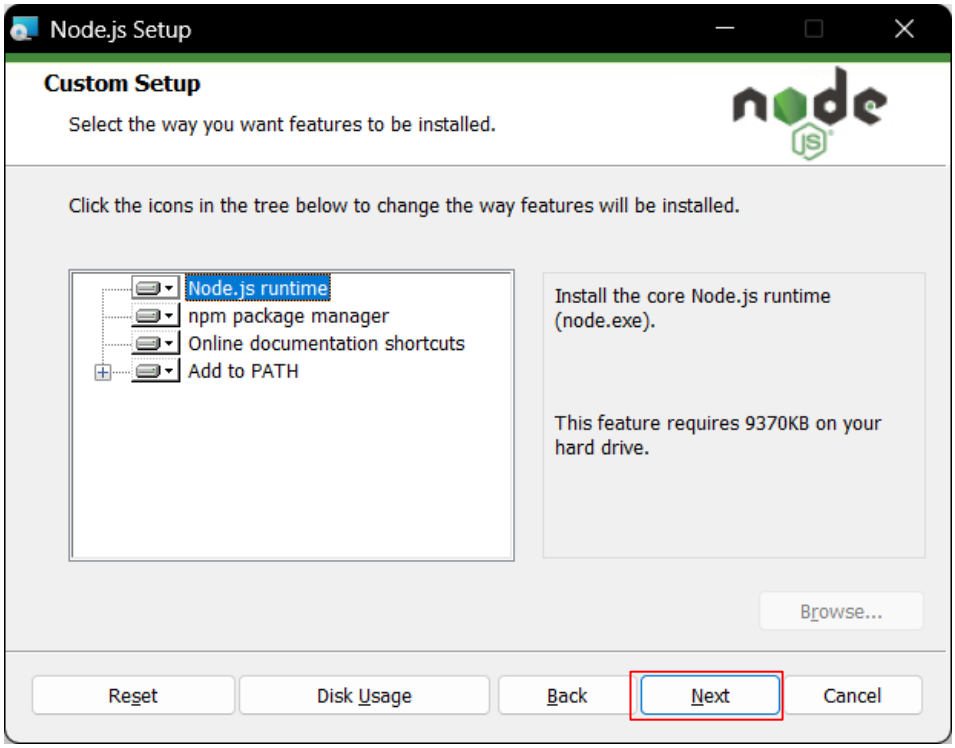
ภาพที่ ก-12 ผลการกด Accept agreement Node.js

กำหนด Folder ที่ต้องการ Install



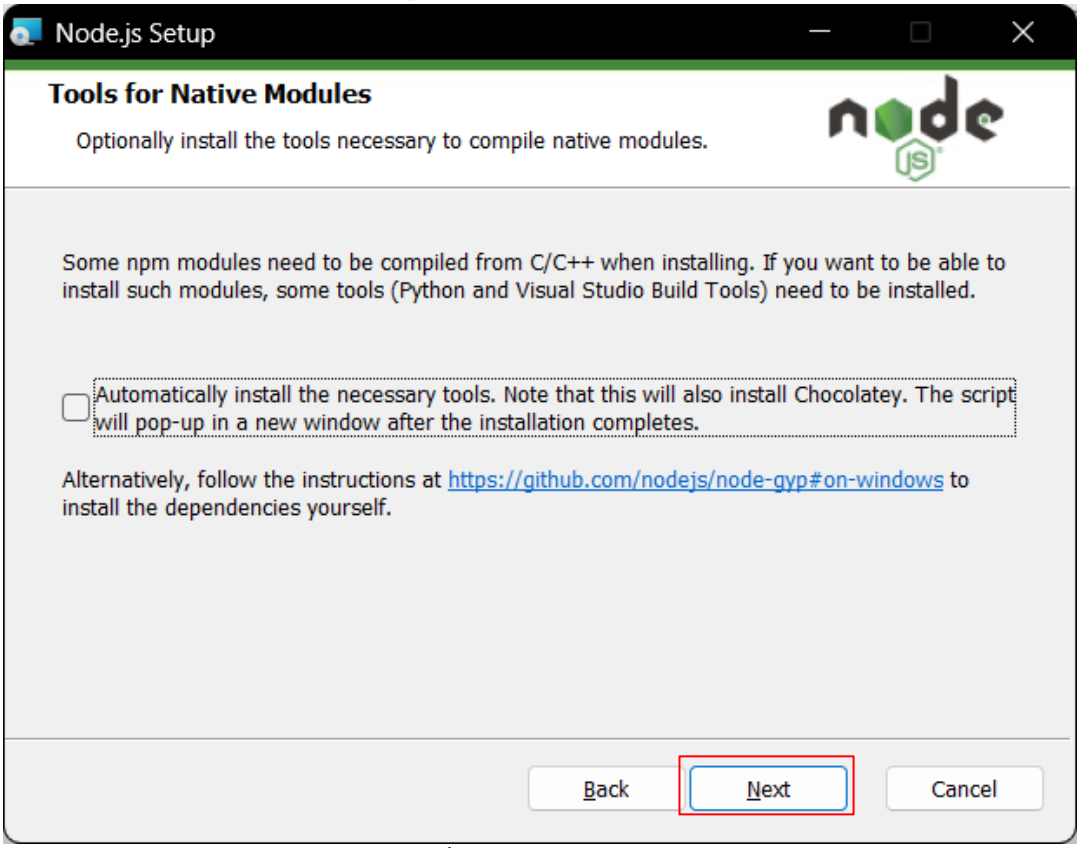
ภาพที่ ก-13 Folder Install Node.js

Custom Setup และกด Next



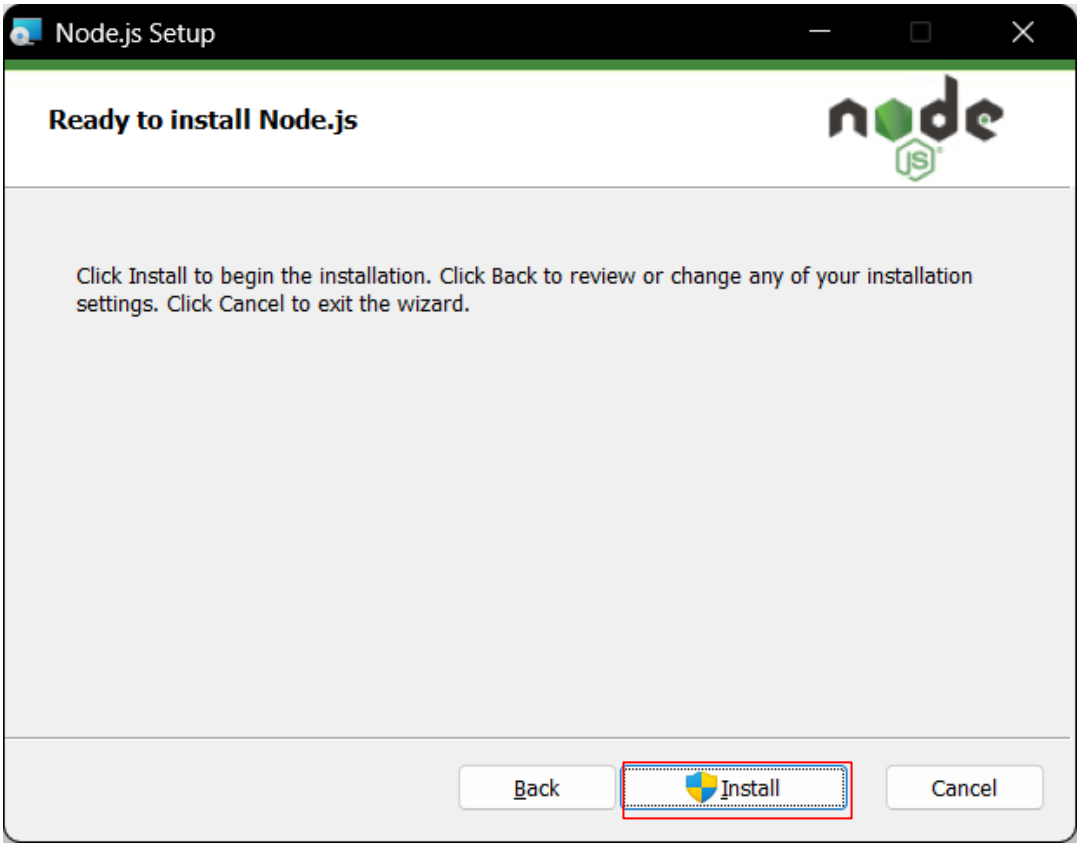
ภาพที่ ก-14 Custom Setup Node.js

หากต้องการ Tools เสริมให้กดติกถูก และกด Next



ภาพที่ ก-15 Tools Node.js

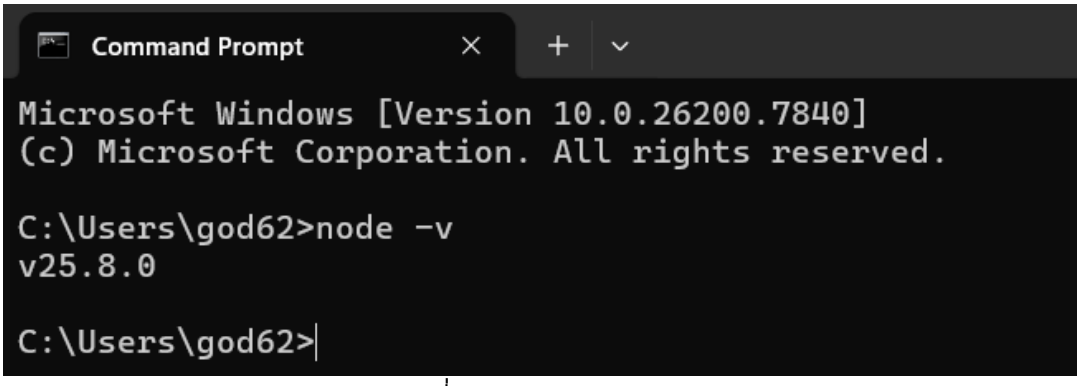
ทำการ Install และรอให้เสร็จสิ้น



ภาพที่ ก-16 Install Node.js

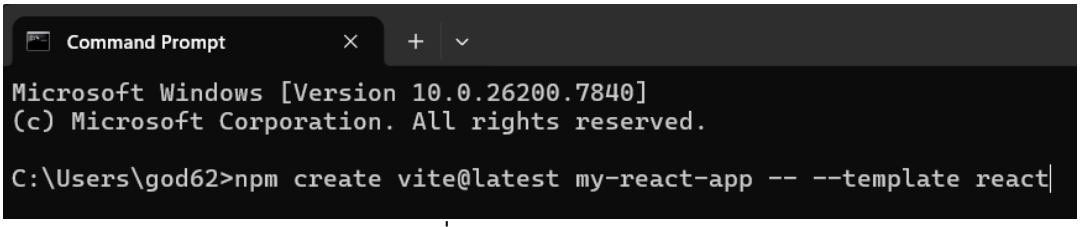
2.2 การใช้งาน React

หลังการ Install Node.js เสร็จสริงให้เช็คว่า Node ติดตั้งสำเร็จโดยการเช็คเวอร์ชั่นผ่าน Command Prompt ด้วยคำสั่ง node -v



ภาพที่ ก-17 Node.js check

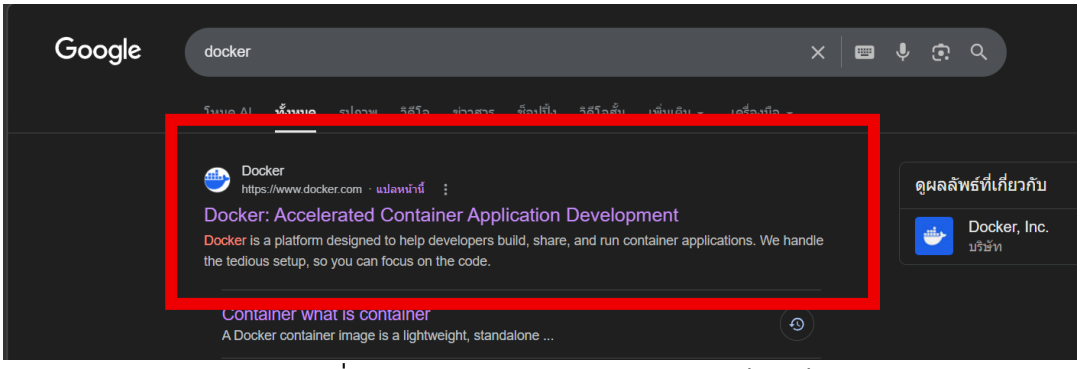
หากมีการ Install Node.js เราสามารถ สร้าง Project ด้วย React ได้ทันที



ภาพที่ ก-18 Node.js React

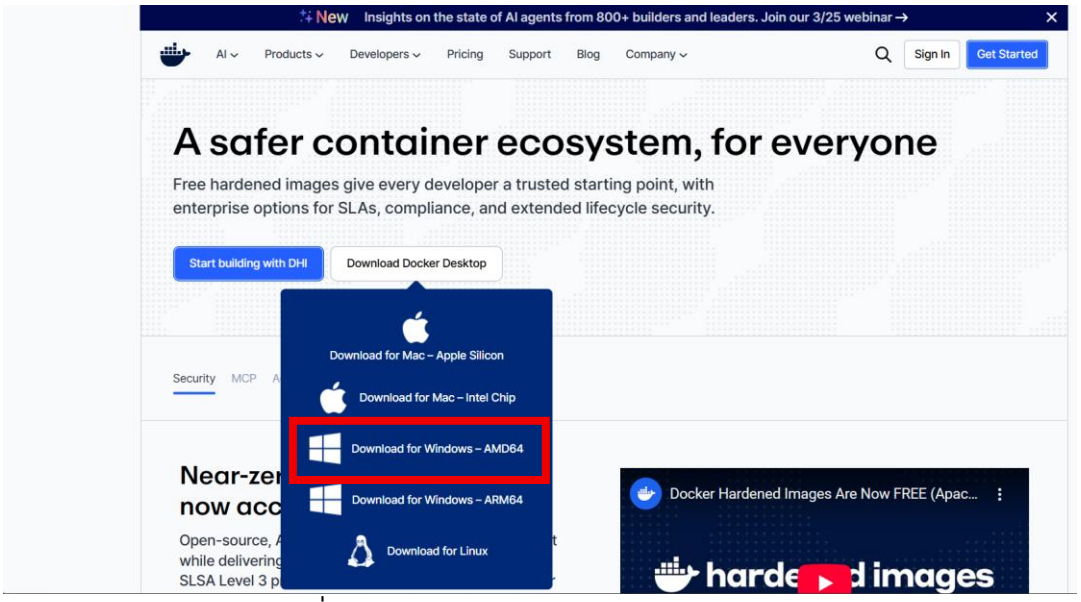
3. ติดตั้ง Docker

เริ่มต้นด้วยการดาวน์โหลด Docker Desktop โดยการเปิดเว็บบราว์เซอร์ที่เราใช้งานอยู่ เช่น Edge , Chrome แล้วค้นหาคำว่า Docker จากนั้นให้คลิกเข้าไปที่เว็บตามรูปภาพที่แสดงให้เห็นดัง



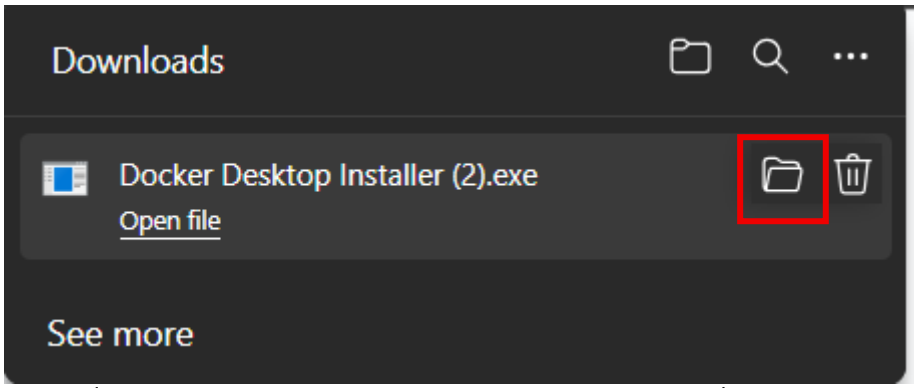
ภาพที่ ก-19 ค้นหา Docker บนเว็บบราว์เซอร์

เมื่อเข้ามาหน้านี้แล้วนั้นให้เลือกดาวน์โหลดตามระบบปฏิบัติการของตัวเอง ซึ่งในที่นี้จะโหลดเป็น Download for Window - AMD64



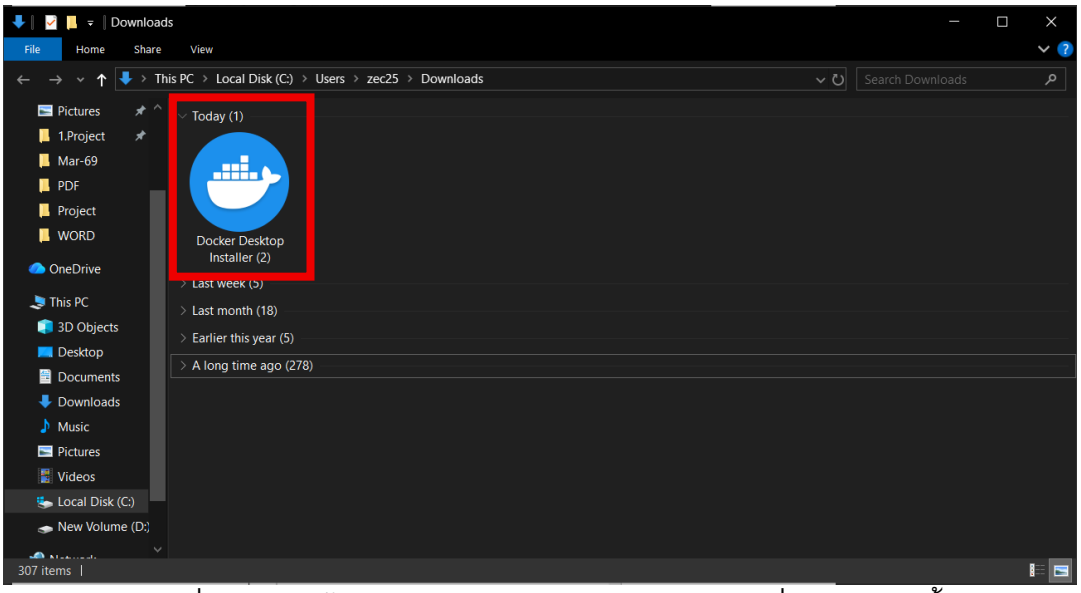
ภาพที่ ก-20 Download for Windows – AMD 64

เมื่อดาวน์โหลดเสร็จแล้ว ให้กดเข้าไปที่โฟลเดอร์ของ Docker Desktop



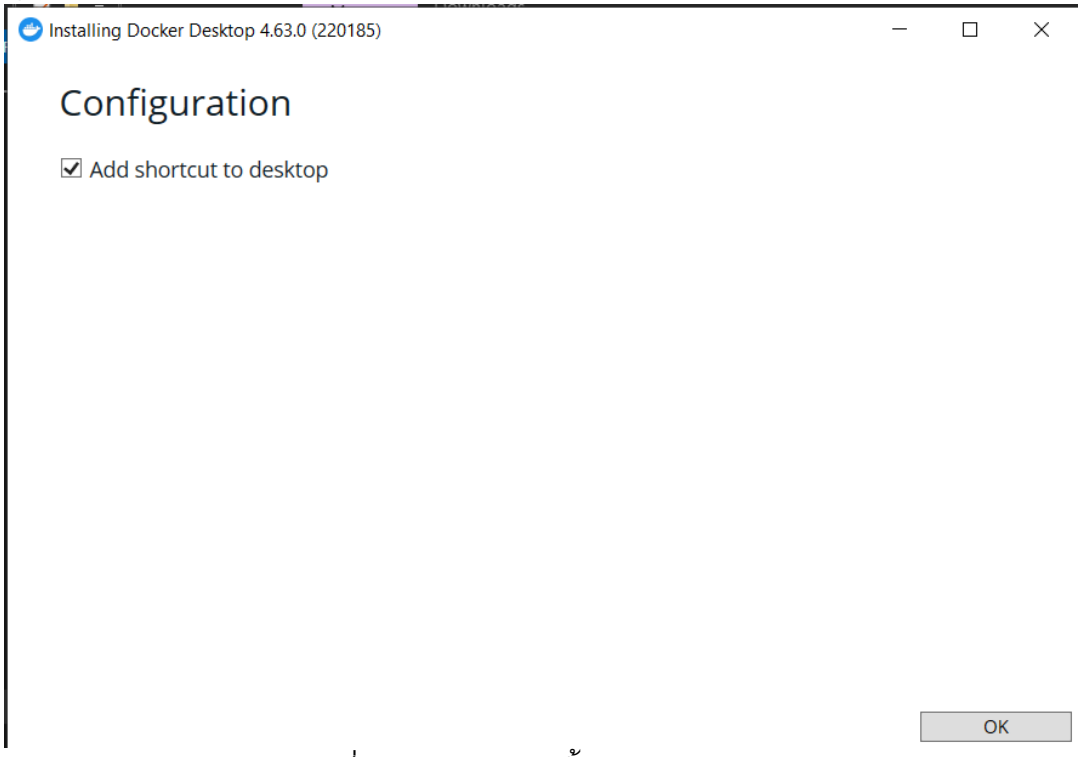
ภาพที่ ก-21 เปิดโฟลเดอร์ของ Docker Desktop Installer ที่ดาวน์โหลดมา

หลังจากที่เข้าโฟลเดอร์ที่ดาวโหลดมาแล้วนั้น ให้ดับเบิ้ลคลิกที่ไฟล์ Docker Desktop Install.exe เพื่อทำการติดตั้ง



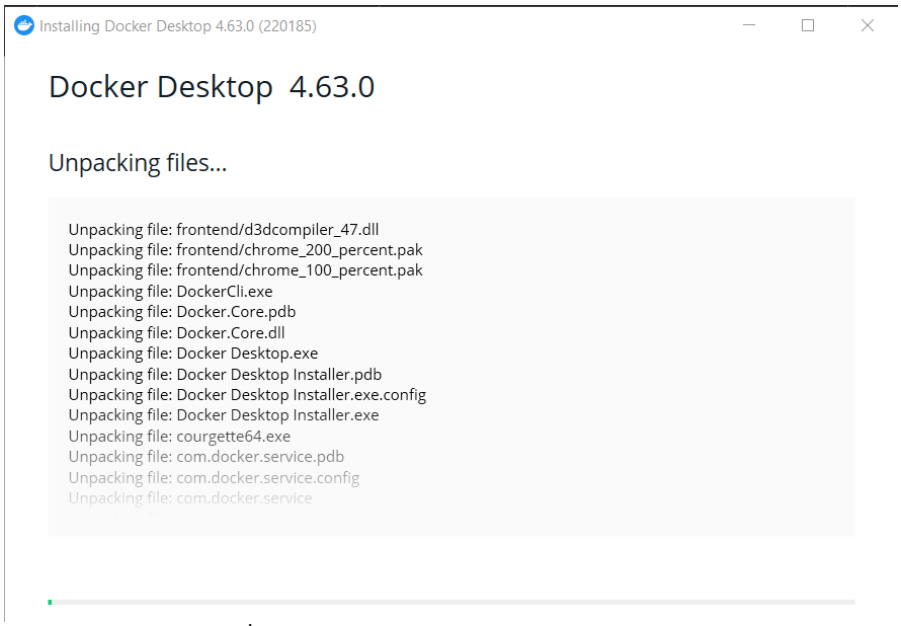
ภาพที่ ก-22 เปิดไฟล์ Docker Desktop Installer.exe เพื่อทำการติดตั้ง

หลังจากกดเปิดไฟล์ Docker Desktop Installer.exe แล้วจะขึ้นหน้าติดตั้งมาให้ทำเครื่องหมายถูกตรง Add Shortcut to desktop เพื่อความสะดวกสบายในการใช้งาน แล้วกดปุ่ม OK



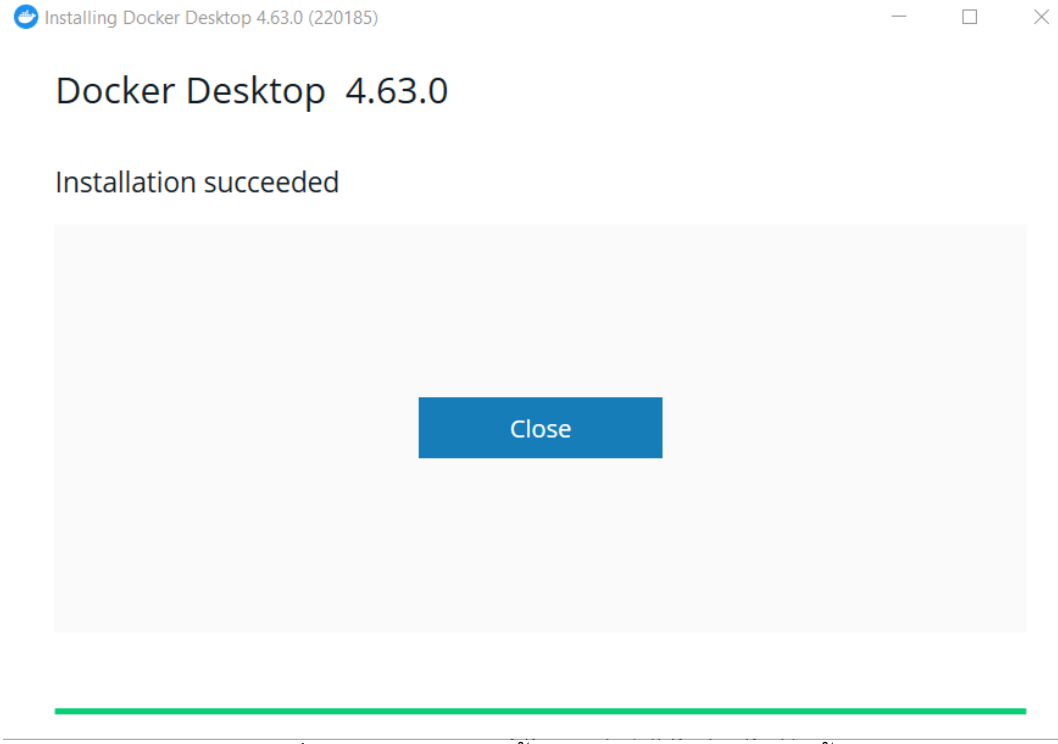
ภาพที่ ก-23 หน้าจอติดตั้ง Docker Desktop

หลังจากกด ปุ่ม Ok ไปแล้วนั้นจะขึ้นขั้นตอนของ Process ของการติดตั้งให้เรารอจนกว่าจะเสร็จ



ภาพที่ ก-24 Process Install Docker Desktop

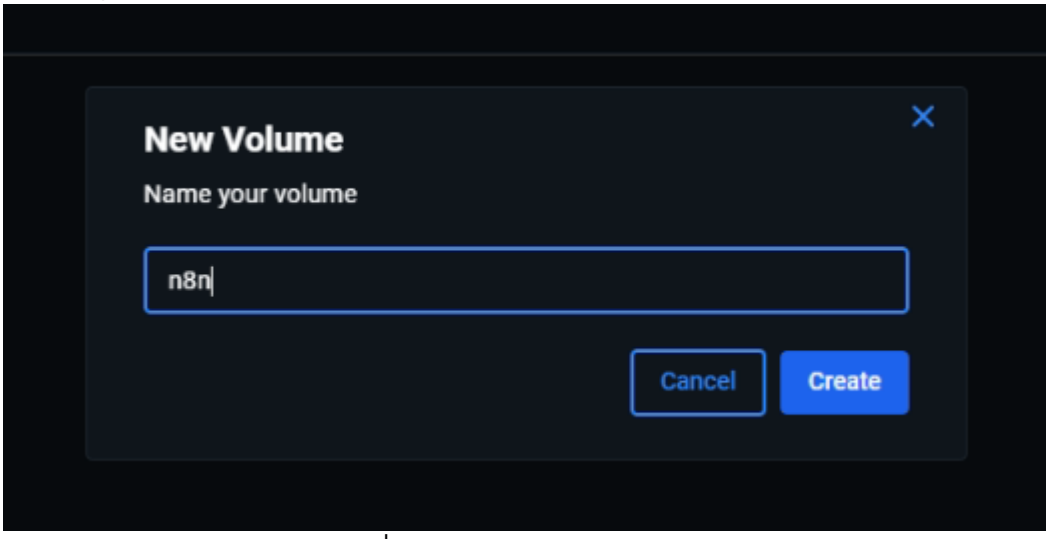
หลังจากที่ติดตั้งเสร็จแล้วนั้นจะขึ้นหน้าจอว่า Installation succeeded แล้วมีให้กดปุ่ม close



ภาพที่ ก-25 หน้าจอติดตั้ง Docker Desktop เสร็จสิ้น

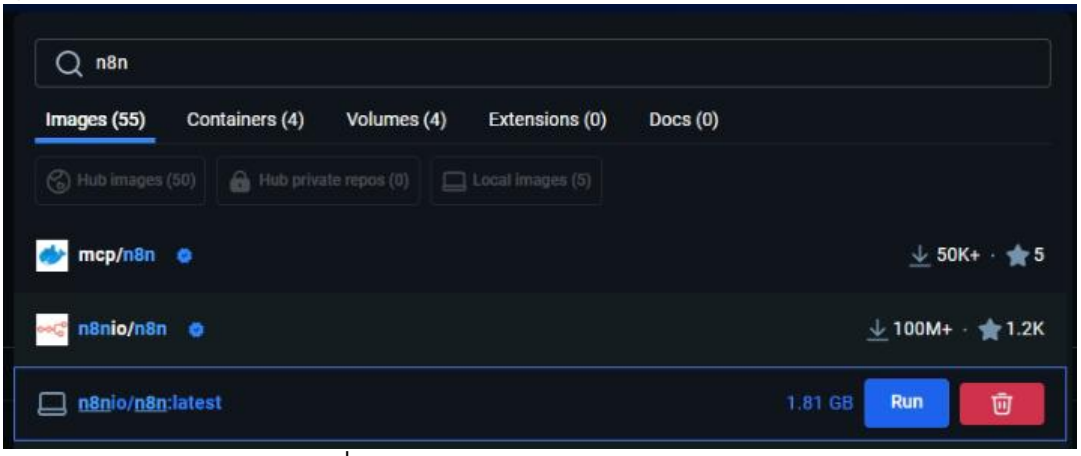
4. n8n

เปิดโปรแกรม Docker Desktop แล้วให้เรากดสร้าง volumes ใหม่ ตั้งชื่อว่า n8n แล้วกดปุ่ม Create



ภาพที่ ก-26 หน้าจอกดสร้าง Volumes

จากนั้นให้ไปกดที่ Images เพื่อทำการติดตั้ง Images ของ n8n เพื่อใช้งาน แล้วกดปุ่ม RUN เพื่อทำการติดตั้ง n8n มาใช้งานใน Docker



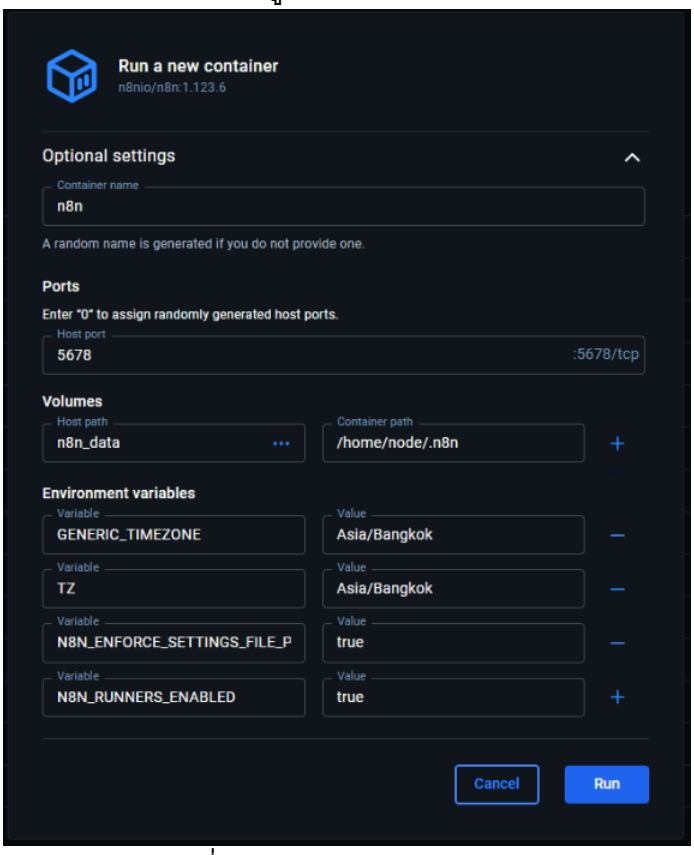
ภาพที่ ก-27 หน้าจอ Download Images n8n

หลังจากที่ดาวน์โหลดเสร็จสิ้นแล้ว ให้ทำการกดปุ่ม Run ดังภาพที่



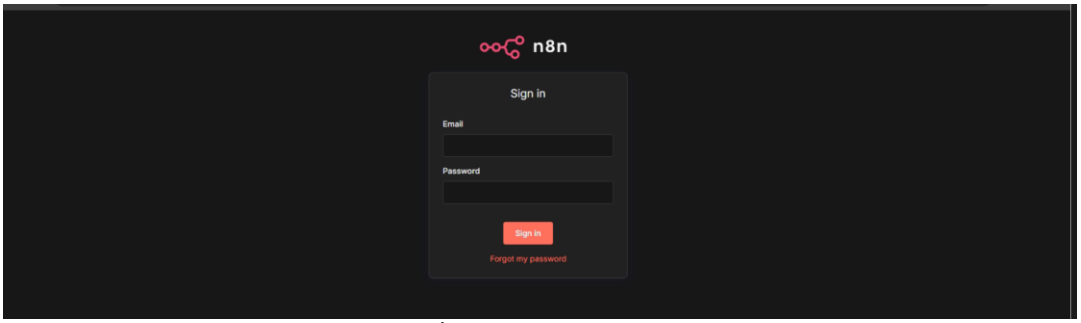
ภาพที่ ก-28 images n8n

หลังจากนั้นให้ทำการ กรอกข้อมูลของการ Config n8n ผ่าน Docker



ภาพที่ ก-29 Config n8n container

เมื่อ Config เสร็จแล้วเราจะรอสักแปบนึงเพื่อให้ Docker ทำงานเสร็จสิ้นจะเปิดหน้าเว็บของ n8n ขึ้นมา



ภาพที่ ก-30 หน้าจอเว็บ n8n

ภาคผนวก ข
คู่มือการใช้งานระบบ

คู่มือการใช้งานระบบ Auto Monitoring

1. การเข้าใช้งานและการอัปเดตข้อมูล

การเข้าสู่ระบบ: เข้าใช้งานผ่าน URL ที่กำหนดไว้

การรีเฟรชข้อมูล (Refresh): Refresh All (ปุ่มหลัก): สำหรับการรีเฟรชข้อมูลของระบบทั้งหมดทุกภาคส่วน

Refresh (เฉพาะส่วน): สำหรับการรีเฟรชข้อมูลเฉพาะเซิร์ฟเวอร์ที่ต้องการ เพื่อดูสถานะล่าสุด



ภาพที่ ข-1 หน้าจอ Home

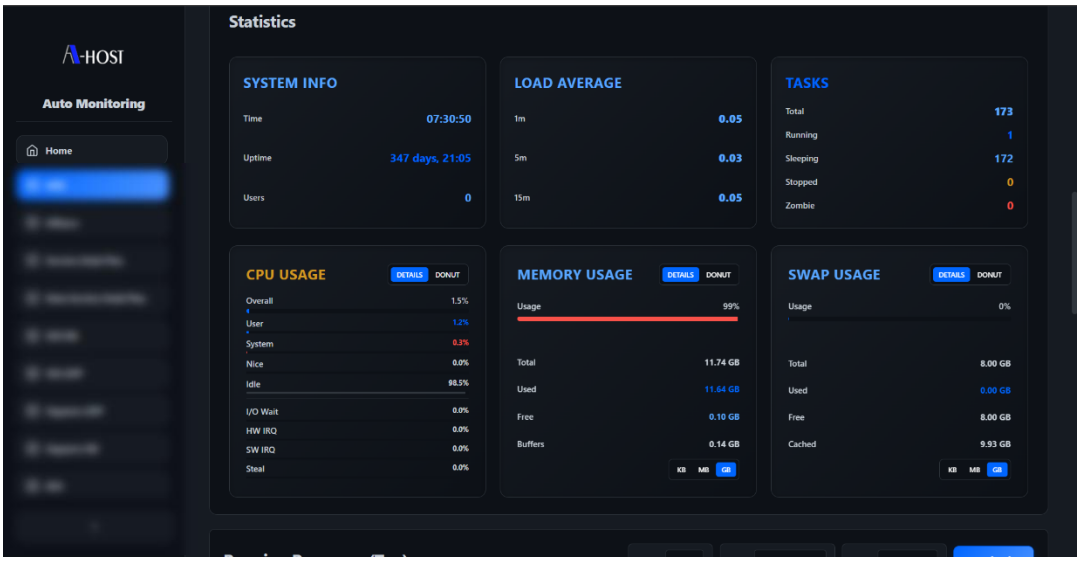
2.การตรวจสอบและเปรียบเทียบข้อมูล (Table & Top View)

การเลือกช่วงเวลา (Date Range): สามารถเลือกช่วงวันที่ต้องการเพื่อนำข้อมูลมาเปรียบเทียบกันได้ ทั้งแบบรายเดือนและรายปี

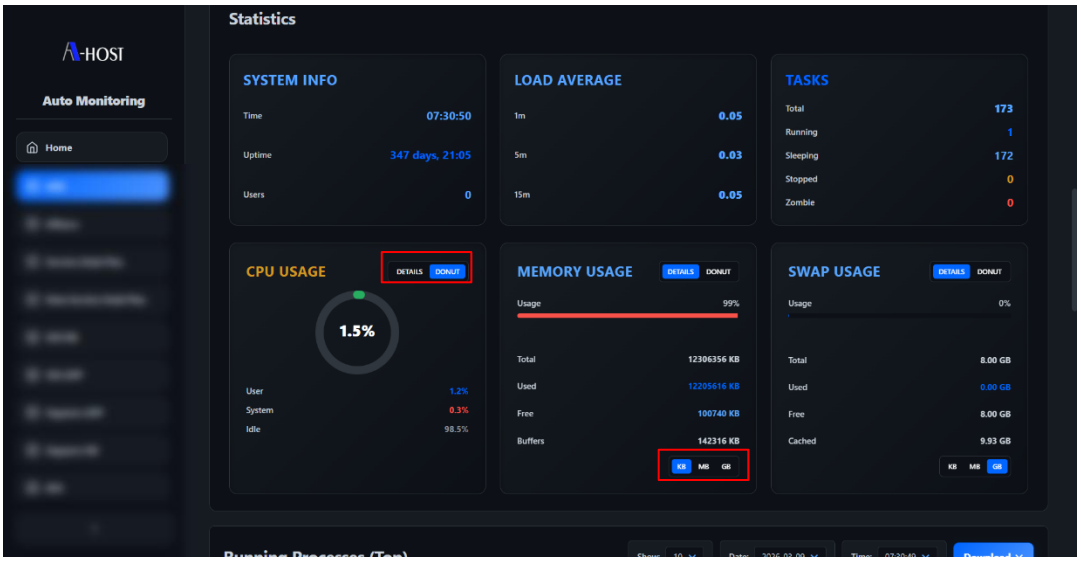


ภาพที่ ข-2 การเลือกช่วงเวลา (Date Range) Trend Charts

สามารถปรับเปลี่ยนหน่วยของข้อมูลได้ตามต้องการระหว่าง GB, MB และ KB
รูปแบบการแสดงผล: เลือกรับชมข้อมูลได้ทั้งแบบ รายละเอียด (Detail) หรือแบบกราฟ (Donut)



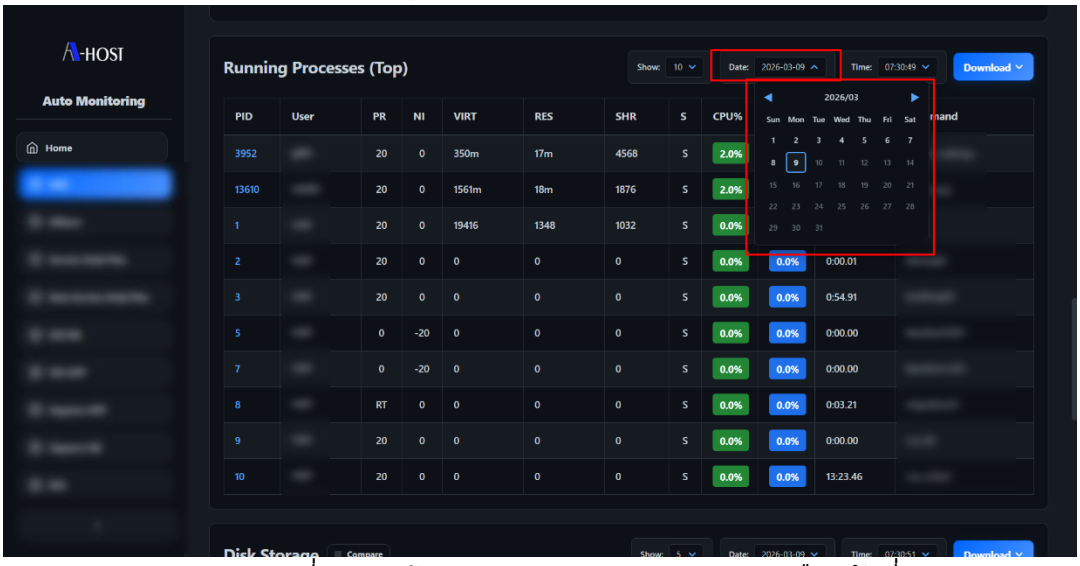
ภาพที่ ข-3 หน้าจอ Statistics



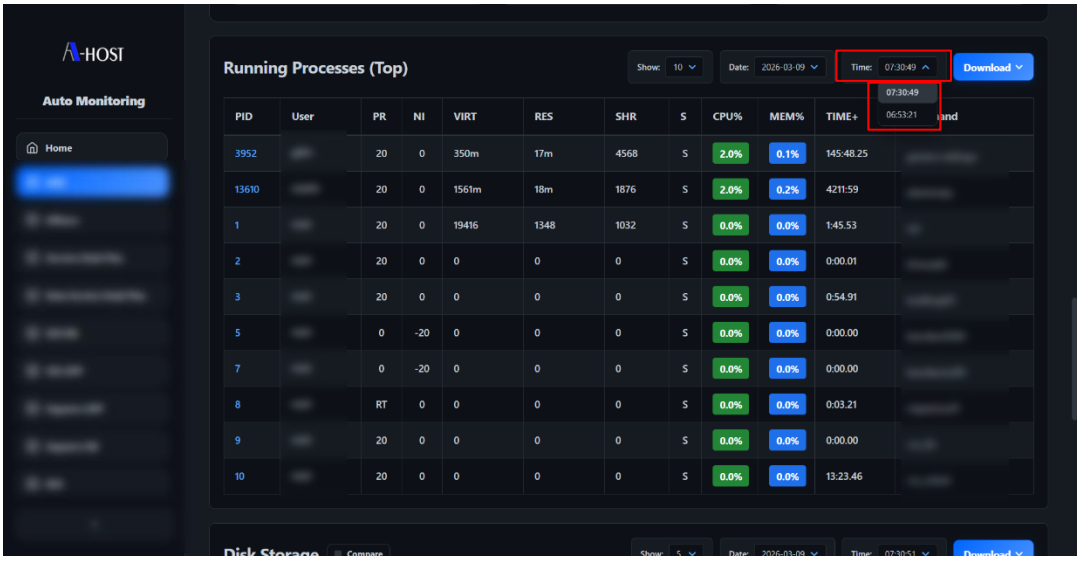
ภาพที่ ข-4 หน้าจอ Statistics รูปแบบการแสดงผลแบบ Donut และปรับค่า KB

3. การตรวจสอบกระบวนการทำงาน (Running Processes)

การดูข้อมูลย้อนหลัง: ในส่วนของ Top Running Processes สามารถเรียกดูข้อมูลย้อนหลังได้ โดยการระบุวันที่และเวลาที่ต้องการตรวจสอบ



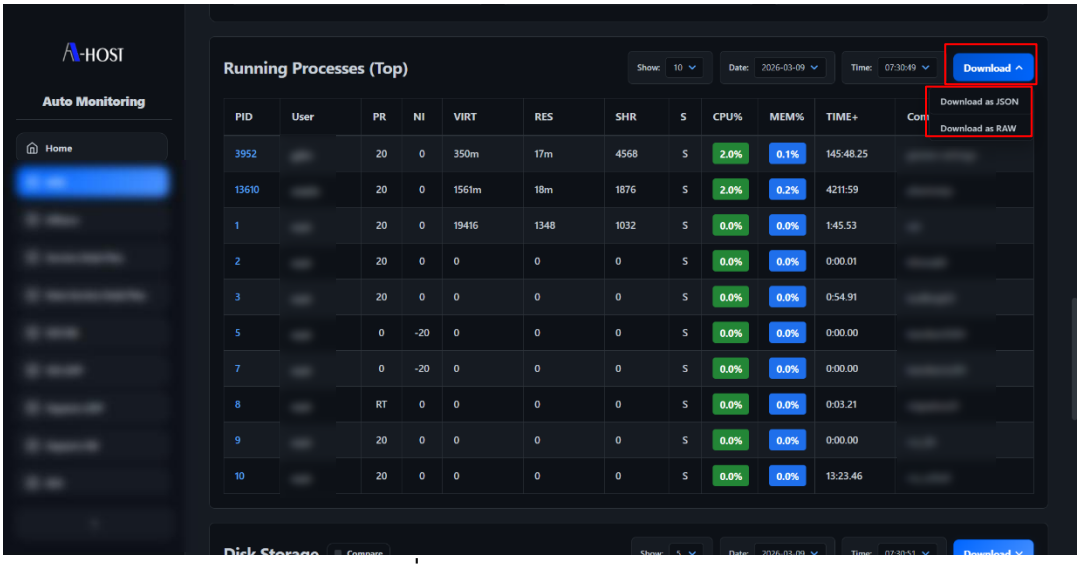
ภาพที่ ข-5 หน้าจอ Top Running Processes เลือกวันที่



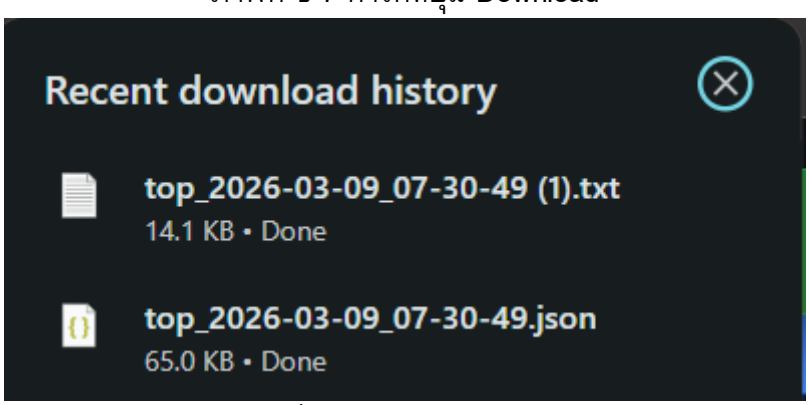
ภาพที่ ข-6 หน้าจอ Top Running Processes เลือกเวลา

4.การดาวน์โหลด (Download)

รูปแบบไฟล์: สามารถดาวน์โหลดข้อมูลได้ 2 รูปแบบ คือ JSON และ RAW (.txt)



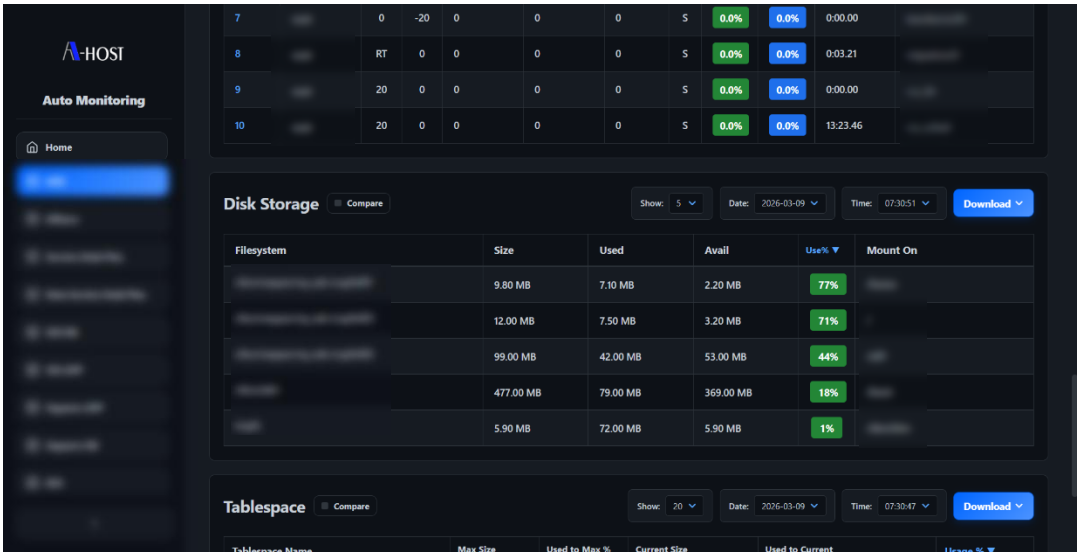
ภาพที่ ข-7 การกดปุ่ม Download



ภาพที่ ข-8 ผลการ Download

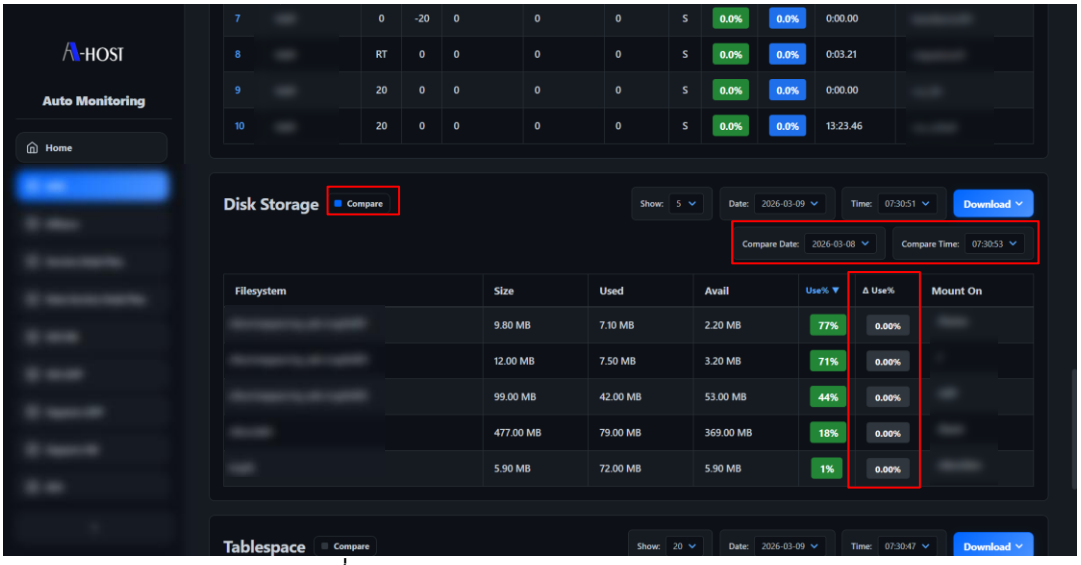
5.การตรวจสอบพื้นที่จัดเก็บข้อมูล (Disk Storage)

ระบบเปรียบเทียบ (Compare): ส่วนนี้มีความพิเศษกว่าหน้า Top คือสามารถเปรียบเทียบค่าปัจจุบันกับค่าในอดีตได้

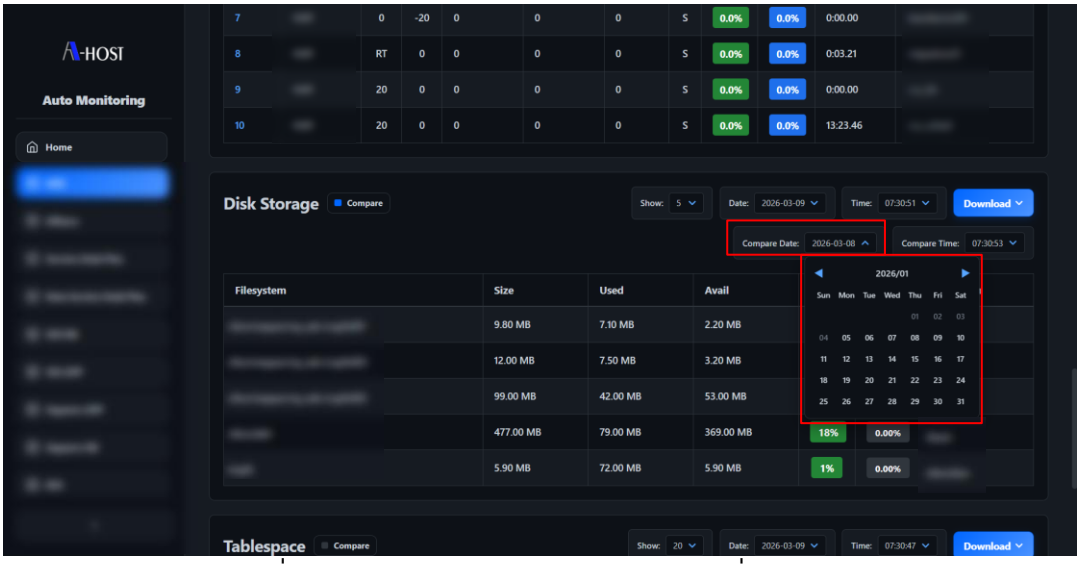


ภาพที่ ข-9 หน้าจอ Disk Storage

ขั้นตอน: เลือกวันและเวลาที่ต้องการเปรียบเทียบ หากข้อมูลมีความเปลี่ยนแปลง ระบบจะแสดงผลต่างให้เห็นอย่างชัดเจน



ภาพที่ ข-10 หน้าจอ Disk Storage Compare



ภาพที่ ข-11 หน้าจอ Disk Storage เลือกวันที่ Compare

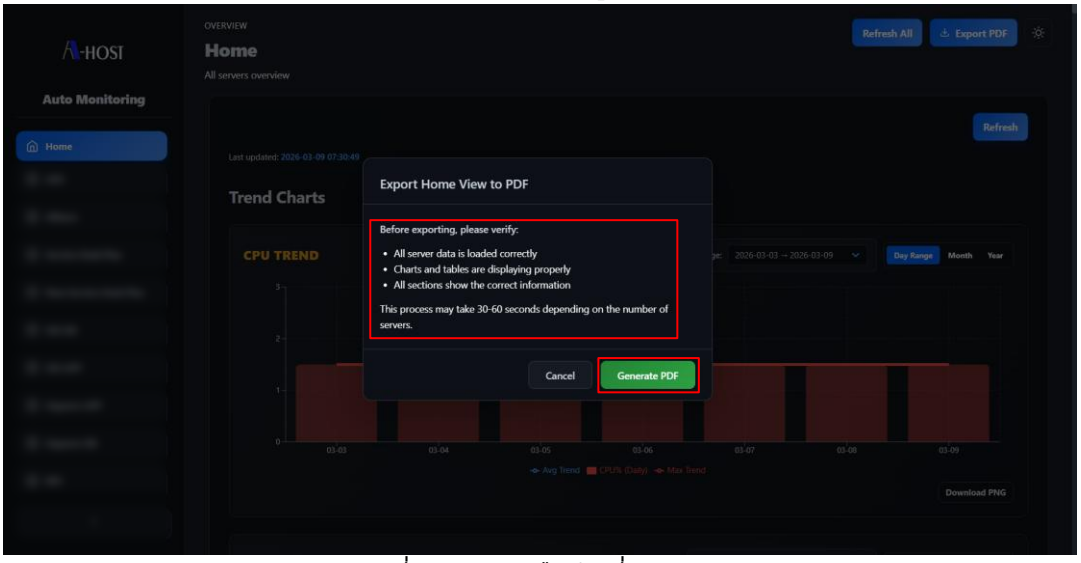
6.การส่งออกข้อมูล (Export)

การ Export ข้อมูลรวม: หากต้องการส่งออกข้อมูลของ จำเป็นต้องกลับไปที่หน้า Home

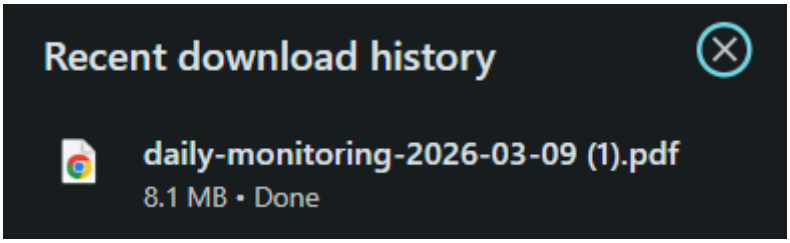


ภาพที่ ข-12 หน้าจอ Home

ข้อควรระวัง: ก่อนทำการ Export กรุณาตรวจสอบความถูกต้องของข้อมูลทุกเซิร์ฟเวอร์ และตรวจสอบว่าระบบโหลดข้อมูลครบถ้วนแล้วหรือไม่



ภาพที่ ข-13 การยืนยันที่จะ Export



ภาพที่ ข-14 ผลการ Export

ภาคผนวก ค
ใบนำส่งและเอกสารสหกิจที่เกี่ยวข้อง

แบบแจ้งรายละเอียดการปฏิบัติงาน และแผนที่ตั้งสถานประกอบการ (IN-SOO4)

งานสหกิจศึกษา คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

IN-S004

แบบแจ้งรายละเอียดที่พักระหว่างการปฏิบัติงานสหกิจศึกษา รายละเอียดงาน ตำแหน่งงาน พนักงานที่ปรึกษา และแผนที่ตั้งสถานประกอบการ

(ผู้ให้ข้อมูล : นิสิต)

เรียน คณบดี คณะ คณะวิทยาการสารสนเทศ **มหาวิทยาลัยบูรพา**

ขอแจ้งรายละเอียดที่พักระหว่างการปฏิบัติงานสหกิจศึกษา รายละเอียดงาน ตำแหน่งงาน พนักงานที่ปรึกษา และแผนที่ตั้งสถานประกอบการ ดังนี้

ข้อมูลนิสิต

ชื่อ-นามสกุล ธาม แก้วเกษ

รหัสประจำตัวนิสิต 65160172 สาขาวิชา คอมพิวเตอร์

โทรศัพท์ 0624739152 E-mail 65160172@go.buu.ac.th

ตำแหน่งงานที่นิสิตปฏิบัติ (Job Position) Cloud Engineer

ชื่อ ที่อยู่ของสถานประกอบการที่ปฏิบัติงานสหกิจศึกษา

ชื่อสถานประกอบการ (ไทย หรือ อังกฤษ) A-HOST Co.,Ltd.(Head Office) อาคารเอส เอ็ม ทาวเวอร์ ชั้น 21

เลขที่ 979/52-55 ถนน ถนน พหลโยธิน ซอย -

ตำบล พญาไท อำเภอ พญาไท

จังหวัด กรุงเทพ รหัสไปรษณีย์ 10400 โทรศัพท์ 02 298 0625

มือถือ - E-mail channelmarketing@a-host.co.th

พนักงานที่ปรึกษา (Job Supervisor) ของนิสิต

ชื่อ-นามสกุล เจตนิพิฐ พสิษฐังกูร

ตำแหน่ง Assistant data engineer แผนก/ฝ่าย SMS

โทรศัพท์ 081-9922026 E-mail jetnipit@a-host.co.th

แนบแผนที่ตั้งสถานประกอบการ

เพื่อความสะดวกในการนิเทศงานสหกิจศึกษา โปรดระบุชื่อถนนและสถานที่สำคัญ ใกล้เคียงที่สามารถเข้าใจง่าย

ธนาคารกสิกรไทย  SM Tower   รพ.พญาไท 2 

ลงชื่อ ธาม แก้วเกษ ผู้ให้ข้อมูล

(ธาม แก้วเกษ)

วันที่ 25 พ.ย. 2568

กรุณาส่งเอกสารฉบับนี้คืนงานสหกิจศึกษาภายในสัปดาห์ที่ 4 ของการปฏิบัติงานของนิสิต
โปรดมอบหมายให้นิสิตเป็นผู้ดำเนินการ โดยจัดส่งทาง E-mail ให้คณาจารย์นิเทศสหกิจศึกษา

แบบแจ้งแผนปฏิบัติงานสหกิจศึกษา (IN-SOO5)

งานสหกิจศึกษา คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

IN-S005

แบบแจ้งแผนปฏิบัติงานสหกิจศึกษา

Report on Action Plan of Cooperative Education Program

(ผู้ให้ข้อมูล: นิสิตร่วมกับพนักงานที่ปรึกษา) /(Information Provider: Student and Job Supervis

1. ชื่อ-นามสกุลนิสิต/ Student Name ธาม แก้วเกษ

รหัสประจำตัวนิสิต /Student ID No. 65160172 สาขาวิชา/Major วิทยาการคอมพิวเตอร์

คณะ/Faculty วิทยาการสารสนเทศ

2. ชื่อ-นามสกุล พนักงานที่ปรึกษา /Job Supervisor Name เจตนิพิฐ พสิษฐ์จังกูร

ตำแหน่ง/Position Assistant data engineer แผนก/ฝ่าย/Department SMS

โทรศัพท์/Telephone No. 081-9922026 อีเมล/E-mail jetnipit@a-host.co.th

ขอแจ้งรายละเอียดเกี่ยวกับแผนปฏิบัติงานสหกิจศึกษา ดังนี้

งานที่ได้รับมอบหมายนิสิตสหกิจศึกษา

ตำแหน่งงานที่นิสิตปฏิบัติ (Job Position) Cloud Engineer

ลักษณะงานที่นิสิตปฏิบัติ (Job Description) DevOps, Mornitoring, ทำ Report ของแต่ละเดือน ,IT support, Health Check รายเดือน

รายละเอียดงาน DevOps: ทำระบบอัตโนมัติให้การพัฒนา-ทดสอบ-ใช้งานเร็วและเสถียร

Monitoring: เฝ้าดูสถานะระบบ (CPU, RAM, Service, Error) แบบต่อเนื่อง

Monthly Report: สรุปผลการทำงาน/ปัญหาระบบประจำเดือน IT Support: แก้ปัญหาและช่วยเหลือผู้ใช้งานระบบ Monthly Health Check: ตรวจสุขภาพระบบรายเดือนเพื่อป้องกันปัญหา

แผนปฏิบัติงานสหกิจศึกษา/Action Plan

หัวข้องาน/ Assignments	เดือนที่ 1				เดือนที่ 2				เดือนที่ 3				เดือนที่ 4			
1. ศึกษาการทำงาน และ รูปแบบการทำงาน	✓	✓														
2. ทดลองทำงานจริงที่ศึกษามา		✓	✓													
3. กำหนดหัวข้อ Project และศึกษา				✓												
4. ออกแบบโครงสร้าง Project					✓											
5. ลงมือทำ Project						✓	✓	✓	✓	✓	✓	✓				
6. Review Project							✓		✓		✓					
7. ทำงานที่ได้รับมอบหมาย					✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
8. ทำรูปเล่ม									✓	✓	✓	✓				
9.																
10.																

ลงชื่อนักศึกษา/Student's Signature ธาม แก้วเกษ

(นายธาม แก้วเกษ)

วันที่ 18 ธันวาคม 2568

ลงชื่อพนักงานที่ปรึกษา / Supervisor's Signature เจตนิพิฐ

(นายเจตนิพิฐ พสิษฐ์จังกูร)

วันที่ 18 ธันวาคม 2568

กรุณาส่งเอกสารฉบับนี้คืนงานสหกิจศึกษาภายในสัปดาห์ที่ 5 ของการปฏิบัติงานของนิสิต
โปรดมอบหมายให้นิสิตเป็นผู้ดำเนินการ โดยจัดส่งทาง E-mail ให้คณาจารย์นิเทศสหกิจศึกษา

แบบแจ้งโครงร่างรายงานการปฏิบัติงานสหกิจศึกษา (IN-SOO6)

IN-S006

งานสหกิจศึกษา คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

แบบแจ้งโครงร่างรายงานการปฏิบัติงานสหกิจศึกษา

Report on the Scheme of Work Report of Cooperative Education Program

คำชี้แจง

รายงานถือเป็นส่วนหนึ่งของการปฏิบัติงานสหกิจศึกษา มีวัตถุประสงค์เพื่อฝึกฝนทักษะการสื่อสาร(Communication Skill) ของนิสิต และจัดทำข้อมูลที่เป็นประโยชน์สำหรับสถานประกอบการ นิสิตจะต้องขอรับคำปรึกษาจากพนักงานที่ปรึกษา (Job Supervisor) เพื่อกำหนดหัวข้อรายงานที่เหมาะสม โดยคำนึงถึงความต้องการของสถานประกอบการเป็นหลัก ตัวอย่างของรายงานได้แก่ผลงานวิจัยที่นิสิตปฏิบัติ รายงานวิชาการที่น่าสนใจ การสรุปข้อมูลหรือสถิติบางประการ การวิเคราะห์และประเมินผลข้อมูล เป็นต้นทั้งนี้รายงานอาจจะจัดทำเป็นกลุ่มของนิสิตสหกิจศึกษามากกว่า 1 คนก็ได้

ในกรณีที่สถานประกอบการไม่ต้องการรายงานในหัวข้อข้างต้น นิสิตจะต้องพิจารณาเรื่องที่ตนสนใจและหยิบยกมาทำรายงาน โดยปรึกษากับพนักงานที่ปรึกษาเสียก่อน ตัวอย่างหัวข้อที่จะใช้เขียนรายงาน ได้แก่ รายงานวิชาการที่นิสิตสนใจ รายงานการปฏิบัติงานที่ได้รับมอบหมาย หรือแผนและวิธีการปฏิบัติงานที่จะทำให้บรรลุถึงวัตถุประสงค์ของการเรียนรู้ที่นิสิตวางเป้าหมายไว้จากการปฏิบัติงานสหกิจศึกษาครั้งนี้ (Learning Objectives) เมื่อกำหนดหัวข้อได้แล้ว ให้นิสิตจัดทำโครงร่างของเนื้อหารายงานพอสังเขป ตามแบบฟอร์มนี้ ทั้งนี้ให้ปรึกษากับ Job Supervisor เสียก่อนแล้วจึงส่งกลับมายังสหกิจศึกษา ภายในระยะเวลาที่กำหนด

ชื่อ-นามสกุล/Name-Surname สาม แก้วเกษ

รหัสประจำตัวนิสิต/Student ID No. 65160172 สาขาวิชา/Major วิทยาการคอมพิวเตอร์

คณะ/Faculty วิทยาการสารสนเทศ

ชื่อสถานประกอบการ/Employer Name บริษัท เอ-โฮสต์ จำกัด

ตำแหน่ง/Position Cloud Engineer แผนก/ฝ่าย/Department System Management Service

ขอแจ้งรายละเอียดเกี่ยวกับโครงร่างรายงานการปฏิบัติงานสหกิจศึกษา ดังนี้ / Work Report Scheme is detailed as the following:

ชื่อโครงงานสหกิจศึกษา (ภาษาไทย) การออกแบบและพัฒนาเว็บแอปพลิเคชัน Automate Daily Monitoring

Work Report (English) Design and Development of An Automate Daily Monitoring Web Application

รายละเอียดเนื้อหา/Detail การออกแบบระบบ Automate สำหรับการ Daily Monitoring โดยใช้ Web Application ในการแสดงผล โดยออกแบบระบบ Front-end ด้วย React และ Back-end ด้วย n8n

ลงชื่อนักศึกษา/Student's Signature

สาม แก้วเกษ

(นายสาม แก้วเกษ)

11 / 3 / 2569

ลงชื่อพนักงานที่ปรึกษา / Supervisor's Signature

(นาย ธนพนัช พลิตธีรกุล)

11 / 3 / 2569

ตำแหน่ง/Position Assistant Data Engineer

กรุณาส่งเอกสารฉบับนี้คืนงานสหกิจศึกษาภายในสัปดาห์ที่ 6-7 ของการปฏิบัติงานของนิสิต

โปรดมอบหมายให้นิสิตเป็นผู้ดำเนินการ โดยจัดส่งทาง E-mail ให้อาจารย์ที่ปรึกษาสหกิจศึกษา และต้นฉบับนำให้อาจารย์ที่ปรึกษาฯ ในวันนิเทศงานสหกิจศึกษา หรือจัดส่งทางไปรษณีย์ งานสหกิจศึกษา คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา

169 ถ.ลงหาดบางแสน ต.แสนสุข อ.เมือง จ.ชลบุรี 20131 โทรศัพท์ 038-103060-1 โทรสาร 038-393245

ประวัติย่อของผู้จัดทำ

ชื่อ – สกุล	นายธาม แก้วเกษ
วัน เดือน ปีเกิด	วันที่ 13 เดือน มิถุนายน พ.ศ. 2546
สถานที่เกิด	จังหวัดจันทบุรี
ที่อยู่ปัจจุบัน	บ้านเลขที่ 9/9 หมู่ 1 ตำบล ชากไทย อำเภอเขาคิชฌกูฏ จังหวัดจันทบุรี 22210
ประวัติการศึกษา	
พ.ศ. 2569 (กำลังศึกษา)	วิทยาศาสตรบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ มหาวิทยาลัยบูรพา
พ.ศ. 2562	มัธยมศึกษาตอนปลาย โรงเรียนเบญจมราชูทิศ จังหวัดจันทบุรี
พ.ศ. 2559	มัธยมศึกษาตอนต้น โรงเรียนเบญจมราชูทิศ จังหวัดจันทบุรี