

ระบบติดตั้งเว็บแอปพลิเคชันและเชื่อมโยงโดเมนอัตโนมัติ

AUTOMATED WEB APPLICATION DEPLOYMENT WITH DOMAIN MAPPING

นายพงศ์สวัสดิ์	ยศคันธิ	รหัสนิสิต 65021903
นายชนกชล	ลำดับ	รหัสนิสิต 65024625

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาบริกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยพะเยา
ปีการศึกษา 2568

ระบบติดตั้งเว็บแอปพลิเคชันและเชื่อมโยงโดเมนอัตโนมัติ

AUTOMATED WEB APPLICATION DEPLOYMENT WITH DOMAIN MAPPING

นายพงศ์สวัสดิ์	ยศคันธิ	รหัสนิสิต 65021903
นายชนกชล	ลำดับ	รหัสนิสิต 65024625

ปริญญาในชื่อนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาบริการคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร
มหาวิทยาลัยพะเยา
ปีการศึกษา 2568

หัวข้อโครงการ	ระบบติดตั้งเว็บแอปพลิเคชันและเชื่อมโถมโถมอัตโนมัติ	
ผู้ดำเนินโครงการ	นายพงศภัค ยศคันธิ	รหัส 65021903
	นายชนะชล ลำดับ	รหัส 65024628
อาจารย์ที่ปรึกษา	ผศ.ดร.วัฒนพงศ์ สุทธภักดี	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์	
คณะ	เทคโนโลยีสารสนเทศและการสื่อสาร	
ปีการศึกษา	2568	

บทคัดย่อ

โครงการนี้มุ่งเน้นการพัฒนาเว็บแอปพลิเคชันที่ช่วยสร้างเว็บไซต์ผ่านอินเทอร์เฟซแบบกราฟิก (GUI) ซึ่งออกแบบมาเพื่ออำนวยความสะดวกให้แก่ผู้ใช้งานทุกระดับ โดยเฉพาะผู้ที่ไม่มีความเชี่ยวชาญด้านการเขียนโปรแกรม ระบบรองรับการสร้างเว็บไซต์ในรูปแบบต่าง ๆ เช่น PHP, Node.js, Golang และ HTML Static พร้อมความสามารถในการติดตั้งและจัดการปลั๊กอินเสริม เพื่อเพิ่มฟังก์ชันการทำงานให้สอดคล้องกับความต้องการเฉพาะแอปพลิเคชันนี้ยังผสานรวมการทำงานร่วมกับ Nginx เพื่อจัดการทรัพฟิกเว็บไซต์ การตั้งค่า HTTPS ด้วยใบรับรอง SSL และการสร้างไฟล์ตั้งค่าเซิร์ฟเวอร์โดยอัตโนมัติ ช่วยให้ผู้ดูแลระบบสามารถปรับแต่งเว็บไซต์ได้อย่างรวดเร็วและมีประสิทธิภาพ นอกจากนี้ระบบได้รับการออกแบบให้ทำงานบนระบบปฏิบัติการ Ubuntu ซึ่งช่วยเพิ่มความเสถียรและเหมาะสมกับสภาพแวดล้อมของ Linux แพลตฟอร์มที่คาดหวังจากโครงการนี้คือการลดขั้นตอนและเวลาในการพัฒนาเว็บไซต์ เพิ่มประสิทธิภาพในการจัดการระบบ และเปิดโอกาสให้ผู้ใช้งานทุกคนสามารถสร้างเว็บไซต์ที่ตอบโจทย์ได้อย่างง่ายดายและรวดเร็ว โดยไม่จำเป็นต้องมีทักษะทางเทคนิคที่ซับซ้อน

Project Title	AUTOMATED WEB APPLICATION DEPLOYMENT WITH DOMAIN MAPPING	
Project Authors	Mr. Pongsapak Yoskhantho	ID 65021903
	Mr. Chanachol Lamdab	ID 65024628
Project Advisor	Asst. Prof. Wattanapong Suttpak, Ph.D.	
Department	Computer Engineering	
Faculty	School of Information and Communication Technology	
Academic Year	2025	

Abstract

This project focuses on developing a web application that facilitates website creation through a graphical user interface (GUI), designed to accommodate users of all levels, especially those without programming expertise. The system supports building various types of websites, such as PHP, Node.js, Golang, and HTML Static, with features for installing and managing additional plugins to enhance functionality based on specific requirements. The application integrates seamlessly with Nginx to handle website traffic, configure HTTPS with SSL certificates, and auto-generate server configuration files. These capabilities enable administrators to customize websites efficiently and effectively. Furthermore, the system is exclusively designed to operate on the Ubuntu operating system, ensuring stability and compatibility with Linux environments. The expected outcomes of this project include reducing the complexity and time required for website development, improving system management efficiency, and empowering users to create websites that meet their needs easily and quickly without requiring advanced technical skills.

กิตติกรรมประกาศ

ปริญญา妮พนธ์ฉบับนี้สำเร็จลุล่วงตามวัตถุประสงค์ได้ด้วยความกรุณาและความช่วยเหลือ
เป็นอย่างดียิ่งจาก ศศ.ดร.วัฒนพงศ์ สุทธกัตติ อาจารย์ที่ปรึกษาปริญญา妮พนธ์ ดร.ภูวิศสรณ์
ภูมิสรณ์และ อาจารย์อภิวัฒน์ วิทยารัชต์ ที่ได้ให้คำแนะนำ ข้อคิดเห็นต่าง ๆ อันเป็นประโยชน์
อย่างยิ่งในการทำวิจัยตลอดจนการตรวจแก้ไขปริญญา妮พนธ์ฉบับนี้จนเสร็จสมบูรณ์

ขอกราบขอบพระคุณบิดา แมรดา และครอบครัว ที่ให้การอุปการะอบรมเลี้ยงดู ตลอดจนส่งเสริม
การศึกษา และให้กำลังใจเป็นอย่างดีเสมอมา

ขอขอบคุณพี่ เพื่อน และน้องสาขาวิศวกรรมคอมพิวเตอร์ทุกคน ที่คบอยู่กับฉันได้ด้วยความห่วงใย
ให้ความช่วยเหลือ คำแนะนำ กำลังใจที่ดีและการให้เสมอมา

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	i
บทคัดย่อภาษาอังกฤษ	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญตาราง	viii
สารบัญรูป	ix

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญ	1
1.2 จุดมุ่งหมายและวัตถุประสงค์	2
1.3 สมมติฐาน	2
1.4 ขอบเขตของการศึกษา	2
1.4.1 ขอบเขตของผู้ใช้งาน	2
1.4.2 ขอบเขตของผู้ดูแลระบบ	3
1.5 ขั้นตอนการดำเนินงาน	3
1.6 เครื่องมือและอุปกรณ์ที่ใช้	4
1.6.1 เครื่องมือและอุปกรณ์ที่ใช้การพัฒนาระบบ	4
1.6.2 เครื่องมือและอุปกรณ์ที่ต้องการสำหรับการใช้งานระบบ	5
1.7 ประโยชน์ที่คาดว่าจะได้รับ	5

สารบัญ (ต่อ)

	หน้า
1.8 โครงสร้างปริญญาภินพนธ์	6
บทที่ 2 งานวิจัยหรือทฤษฎีที่เกี่ยวข้อง	
2.1 ทฤษฎีที่เกี่ยวข้อง.....	2
2.1.1 Web Server	2
2.1.2 Nginx.....	14
2.1.3 Docker.....	15
2.1.4 Kubernetes.....	18
2.1.5 SSL	20
2.2 งานวิจัยที่เกี่ยวข้อง	22
2.2.1 cPanel	22
2.2.2 Plesk.....	25
2.2.3 DirectAdmin.....	27
2.2.4 Webmin	30
2.2.5 ISPConfig	33
บทที่ 3 วิธีการดำเนินงาน	
3.1 การวิเคราะห์ความต้องการของระบบ	39
3.1.1 การกำหนดพื้นที่ในการทำงานหลักของระบบ.....	39

สารบัญ (ต่อ)

	หน้า
3.2 การออกแบบสถาปัตยกรรมระบบ.....	41
3.2.1 แผนภาพการออกแบบพังก์ชันการในด้านของผู้ใช้งานและด้านของผู้ดูแลระบบ	41
3.2.2 แผนภาพรวมของระบบ.	45
3.3 การออกแบบแผนภาพระบบ	47
3.3.1 แผนภาพกิจกรรม (Active Diagram)	47
3.3.2 แผนภาพพยุลสเคส (Use Case Diagram)	53
3.4 การออกแบบฐานข้อมูล	58
3.4.1 แผนภาพแสดงความสัมพันธ์ระหว่างฐานข้อมูล	58
3.5 ส่วนออกแบบการติดต่อประสานงานผู้ใช้งาน	65
3.5.1 หน้าเข้าสู่ระบบและลงทะเบียน.....	65
3.5.2 หน้าแสดงเว็บไซต์ของผู้ใช้ทั้งหมด	67
3.5.3 หน้าสร้างเว็บไซต์.....	67
3.5.4 หน้าติดตั้งส่วนเสริมเสริม.....	68
3.5.5 หน้าตั้งค่าเว็บไซต์.....	68
3.5.6 หน้าวิเคราะห์ข้อมูลเว็บไซต์	69
3.5.7 หน้าของผู้ดูแลระบบ	71
3.5.8 หน้าของการตั้งค่าข้อมูลส่วนตัวของผู้ใช้งาน	71
3.6 บทสรุป	72

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดสอบ

4.1 ผลการพัฒนาระบบ	73
4.1.1 ส่วนของผู้ใช้งานระบบ	73
4.1.2 พั้งก์ชันการทำงานหลัก	74
4.2 ผลการทดสอบระบบ	79
ผลการทดสอบพั้งก์ชันการทำงานในแต่ละส่วน	79

บทที่ 5 บทสรุป

5.1 สรุปผลการพัฒนา	92
5.2 อภิปรายผล	93
5.3 ข้อเสนอแนะ	93
5.4 สรุป	94
เอกสารอ้างอิง	xii
ภาคผนวก	xv
ภาคผนวก ก	xvi
ประวัติผู้เขียน	xxi

สารบัญตาราง

ตารางที่	หน้า
1.1 ระยะเวลาที่ใช้ในการศึกษาและการจัดทำระบบจะเริ่มต้นการดำเนินงานตั้งแต่เดือนตุลาคมปี พ.ศ. 2567 ถึง เดือนตุลาคม พ.ศ. 2568	3
2.1 ตารางเบรียบเทียบความสามารถของงานที่เกี่ยวข้องและโปรเจกต์ ADDMS	36
2.2 ตารางเบรียบเทียบภาษาที่รองรับของงานที่เกี่ยวข้องและโปรเจกต์ ADDMS	38
3.1 ตารางผู้ใช้ (USERS)	60
3.2 ตารางเว็บไซต์ (WEBSITES)	60
3.3 ตารางชื่อโดเมน (DOMAIN)	61
3.4 ตารางคำสั่งเริ่มต้นเซิร์ฟเวอร์ (START SERVERS)	62
3.5 ตารางการติดตั้งปลั๊กอินและคำสั่ง (INSTALL PLUGIN MODELS)	63
3.6 ตารางสถานะเซิร์ฟเวอร์ (SERVER STATUSES)	63
4.1 ตารางการทดสอบการทำงานของฟังก์ชัน	79

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.1 แผนผังสถาปัตยกรรมของเว็บเซิร์ฟเวอร์	8
2.2 ตัวอย่างการเขียนภาษา PHP	9
2.3 ตัวอย่างการเขียนภาษา JAVA SERVLET	10
2.4 ตัวอย่างการเขียนภาษา NODE.JS.....	11
2.5 ตัวอย่างการเขียนภาษา Go	12
2.6 ตัวอย่างการเขียนภาษา ASP.NET.....	13
2.7 ตัวอย่างคำสั่ง NGINX	17
2.8 ตัวอย่างคำสั่ง NGINX	17
2.9 ตัวอย่างคำสั่ง NGINX	18
2.10 ตัวอย่างคำสั่ง TOMCAT.....	18
2.11 องค์ประกอบทั้งหมดที่เกี่ยวข้องกับการแสดงผล SSL CERTIFICATE ในเบราว์เซอร์ทั่วไป	22
2.12 cPANEL CONTROL PANEL	23
2.13 PLESK CONTROL PANEL	25
2.14 DIRECTADMIN CONTROL PANEL	28
2.15 WEBMIN CONTROL PANEL	30
2.16 ISPCONFIG CONTROL PANEL	33
3.1 แผนภาพการออกแบบฟังก์ชันการทำงานหลักในด้านของผู้ใช้งาน	43
3.2 แผนภาพการออกแบบฟังก์ชันการทำงานของผู้ดูแลระบบ	44
3.3 แผนภาพรวมระบบ.....	46
3.4 แผนภาพของกิจกรรมของหน้าเข้าสู่ระบบ	49
3.5 แผนกิจกรรมของหน้าสร้างเว็บไซต์ใหม่	50
3.6 แผนภาพกิจกรรมของหน้าแสดงเว็บไซต์ที่มีอยู่ของผู้ใช้	51
3.7 แผนกิจกรรมของหน้าแสดงการตั้งค่าเว็บไซต์	52

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.8 แผนภาพযুสเคสการลงชื่อเข้าใช้และลงทะเบียนใหม่	54
3.9 แผนภาพযুสเคสสร้างเว็บไซต์	55
3.10 แผนภาพযুสเคสแสดงข้อมูลเริบไซต์ทั้งหมดของผู้ใช้รวมถึงการเข้าสู่ระบบ	56
3.11 แผนภาพযুสเคสของการตั้งค่าเว็บไซต์	57
3.12 แผนภาพযুสเคสของผู้ดูแลระบบ	58
3.13 แผนภาพแสดงความล้มเหลวระหว่างข้อมูล	59
3.14 หน้าของการเข้าสู่ระบบ	66
3.15 หน้าของการเข้าสู่ระบบผู้ใช้งานใหม่	66
3.16 หน้าแสดงเว็บไซต์ของผู้ใช้ทั้งหมด	67
3.17 หน้าสร้างเว็บไซต์	68
3.18 หน้าติดตั้งส่วนเสริมเสริม	68
3.19 หน้าตั้งค่าเว็บไซต์	69
3.20 หน้าวิเคราะห์ข้อมูลเว็บไซต์	70
3.21 หน้าของผู้ดูแลระบบ	71
3.22 หน้าของการตั้งค่าข้อมูลส่วนตัวของผู้ใช้งาน	72
4.1 หน้า LANDINGPAGE หรือโฮมเพจ	82
4.2 หน้าสำหรับการลงชื่อเข้าใช้งาน	83
4.3 หน้าสำหรับระบบการลงทะเบียนผู้ใช้ใหม่	84
4.4 หน้า DASHBOARD	85
4.5 หน้าการติดตั้งเว็บไซต์ใหม่	86
4.6 หน้าการติดตั้งส่วนเสริมและแก้ไขไฟล์	87
4.7 หน้าFORGOT PASSWORD	88
4.8 หน้าແຜງគົບຄຸມຜູ້ດູແລຣະບປ	89

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.9 หน้าการจัดการผู้ใช้งาน	90
4.10 หน้าการจัดการเว็บไซต์	91

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบัน การบริหารจัดการเว็บไซต์และเซิร์ฟเวอร์นับเป็นภารกิจที่มีความสำคัญและมีความซับซ้อนสูง โดยเฉพาะบนระบบปฏิบัติการ Linux ที่ผู้ดูแลระบบส่วนใหญ่มากใช้การทำงานผ่าน Command Line Interface (CLI) หรือ Terminal ซึ่งมีข้อดีด้านความยืดหยุ่นและความเร็วในการควบคุมการทำงาน แต่ในขณะเดียวกันก็มีข้อจำกัดที่สำคัญ เช่น การต้องจดจำคำสั่งจำนวนมาก การเรียนรู้ที่ใช้เวลา lange และความเสี่ยงจากการพิมพ์คำสั่งผิดพลาดที่อาจส่งผลกระทบต่อความปลอดภัยและความต่อเนื่องของระบบ ทำให้วิธีการนี้ไม่เหมาะสมกับผู้ใช้งานที่ขาดความรู้เชิงเทคนิคอย่างเพียงพอ

เพื่อแก้ไขข้อจำกัดดังกล่าว จึงได้มีการพัฒนาซอฟต์แวร์จัดการระบบในรูปแบบ Graphical User Interface (GUI) เช่น Plesk หรือ cPanel ที่ช่วยอำนวยความสะดวกในการทำงานด้วยอินเทอร์เฟซที่เป็นมิตรและสามารถเข้าถึงผ่านเว็บเบราว์เซอร์ได้ง่าย เครื่องมือเหล่านี้ช่วยลดความยุ่งยากในการดูแลระบบและเหมาะสมสำหรับผู้ใช้ที่ไม่มีทักษะเชิงเทคนิค แต่ก็ยังมีข้อจำกัดในด้านค่าใช้จ่ายที่สูง ความซับซ้อนของฟีเจอร์ที่เกินความจำเป็น และข้อจำกัดในการปรับแต่งให้ตรงตามความต้องการเฉพาะของผู้ใช้ ส่งผลให้การเลือกใช้งานระหว่าง CLI และ GUI ยังคงไม่สามารถตอบโจทย์ได้อย่างครบถ้วน โดยเฉพาะกับธุรกิจขนาดเล็กและกลางที่มีทรัพยากรจำกัด

ดังนั้น จึงเกิดความจำเป็นในการพัฒนาระบบจัดการเว็บไซต์ที่สามารถสนับสนุนผู้ใช้ทุกระดับ ทักษะและมีค่าใช้จ่ายที่เหมาะสม อีกทั้งยังต้องมีความสามารถในการปรับแต่งและขยายระบบ รวมถึงรองรับมาตรฐานการด้านความปลอดภัยที่ทันสมัย ระบบในลักษณะนี้จะช่วยให้การจัดการเว็บไซต์ครอบคลุมทั้งด้านการบริหารโดยเมน เว็บเซิร์ฟเวอร์ ฐานข้อมูล การติดตั้งซอฟต์แวร์ ตลอดจนการดูแลความปลอดภัยของระบบ ซึ่งจะเป็นประโยชน์โดยตรงต่อผู้ดูแลระบบ นักพัฒนา และองค์กรธุรกิจที่ต้องการเพิ่มประสิทธิภาพ ลดความเสี่ยง และสร้างความยั่งยืนในการดำเนินงานในระยะยาว

1.2 จุดมุ่งหมายและวัตถุประสงค์

- 1) เพื่อพัฒนาเว็บคอนโทรลพาเนลในรูปแบบเว็บแอปพลิเคชัน สำหรับสร้างสิ่งแวดล้อมในการพัฒนาเว็บไซต์แบบมีการประมวลผลผั่งเมฆข่ายและฐานข้อมูล
- 2) เพื่อลดความซับซ้อนในการสร้างสิ่งแวดล้อมในการพัฒนาเว็บไซต์แบบมีการประมวลผลผั่งเมฆและฐานข้อมูล

1.3 สมมติฐาน

การบริหารจัดการเว็บไซต์และเซิร์ฟเวอร์บนระบบปฏิบัติการ Linux ผ่าน Terminal มีความยืดหยุ่นสูง และสามารถควบคุมระบบได้อย่างละเอียด อย่างไรก็ตาม การใช้งาน Terminal มีความซับซ้อนและต้องการความชำนาญเฉพาะทาง ซึ่งอาจเป็นอุปสรรคสำหรับผู้ใช้งานที่ไม่มีพื้นฐานด้านเทคโนโลยี ขณะที่โปรแกรมจัดการระบบแบบกราฟิก เช่น Plesk หรือ cPanel แม้จะมีความสะดวกสบายในการใช้งาน แต่ก็มักมาพร้อมกับค่าใช้จ่ายสูง ฟีเจอร์ที่ซับซ้อนเกินความจำเป็น และข้อจำกัดในการปรับแต่งที่อาจไม่ตอบโจทย์ความต้องการของผู้ใช้งานบางกลุ่ม

ดังนั้น การพัฒนาระบบจัดการเว็บไซต์แบบ Open Source ที่สามารถสามารถข้อดีของ Terminal และ GUI เข้าด้วยกัน จึงมีความสำคัญอย่างยิ่ง ระบบดังกล่าวควรลดความซับซ้อนในการจัดการเพิ่มประสิทธิภาพการทำงานและลดค่าใช้จ่าย พร้อมทั้งรองรับการใช้งานสำหรับผู้ใช้ทุกระดับทักษะ โดยการออกแบบส่วนติดต่อผู้ใช้ให้เข้าใจง่ายและเป็นมิตร การจัดการงานต่าง ๆ เช่น การตั้งค่าเว็บเซิร์ฟเวอร์ การจัดการฐานข้อมูลและการติดต่อสื่อสาร ควรสามารถทำได้อย่างครอบคลุมและยืดหยุ่น เพื่อให้ตอบสนองความต้องการที่หลากหลายของธุรกิจขนาดเล็กถึงกลาง

1.4 ขอบเขตของการศึกษา

1.4.1 ขอบเขตของผู้ใช้งาน

- 1) สามารถลงท่าเบียนและเข้าสู่ระบบได้
- 2) สามารถตรวจสอบสถานะของเว็บแอปพลิเคชันได้
- 3) สามารถสร้างเว็บแอปพลิเคชันได้อย่างง่ายดาย
- 4) สามารถเพิ่มส่วนเซริมและโมดูลเสริมได้ตามเฟรมเวิร์กที่ต้องการ
- 5) สามารถเลือกภาษาหรือเฟรมเวิร์กที่ต้องการได้
- 6) สามารถจัดการหลายเว็บไซต์ในบัญชีเดียวได้

- 7) สามารถตั้งค่าข้อมูลโดเมนและเชื่อมต่อเว็บไซต์กับ HTTPS ผ่าน GUI ได้
 - 8) สามารถอัปโหลด ดาวน์โหลดได้และไฟล์ปีบอัดสำหรับรับเว็บไซต์ที่สร้างขึ้นได้

1.4.2 ขอบเขตของผู้ดูแลระบบ

- 1) สามารถดูพื้นที่ที่ใช้งานของแต่ละผู้ใช้งานได้
 - 2) สามารถจำกัดพื้นที่ที่ใช้งานได้
 - 3) สามารถจัดการและกำหนดสิทธิ์การใช้งานของผู้ใช้งานได้
 - 4) สามารถจัดการเรื่องไชร์ตของผู้ใช้งานได้

1.5 ขั้นตอนการดำเนินงาน

ระยะเวลาที่ใช้ในการศึกษาและการจัดทำระบบจะเริ่มต้นการดำเนินงานตั้งแต่เดือนตุลาคมปี พ.ศ. 2567 ถึง เดือนตุลาคม พ.ศ. 2568 รวมเป็นระยะเวลา 13 เดือน แผนการดำเนินงานแสดงดังตารางที่ 1.1

ตารางที่ 1.1 ระยะเวลาที่ใช้ในการศึกษาและการจัดทำระบบจะเริ่มต้นการดำเนินงานตั้งแต่เดือน
ตุลาคม ปี พ.ศ. 2567 ถึง เดือนตุลาคม พ.ศ. 2568

ตารางที่ 1.1 ระยะเวลาที่ใช้ในการศึกษาและการจัดทำระบบฐานข้อมูลตั้งแต่เดือน

ตุลาคม ปี พ.ศ. 2567 ถึง เดือนตุลาคม พ.ศ. 2568 (ต่อ)

ระยะเวลา การดำเนินงาน	ปีการศึกษา 2567						ปีการศึกษา 2568					
	๓.๓	๔.๔	๕.๕	๖.๖	๗.๗	๘.๘	๙.๙	๑๐.๑๐	๑๑.๑๑	๑๒.๑๒	๑๓.๑๓	๑๔.๑๔
4.ดำเนินการ สร้าง โปรแกรม			◀								▶	
5.ทดสอบ โปรแกรมและ แก้ไข ข้อผิดพลาด			◀								▶	
6.ประเมินผล ข้อเสนอแนะ และสรุปผล			◀								▶	
7.จัดทำเอกสาร			◀								▶	
8.นำเสนอ											↔	
9.ส่งมอบงาน											↔	

1.6 เครื่องมือและอุปกรณ์ที่ใช้

1.6.1 เครื่องมือและอุปกรณ์ที่ใช้ในการพัฒนาระบบ

1.6.1.1 ฮาร์ดแวร์

1) คอมพิวเตอร์ส่วนบุคคลสำหรับการพัฒนาส่วนของหน้าบ้าน

- หน่วยประมวลผล AMD Ryzen 7 4800H
- หน่วยความจำสำรอง 16 กิกะไบต์ (GB)
- การ์ดจอ GPU NVIDIA GeForce RTX 3050
- พื้นที่จัดเก็บ 512 กิกะไบต์ (GB) SSD PCIe M.2

2) คอมพิวเตอร์ส่วนบุคคลสำหรับการพัฒนาส่วนของหลังบ้าน

- หน่วยประมวลผล Ryzen 12000H
- หน่วยความจำสำรอง RAM 16 กิกะไบต์ (GB)

- gpu ต้อง Nvidia GTX 1050 TI
 - พื้นที่จัดเก็บ SSD 500 กิกะไบต์ (GB) HDD 1000 กิกะไบต์ (GB)

1.6.1.2 ซอฟต์แวร์

- 1) ระบบปฏิบัติการ Ubuntu Linux 22.04.3 LTS
 - 2) เครื่องมือการพัฒนาซอฟต์แวร์ Visual Studio Code, HeidiSQL
 - 3) ภาษาที่ใช้ในการเขียนซอฟต์แวร์ Typescript, Golang
 - 4) เฟรมเวิร์ก Go Fiber, Next.js, GORM
 - 5) ฐานข้อมูล MariaDB
 - 6) เว็บเซิร์ฟเวอร์ Nginx

1.6.2 เครื่องมือและอุปกรณ์ที่ต้องการสำหรับการใช้งานระบบ

1.6.2.1 ສ້າງໝັດແວ່ງ

- 1) หน่วยประมวลผลกลาง (CPU) จำนวนแกนประมวลผล (Core) ไม่น้อยกว่า 4 แกน
 - 2) หน่วยความจำหลัก (RAM) ความจุไม่น้อยกว่า 8 กิกะไบต์ (GB)
 - 3) หน่วยประมวลผลกราฟิก (GPU) หน่วยความจำวิศีโอลามิ่งอยกว่า 2 กิกะไบต์ (GB)
 - 4) หน่วยจัดเก็บข้อมูล (Storage) ฮาร์ดดิสก์ไดร์ฟ (HDD) หรือโซลิดสเตตไดร์ฟ (SSD) ที่มีความจุไม่น้อยกว่า 50 กิกะไบต์ (GB)

1.6.2.2 ซอฟต์แวร์

- 1) ระบบปฏิบัติการ Windows 7 ขึ้นไป

1.7 ประโยชน์ที่คาดว่าจะได้รับ

- 1) เพิ่มความเร็วและประสิทธิภาพในการพัฒนาเว็บคอนโทรลพาเนลช่วยลดเวลาและความยุ่งยากในการตั้งค่าสภาพแวดล้อม ทำให้นักพัฒนาสามารถเริ่มงานได้ทันทีและลดข้อผิดพลาดจากการตั้งค่าที่ซับซ้อน
 - 2) ลดความซับซ้อนสำหรับนักพัฒนาระบบที่ออกแบบมาให้ใช้งานง่ายช่วยให้นักพัฒนาไม่ต้องมีความเชี่ยวชาญสูงในการตั้งค่าเซิร์ฟเวอร์หรือฐานข้อมูล ทำให้สามารถโฟกัสที่การพัฒนาเว็บไซต์โดยตรง

1.8 โครงสร้างปริญญา尼พนธ์

โครงงานฉบับนี้แบ่งออกเป็น 5 บท แต่ละบทประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงาน สมมติฐานของโครงงาน ขั้นตอนการศึกษา เครื่องมือและอุปกรณ์ที่ใช้ในการพัฒนาโครงการ ADDMS ประโยชน์ที่คาดว่าจะได้รับ รวมถึงโครงสร้างของการศึกษา

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง กล่าวถึงทฤษฎีและหลักการพื้นฐานของการพัฒนา ระบบจัดการเว็บไซต์อัตโนมัติ (Automated Web Application Deployment with Domain Mapping) เช่น การจัดการเว็บเซิร์ฟเวอร์ Nginx การทำงานของฐานข้อมูล การเชื่อมต่อ API การออกแบบเว็บแอปพลิเคชันและงานวิจัยที่เกี่ยวข้องกับเครื่องมือช่วยสร้างโครงสร้างพื้นฐานของเว็บไซต์ รวมถึงแนวทางการออกแบบระบบเพื่อความปลอดภัยและความเสถียร

บทที่ 3 วิธีการดำเนินการวิจัย กล่าวถึงการวิเคราะห์และออกแบบระบบ ADDMS โดยใช้การเขียนแผนภาพแสดงการทำงานของผู้ใช้ระบบ (Use Case Diagram) แผนภาพแสดงกิจกรรมการทำงานของระบบ (Activity Diagram) และแบบจำลองความสัมพันธ์ระหว่างข้อมูล พร้อมแผนภาพ ER-Diagram (Entity-Relationship Diagram) เพื่อแสดงความสัมพันธ์ของข้อมูลในระบบ รวมถึงขั้นตอนการพัฒนา พังก์ชันหลักและการทดสอบระบบ

บทที่ 4 ผลการดำเนินงานและการทดสอบระบบ กล่าวถึงการดำเนินงานและขั้นตอนการพัฒนาระบบ ADDMS (Automated Web Application Deployment with Domain Mapping) รวมถึงการทดสอบประสิทธิภาพของระบบในด้านต่าง ๆ เช่น การสร้างและจัดการเว็บไซต์ การเชื่อมต่อโดเมน การทำงานของเว็บเซิร์ฟเวอร์ (Nginx) ความถูกต้องของการจัดการฐานข้อมูล ความปลอดภัยของระบบ และการประเมินประสิทธิภาพในการใช้งานจริง

บทที่ 5 สรุปผลการดำเนินงานและแนวทางในการพัฒนาต่อไปในอนาคต กล่าวถึงสรุปผลการดำเนินงานของระบบ ADDMS ที่ได้พัฒนา รวมถึงข้อจำกัดที่พบในกระบวนการพัฒนาและแนวทางในการพัฒนาระบบท่อไปในอนาคต เช่น การเพิ่มการรองรับเฟรมเวิร์กใหม่ ๆ (Golang, Node.js, PHP เป็นต้น) การปรับปรุงประสิทธิภาพและความปลอดภัย การเพิ่มพังก์ชันการทำงานอัตโนมัติ และการต่อยอดเพื่อรองรับการใช้งานในระดับองค์กรขนาดใหญ่

บทที่ 2

งานวิจัยหรือทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 Web Server

เว็บเซิร์ฟเวอร์ (Web Server) เป็นซอฟต์แวร์และฮาร์ดแวร์ที่ทำหน้าที่รับคำขอจากผู้ใช้ผ่านทางโปรโตคอล HTTP หรือ HTTPS และส่งข้อมูลที่ร้องขอ เช่น หน้าเว็บ รูปภาพ หรือไฟล์อื่น ๆ กลับไปยังผู้ใช้ โดยทั่วไปแล้ว เว็บเซิร์ฟเวอร์จะจัดเก็บไฟล์ต่าง ๆ ที่เป็นส่วนประกอบของเว็บไซต์ เช่น HTML CSS JavaScript และสื่ออื่น ๆ ที่จำเป็นสำหรับการแสดงผลหน้าเว็บ [1] [2]

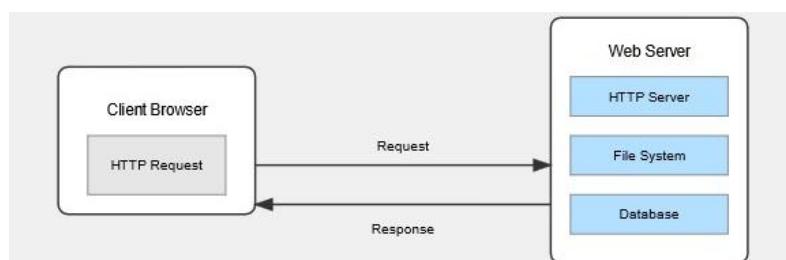
หน้าที่ของเว็บเซิร์ฟเวอร์

- 1) รับคำขอจากผู้ใช้ เมื่อผู้ใช้ต้องการเข้าถึงเว็บไซต์ จะมีการส่งคำขอ (request) ผ่านโปรโตคอล HTTP หรือ HTTPS ไปยังเว็บเซิร์ฟเวอร์
- 2) ประมวลผลคำขอ เว็บเซิร์ฟเวอร์จะทำการประมวลผลคำขอโดยการค้นหาไฟล์หรือข้อมูลที่เกี่ยวข้อง ซึ่งอาจรวมถึง HTML CSS รูปภาพ และ JavaScript
- 3) ส่งข้อมูลกลับไปยังผู้ใช้ หลังจากประมวลผลคำขอเสร็จสิ้น เว็บเซิร์ฟเวอร์จะส่งข้อมูลหรือหน้าเว็บที่ร้องขอกลับไปยังผู้ใช้

ตัวอย่างของเว็บเซิร์ฟเวอร์

- 1) Apache HTTP Server เป็นหนึ่งในเว็บเซิร์ฟเวอร์ที่ได้รับความนิยมมากที่สุดในโลก มีจุดเด่นด้านความเสถียรและความยืดหยุ่น โดยสามารถปรับแต่งการทำงานได้ผ่านระบบโมดูล เช่น โมดูลสำหรับการจัดการไดเรกทอรี การรองรับหลายโปรโตคอลและการประมวลผลด้วยภาษาสคริปต์ต่าง ๆ เช่น PHP และ Perl [3] [4]
- 2) Nginx มีชื่อเสียงในด้านการจัดการคำขอจำนวนมากพร้อมกัน (C10K problem) โดยใช้ทรัพยากรต่ำ นอกจากรับคำขอที่เป็น Reverse Proxy และ Load Balancer ได้อย่างมีประสิทธิภาพ จึงเหมาะสมสำหรับเว็บไซต์ที่มีปริมาณทราฟพิกสูงและต้องการความเร็วในการโหลดข้อมูล [3] [4]

- 3) Microsoft IIS เป็นผลิตภัณฑ์ที่พัฒนาโดย Microsoft ออกแบบมาสำหรับสภาพแวดล้อมที่ใช้เทคโนโลยีของ Microsoft เช่น ASP.NET และ .NET Core โดยมีจุดเด่นในด้านความเข้ากันได้กับระบบปฏิบัติการ Windows และการบริหารจัดการผ่าน GUI ที่ใช้งานง่าย [3]
- 4) LiteSpeed เป็นเว็บเซิร์ฟเวอร์เชิงพาณิชย์ที่มีจุดเด่นด้านความเร็วและการประยัดทรัพยากรโดยสามารถจัดการทรัพยากรได้ดีกว่า Apache ในหลายกรณีเวอร์ชันโอลูเพนซอร์สอย่าง OpenLiteSpeed มีประสิทธิภาพใกล้เคียงกันและเป็นทางเลือกสำหรับองค์กรที่ต้องการลดค่าใช้จ่าย [3] [4]
- 5) Caddy เป็นเว็บเซิร์ฟเวอร์ที่เน้นการใช้งานง่าย โดยติดตั้งและกำหนดค่า HTTPS ได้โดยอัตโนมัติรองรับการพัฒนาแอปพลิเคชันด้วยภาษา Go และมีระบบจัดการโครงสร้างที่เหมาะสมสำหรับโครงการที่ต้องการความปลอดภัยเป็นพิเศษ [4]
- 6) OpenLiteSpeed เป็นเวอร์ชันโอลูเพนซอร์สของ LiteSpeed ที่ออกแบบมาให้รองรับการทำงานกับหลายภาษาโปรแกรม มีประสิทธิภาพสูง และเหมาะสมสำหรับนักพัฒนาที่ต้องการใช้งานฟีเจอร์ LiteSpeed โดยไม่ต้องใช้จ่าย [4]
- 7) Lighttpd เป็นเว็บเซิร์ฟเวอร์ที่เน้นความเร็วและการประยัดทรัพยากร จึงเหมาะสมสำหรับแอปพลิเคชันที่ต้องการความเบาและการตอบสนองที่รวดเร็ว เช่น IoT และแอปพลิเคชันในระบบผังตัว [4]
- 8) Apache Tomcat เป็นเซิร์ฟเวอร์ที่พัฒนาขึ้นเพื่อรับแอปพลิเคชันที่ใช้ Java โดยเฉพาะ เช่น JSP และ Servlet แม้จะไม่ใช่เว็บเซิร์ฟเวอร์แบบดั้งเดิม แต่มีความเหมาะสมสำหรับโครงสร้างที่เน้นเทคโนโลยี Java [4]
- 9) Node.js เป็นรันไทม์สำหรับ JavaScript ที่สามารถทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ได้ด้วยการใช้โมดูล http ของตัวเอง แม้จะไม่ได้ออกแบบมาเป็นเว็บเซิร์ฟเวอร์โดยตรง แต่ Node.js มีความยืดหยุ่นสูงและเหมาะสมสำหรับแอปพลิเคชันที่ต้องการประสิทธิภาพแบบเรียลไทม์ เช่น ระบบแชท หรือการสตรีมข้อมูล [4]



รูปที่ 2.1 แผนผังสถาปัตยกรรมของเว็บเซิร์ฟเวอร์

2.1.1.1 Programming Language

Programming Language คือ ชุดคำสั่งที่ใช้ในการเขียนโปรแกรมเพื่อให้คอมพิวเตอร์ทำงาน ตามที่ต้องการ โดยมีโครงสร้างและกฎเกณฑ์ที่ชัดเจน ช่วยให้นักพัฒนาสามารถสร้างโปรแกรมที่ตอบโจทย์ด้านการคำนวณ การประมวลผล และการสื่อสารข้อมูลในระบบคอมพิวเตอร์

■ PHP

PHP (Hypertext Preprocessor) [5] [6] [7] เป็นภาษาโปรแกรมประเภท Server-Side Scripting ที่ออกแบบมาเฉพาะสำหรับการพัฒนาเว็บแอปพลิเคชัน โดย PHP จะทำงานบนผู้เซิร์ฟเวอร์ และส่งผลลัพธ์เป็น HTML หรือข้อมูลในรูปแบบอื่นไปยังเบราว์เซอร์ของผู้ใช้งาน

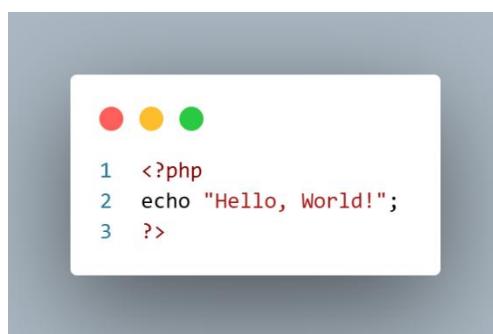
ข้อดีของ PHP

- ง่ายต่อการเรียนรู้และใช้งาน โครงสร้างโค้ดและคำสั่งเข้าใจง่าย หมายความว่าผู้เริ่มต้นสามารถเรียนรู้ได้เร็ว
- เป็นโอเพ่นซอร์ส สามารถใช้งานได้ฟรี พร้อมทั้งมีชุมชนขนาดใหญ่ที่ช่วยสนับสนุนการพัฒนา
- รองรับการทำงานร่วมกับฐานข้อมูลหลากหลาย เช่น MySQL, PostgreSQL

ข้อจำกัดของ PHP

- ข้อกังวลเรื่องความปลอดภัย หากเขียนโค้ดหรือกำหนดค่าที่ไม่เหมาะสม อาจทำให้เกิดช่องโหว่ทางความปลอดภัย
- ประสิทธิภาพในプロジェクトขนาดใหญ่ อาจไม่เหมาะสมสำหรับแอปพลิเคชันที่ต้องการความเสถียรและประสิทธิภาพสูง

ตัวอย่างการใช้งาน



รูปที่ 2.2 ตัวอย่างการเขียนภาษา PHP

■ Java Servlet

Java Servlet [8] [9] เป็นเทคโนโลยีของภาษา Java ที่ช่วยในการพัฒนาเว็บแอปพลิเคชันแบบโคนามิก ทำงานในผู้เซิร์ฟเวอร์เพื่อจัดการคำขอและ การตอบกลับในรูปแบบ HTTP

ข้อดีของ Java Servlet

- 1) การขยายตัวและรองรับการทำงานระดับสูง เหมาะสมสำหรับโปรเจกต์ขนาดใหญ่หรือระบบองค์กร
- 2) ความปลอดภัยสูง ทำงานภายใต้แพลตฟอร์ม Java EE ซึ่งมีมาตรฐานความปลอดภัยที่เข้มงวด

ข้อจำกัดของ Java Servlet

- 1) ความซับซ้อนในขั้นตอนการตั้งค่า ต้องใช้เวลาในการเรียนรู้และติดตั้ง
 - 2) ทรัพยากระบบที่ต้องใช้สูงกว่า เมื่อเทียบกับเทคโนโลยีเว็บเซิร์ฟเวอร์อื่น
- ตัวอย่างการใช้งาน



```

1 import java.io.*;
2 import javax.servlet.*;
3 import javax.servlet.http.*;
4
5 public class HelloWorldServlet extends HttpServlet {
6     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
7         response.setContentType("text/html");
8         PrintWriter out = response.getWriter();
9         out.println("<h1>Hello, World!</h1>");
10    }
11 }

```

รูปที่ 2.3ตัวอย่างการเขียนภาษา Java Servlet

■ Node.js

Node.js [10] [11] เป็นแพลตฟอร์มที่ใช้รัน JavaScript ผู้เซิร์ฟเวอร์ ช่วยให้สามารถพัฒนาแอปพลิเคชันที่รองรับคำขอพร้อมกันได้อย่างรวดเร็ว

ข้อดีของ Node.js

- 1) รองรับการทำงานแบบอะซิงโคโรนัส (Asynchronous) เพิ่มประสิทธิภาพในการจัดการคำขอหลายคำขอพร้อมกัน
- 2) ใช้ JavaScript ทั้งระบบ ลดความยุ่งยากสำหรับนักพัฒนาที่มีความชำนาญใน JavaScript

ข้อจำกัดของ Node.js

- 1) ไม่เหมาะสมกับงานที่เน้นการใช้ CPU สูง เช่น การประมวลผลที่ซับซ้อน
 - 2) การจัดการ Callback Functions อาจเกิดปัญหา Callback Hell หากออกแบบได้ไม่ดี
- ตัวอย่างการใช้งาน**



```

 1  const http = require('http');
 2
 3  http.createServer((req, res) => {
 4      res.writeHead(200, {'Content-Type': 'text/html'});
 5      res.end('<h1>Hello, World!</h1>');
 6  }).listen(3000);

```

รูปที่ 2.4 ตัวอย่างการเขียนภาษา Node.js

■ Go

Go (Golang) [12] [13] เป็นภาษาโปรแกรมที่พัฒนาโดย Google ซึ่งออกแบบมาเพื่อความรวดเร็วและเรียบง่าย รองรับงานระบบเครือข่ายและการประมวลผลแบบขนาน (Concurrency)

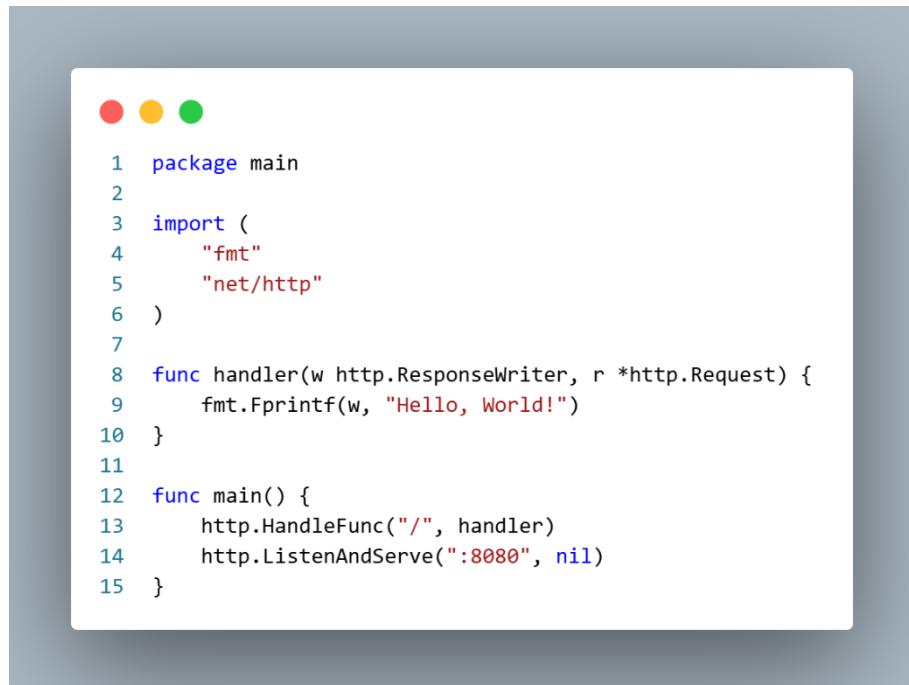
ข้อดีของ Go

- 1) ประสิทธิภาพสูง รันได้รวดเร็ว รองรับงานที่ใช้ทรัพยากร CPU มาก
- 2) การเขียนโปรแกรมแบบ Concurrency ง่ายดายด้วย Goroutines

ข้อจำกัดของ Go

- 1) ชุมชนและทรัพยากรยังมีจำกัด เมื่อเทียบกับภาษาโปรแกรมอื่น
- 2) ขาดฟีเจอร์ที่มีความยืดหยุ่นสูง เช่น การเขียนโปรแกรมเชิงวัตถุชั้นสูง

ตัวอย่างการใช้งาน



```

1 package main
2
3 import (
4     "fmt"
5     "net/http"
6 )
7
8 func handler(w http.ResponseWriter, r *http.Request) {
9     fmt.Fprintf(w, "Hello, World!")
10 }
11
12 func main() {
13     http.HandleFunc("/", handler)
14     http.ListenAndServe(":8080", nil)
15 }

```

รูปที่ 2.5 ตัวอย่างการเขียนภาษา Go

■ ASP.NET

ASP.NET [14] เป็นเฟรมเวิร์กของ Microsoft สำหรับพัฒนาเว็บแอปพลิเคชัน โดยทำงานร่วมกับภาษาโปรแกรม เช่น C# และ VB.NET

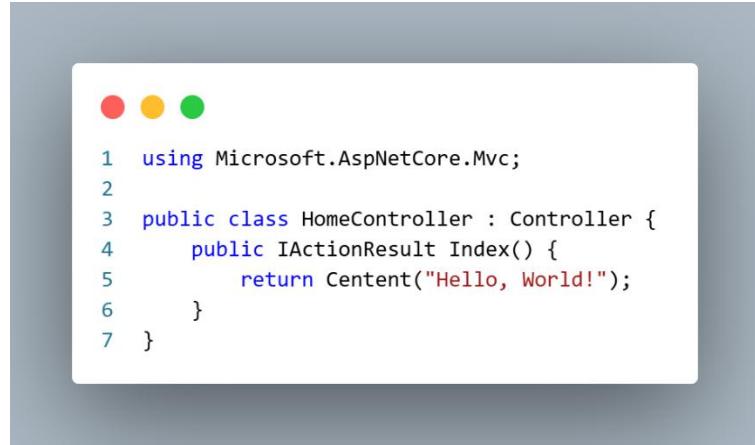
ข้อดีของ ASP.NET

- 1) รองรับสถาปัตยกรรม MVC (Model–View–Controller) ช่วยให้โครงสร้างโค้ดชัดเจนและดูแลรักษาง่าย
- 2) การพسانรวมกับผลิตภัณฑ์ Microsoft อื่น ๆ เช่น SQL Server, Azure

ข้อจำกัดของ ASP.NET

- 1) การพึ่งพา Windows Server อาจไม่เหมาะสมสำหรับองค์กรที่ใช้ระบบปฏิบัติการอื่น
- 2) ค่าใช้จ่าย การใช้งานบางส่วนอาจมีค่าใช้จ่าย เช่น ลิขสิทธิ์ Windows Server

ตัวอย่างการใช้งาน



รูปที่ 2.6 ตัวอย่างการเขียนภาษา ASP.NET

2.1.1.2 ความแตกต่างระหว่างกลุ่มเซิร์ฟเวอร์ และกลุ่มภาษาโปรแกรม (PHP, Java, Servlet, Node.js, Go, ASP)

การพัฒนาเว็บแอปพลิเคชันต้องอาศัยทั้ง เซิร์ฟเวอร์ (Server) และ ภาษาโปรแกรม (Programming Language) เพื่อประมวลผลและตอบสนองคำขอจากผู้ใช้งาน ทั้งสองกลุ่มนี้มีบทบาทที่แตกต่างกันในระบบเว็บ ซึ่งสามารถเปรียบเทียบได้ดังนี้

กลุ่มเซิร์ฟเวอร์ (Nginx, Apache, IIS)

เซิร์ฟเวอร์เป็นซอฟต์แวร์ที่จัดการคำขอ HTTP และตอบกลับไปยังผู้ใช้ในรูปแบบของเนื้อหาเว็บ หรือข้อมูล โดยเซิร์ฟเวอร์ที่นิยมใช้ประกอบด้วย Nginx Apache และ IIS

■ Nginx

ลักษณะ เซิร์ฟเวอร์ที่ออกแบบมาเพื่อจัดการการเชื่อมต่อจำนวนมากอย่างมีประสิทธิภาพ ด้วยการทำงานแบบ Asynchronous และ Event-Driven

ข้อดี

- 1) ประสิทธิภาพสูงสำหรับการให้บริการเว็บที่มีการเข้าชมสูง
- 2) รองรับ Reverse Proxy และ Load Balancing
- 3) ใช้หน่วยความจำน้อยเมื่อเทียบกับ Apache

ข้อจำกัด การกำหนดค่าซับซ้อนกว่า Apache ในบางกรณี

■ Apache

ลักษณะ เซิร์ฟเวอร์ที่มีค่าที่ต่ำ รองรับโมดูลหลากหลายแบบ Dynamic Module

ข้อดี

- 1) ความสามารถในการจัดการเนื้อหาแบบโอนามิกและคงที่ได้ดี
- 2) มีชุมชนขนาดใหญ่พร้อมเอกสารสนับสนุน

ข้อจำกัด ใช้ทรัพยากรากกว่า Nginx เมื่อรองรับคำขอพร้อมกันจำนวนมาก

■ IIS (Internet Information Services)

ลักษณะ เว็บเซิร์ฟเวอร์ของ Microsoft ที่ออกแบบมาสำหรับ Windows Server

ข้อดี

- 1) ผู้สนับสนุนกับเทคโนโลยีของ Microsoft ได้ดี เช่น ASP.NET
- 2) รองรับการจัดการและการรักษาความปลอดภัยที่เหมาะสมในระบบองค์กร

ข้อจำกัด

- 1) การใช้งานจำกัดเฉพาะแพลตฟอร์ม Windows
- 2) ประสิทธิภาพอาจต่ำกว่า Nginx หรือ Apache ในบางกรณี

กลุ่มภาษาโปรแกรม

ภาษาโปรแกรมใช้สำหรับการพัฒนาแอปพลิเคชันผ่านเซิร์ฟเวอร์ โดยจัดการกับคำขอจากผู้ใช้ และส่งข้อมูลหรือเนื้อหาเว็บกลับไป

กลุ่มเซิร์ฟเวอร์ (Nginx, Apache, IIS) มุ่งเน้นไปที่การให้บริการเนื้อหาเว็บและจัดการคำขอจากผู้ใช้ ขณะที่กลุ่มภาษาโปรแกรม (PHP, JAVA Servlet, Node.js, Go, ASP) มุ่งเน้นไปที่การพัฒนาแอปพลิเคชันที่ทำงานบนเซิร์ฟเวอร์ ทั้งสองกลุ่มนี้มีจุดเด่นและข้อจำกัดที่แตกต่างกัน ซึ่งขึ้นอยู่กับความต้องการเฉพาะของโครงการหรือองค์กร

2.1.2 Nginx

Nginx (อ่านว่า "Engine-X") [15] [16] [17] [18] เป็นเว็บเซิร์ฟเวอร์โอเพนซอร์สที่ได้รับการพัฒนาโดย Igor Sysoev ในปี ค.ศ. 2004 โดยมีเป้าหมายเพื่อแก้ไขปัญหา C10K problem ซึ่งเกี่ยวกับการจัดการการเชื่อมต่อจำนวนมากพร้อมกันในเวลาเดียวกัน ปัจจุบัน Nginx ได้รับการยอมรับในฐานะหนึ่งในเว็บเซิร์ฟเวอร์ที่ได้รับความนิยมและใช้กันอย่างแพร่หลายในระดับโลก เนื่องจากมีประสิทธิภาพสูง และความยืดหยุ่นในการใช้งาน

คุณสมบัติเด่นของ Nginx

- 1) ประสิทธิภาพสูง Nginx สามารถจัดการคำขอพร้อมกันได้หลายพันคำขอ โดยใช้ทรัพยากระบบน้อยกว่าเว็บเซิร์ฟเวอร์อื่น เช่น Apache
- 2) สถาปัตยกรรมแบบซิงโครนัส Nginx ใช้ event-driven architecture ซึ่งช่วยให้สามารถจัดการคำขอได้อย่างมีประสิทธิภาพ โดยไม่ต้องสร้างเครดิทใหม่สำหรับแต่ละคำขอ

- 3) รองรับโปรโตคอลหลากหลาย นอกจากการรองรับ HTTP/HTTPS แล้ว Nginx ยังสามารถจัดการโปรโตคอลอื่น ๆ เช่น WebSocket, HTTP/2 และ RTMP สำหรับการสตรีมวิดีโอ
- 4) พัฒนาการทำงานที่ยืดหยุ่น Nginx สามารถทำหน้าที่เป็น reverse proxy server, load balancer และ HTTP cache ซึ่งหมายความว่าสามารถใช้งานในระบบที่ซับซ้อน

ข้อดีของ Nginx

- 1) รองรับการเชื่อมต่อพร้อมกันจำนวนมาก ด้วยสถาปัตยกรรมแบบ event-driven ทำให้ Nginx สามารถรองรับการเชื่อมต่อพร้อมกันได้เป็นจำนวนมาก
- 2) การใช้ทรัพยากรอบตัว Nginx ถูกออกแบบมาให้ใช้ RAM และ CPU น้อย ทำให้เหมาะสมสำหรับการใช้งานในเซิร์ฟเวอร์ที่มีข้อจำกัดด้านทรัพยากร
- 3) ความเสถียรสูง Nginx สามารถทำงานได้อย่างต่อเนื่องแม้ในสถานการณ์ที่มีการใช้งานสูง เช่น เว็บไซต์ที่มีผู้ใช้งานจำนวนมาก
- 4) การตั้งค่าที่ยืดหยุ่น แม้การตั้งค่าอาจดูซับซ้อนสำหรับผู้เริ่มต้น แต่ Nginx มีเอกสารสนับสนุนและชุมชนผู้ใช้งานที่กว้างขวาง ช่วยให้การปรับแต่งสามารถทำได้ง่ายขึ้น

ข้อจำกัดของ Nginx

- 1) การตั้งค่าที่ซับซ้อนสำหรับผู้เริ่มต้น การกำหนดค่า Nginx อาจเป็นเรื่องยากสำหรับผู้ที่ไม่มีประสบการณ์ แม้ว่าจะมีเอกสารช่วยเหลืออย่างละเอียด
- 2) รองรับโมดูลน้อยกว่า Apache ในขณะที่ Apache มีโมดูลจำนวนมากที่สามารถใช้งานได้ทันที Nginx อาจต้องพึ่งพาเครื่องมือภายนอกสำหรับฟังก์ชันบางอย่าง
- 3) ข้อจำกัดในการสนับสนุนบางฟังก์ชัน Nginx อาจต้องใช้วิธีการพิเศษหรือการตั้งค่าเพิ่มเติม สำหรับการใช้งานบางประเภท เช่น การสนับสนุนสำหรับ Dynamic Content Nginx เป็นตัวเลือกที่ยอดเยี่ยมสำหรับผู้พัฒนาเว็บและองค์กรที่ต้องการประสิทธิภาพสูงและความสามารถในการจัดการผู้ใช้งานจำนวนมาก โดยเฉพาะในบริบทของเว็บไซต์หรือแอปพลิเคชันที่ต้องการความเร็วและเสถียรภาพ

2.1.3 Docker

Docker [19] [20] เป็นแพลตฟอร์มโอเพนซอร์สที่ช่วยในการสร้าง ทดสอบและปรับใช้อัปพลิเคชันในรูปแบบของ คอนเทนเนอร์ (Containers) ซึ่งเป็นสภาพแวดล้อมแยกจากกัน ทำให้อัปพลิเคชันสามารถทำงานได้อย่างอิสระจากระบบปฏิบัติการหลัก โดย Docker ช่วยเพิ่มความสะดวกและประสิทธิภาพในการพัฒนาและการจัดการอัปพลิเคชัน โดยเฉพาะในโลกของ DevOps และ Cloud Computing

หน้าที่ของ Docker

- สร้างค่อนเทนเนอร์ Docker ช่วยสร้างค่อนเทนเนอร์ที่บรรจุทุกสิ่งที่จำเป็นสำหรับการทำงานของแอปพลิเคชัน เช่น ไลบรารี การตั้งค่า และโค้ดของแอปพลิเคชัน
- จัดการค่อนเทนเนอร์ สามารถเริ่มต้น (start) หยุด (stop) และลบ (remove) ค่อนเทนเนอร์ ได้อย่างง่ายดายผ่านคำสั่ง
- แชร์และปรับใช้ Docker Hub ช่วยให้ผู้ใช้งานสามารถแชร์และดาวน์โหลด Docker Images เพื่อใช้งานซ้ำหรือปรับแต่งได้

คุณสมบัติของ Docker

- ประสิทธิภาพสูง Docker ใช้เคอร์เนลของระบบปฏิบัติการเดียวกัน (Shared Kernel) สำหรับการรันค่อนเทนเนอร์ ทำให้ลดการใช้ทรัพยากรและเพิ่มความเร็ว
- การเริ่มต้นที่รวดเร็ว ค่อนเทนเนอร์สามารถเริ่มทำงานได้ในเวลาไม่กี่วินาที เนื่องจากไม่มีการจำลองระบบปฏิบัติการทั้งหมดเหมือน Virtual Machine
- ความยืดหยุ่น รองรับการใช้งานบนแพลตฟอร์มหลากหลาย เช่น Linux Windows และ macOS

ข้อดีของ Docker

- ลดความซับซ้อนในการจัดการสภาพแวดล้อม ค่อนเทนเนอร์ช่วยให้แอปพลิเคชันรันในสภาพแวดล้อมเดียวกันในทุกขั้นตอน ตั้งแต่การพัฒนาไปจนถึงการผลิต
- การปรับขนาดที่ง่ายดาย สามารถเพิ่มหรือลดจำนวนค่อนเทนเนอร์ได้ตามความต้องการของระบบ
- ลดปัญหาความเข้ากันไม่ได้ของสภาพแวดล้อม ค่อนเทนเนอร์บรรจุทุกสิ่งที่จำเป็น ทำให้ลดปัญหาจากการเปลี่ยนแปลงของระบบปฏิบัติการหรือเวอร์ชันของไลบรารี

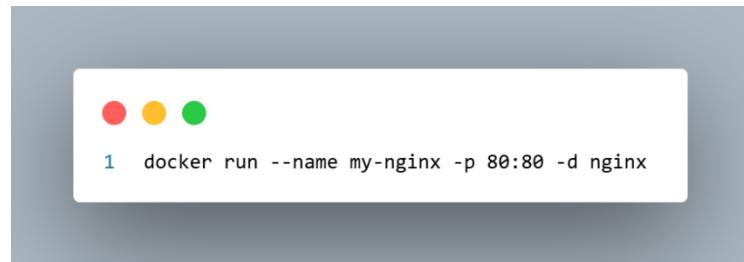
ข้อจำกัดของ Docker

- การจัดการเครื่อข่ายซับซ้อน การตั้งค่าการเชื่อมต่อระหว่างค่อนเทนเนอร์ในระบบขนาดใหญ่มีความซับซ้อน
- ใช้ทรัพยากรยะห์ การใช้ Docker จำเป็นต้องมีระบบปฏิบัติการหลัก สำหรับติดตั้งซอฟต์แวร์ในการสร้างค่อนเทนเนอร์
- ประสิทธิภาพในการประมวลผล เนื่องจากต้องมีระบบปฏิบัติการหลักสำหรับติดตั้งซอฟต์แวร์ จึงต้องมีการใช้พลังประมวลผลเพิ่มเติมในแต่ละค่อนเทนเนอร์และค่อนเทนเนอร์มายังระบบปฏิบัติการหลัก
- เวลาการเรียนรู้การใช้ Docker เป็นต้นสามารถใช้งานได้ง่าย แต่หากต้องการควบคุมองค์ประกอบขั้นสูง เช่น การสร้างอิเมจใหม่สำหรับเชื่อมต่อระหว่างค่อนเทนเนอร์ เป็นต้น

ตัวอย่างการใช้งาน Docker กับ Web Server

1) Nginx

Nginx สามารถรันบน Docker ได้อย่างง่ายดายและมีประสิทธิภาพสูงสำหรับการจัดการคำขอ HTTP ตัวอย่างคำสั่ง

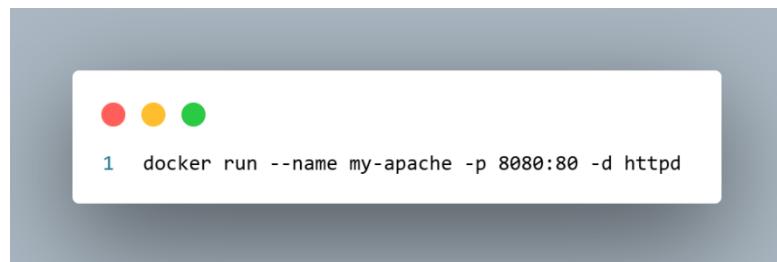


```
1 docker run --name my-nginx -p 80:80 -d nginx
```

รูปที่ 2.7 ตัวอย่างคำสั่ง Nginx

2) Apache HTTP Server

Apache HTTP Server เป็นอีกหนึ่งเว็บเซิร์ฟเวอร์ยอดนิยมที่ใช้งานร่วมกับ Docker ได้ดี ตัวอย่างคำสั่ง

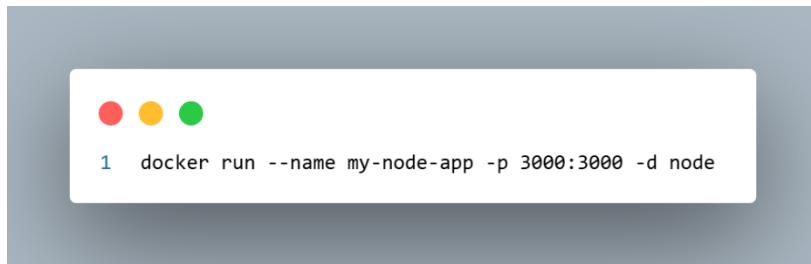


```
1 docker run --name my-apache -p 8080:80 -d httpd
```

รูปที่ 2.8 ตัวอย่างคำสั่ง Nginx

3) Node.js

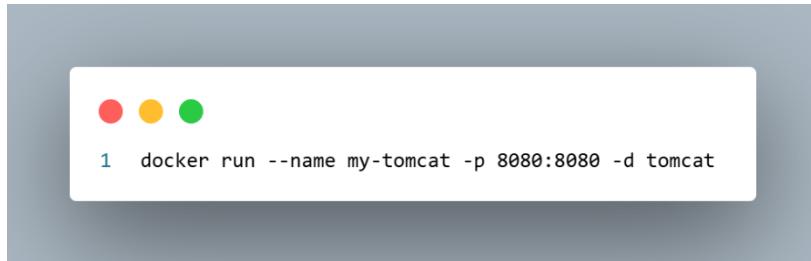
Docker ช่วยให้การสร้างและรันแอปพลิเคชัน Node.js ง่ายขึ้น
ตัวอย่างคำสั่ง



รูปที่ 2.9 ตัวอย่างคำสั่ง Nginx

4) Tomcat

สำหรับแอปพลิเคชัน Java Servlet, Tomcat สามารถรันใน Docker ได้อย่างมีประสิทธิภาพ
ตัวอย่างคำสั่ง



รูปที่ 2.10 ตัวอย่างคำสั่ง Tomcat

Docker เป็นเครื่องมือที่มีประสิทธิภาพสำหรับนักพัฒนาที่ต้องการสร้างและจัดการแอปพลิเคชันในรูปแบบที่ยืดหยุ่นและง่ายต่อการปรับใช้ โดยเฉพาะในโลกของ DevOps และ Cloud Computing

2.1.4 Kubernetes

Kubernetes (K8s) [21] [22] [23] เป็นแพลตฟอร์มโอเพนซอร์สที่ได้รับการยอมรับมาเพื่อบริหารจัดการแอปพลิเคชันที่รันในคอนเทนเนอร์ โดยเฉพาะในการปรับใช้ (deployment) การปรับขนาด (scaling) และการจัดการคอนเทนเนอร์ต่าง ๆ แบบอัตโนมัติ Kubernetes ช่วยลดความซับซ้อนของระบบขนาดใหญ่ และเพิ่มประสิทธิภาพในการจัดการโครงสร้างพื้นฐานที่มีการเปลี่ยนแปลงอย่างรวดเร็วในยุค Cloud Computing และ Microservices Architecture

หน้าที่ของ Kubernetes

- 1) การปรับใช้แอปพลิเคชัน (Automating Deployment) Kubernetes ช่วยในการปรับใช้แอปพลิเคชันในค่อนเทนเนอร์อย่างเป็นระบบ โดยมีความสามารถในการสร้างและจัดการค่อนเทนเนอร์ได้อัตโนมัติ
- 2) การปรับขนาด (Scaling Applications) รองรับการปรับขนาดของแอปพลิเคชันทั้งในแนวนอน (horizontal scaling) และแนวตั้ง (vertical scaling) โดยอิงตามทรัพยากรหรือโหลดงาน
- 3) การกระจายโหลด (Load Balancing) Kubernetes มีระบบ Load Balancer ที่ช่วยกระจายทรัพยากรไปยังค่อนเทนเนอร์ต่าง ๆ อย่างสมดุล ทำให้แอปพลิเคชันทำงานได้อย่างต่อเนื่อง
- 4) การตรวจสอบและการจัดการ (Monitoring and Management) มีเครื่องมือในตัวสำหรับติดตามสถานะของแอปพลิเคชัน เช่น การตรวจสอบการใช้งานทรัพยากร และการจัดการข้อมูลแพด

โครงสร้างของ Kubernetes

Kubernetes แบ่งโครงสร้างการทำงานออกเป็นสองส่วนหลัก

- 1) Master Node ทำหน้าที่ควบคุมและจัดการ Cluster โดยรับผิดชอบการจัดสรรทรัพยากรและควบคุม Worker Nodes ผ่านส่วนประกอบสำคัญ เช่น
 - API Server ตัวกลางสำหรับการสื่อสารระหว่างผู้ใช้งานและระบบ
 - Controller Manager ดูแลกระบวนการควบคุม เช่น การสร้าง Pod หรือการกู้คืนระบบ
 - Scheduler จัดการการวางแผนตำแหน่งของค่อนเทนเนอร์ใน Worker Nodes
- 2) Worker Nodes ทำหน้าที่รันค่อนเทนเนอร์ต่าง ๆ ที่ได้รับมอบหมายจาก Master Node โดยมีส่วนประกอบสำคัญ
 - Kubelet ตัวแทนที่ทำหน้าที่เชื่อมต่อ Worker Node กับ Master Node
 - Container Runtime เช่น Docker หรือ containerd สำหรับรันค่อนเทนเนอร์
 - Kube-Proxy จัดการการสื่อสารเครือข่ายภายในและระหว่าง Pods

ส่วนประกอบหลักของ Kubernetes

- 1) Pod หน่วยการทำงานที่เล็กที่สุดใน Kubernetes ซึ่งสามารถประกอบด้วยหนึ่งหรือหลายค่อนเทนเนอร์ที่แชร์ทรัพยากร เช่น IP Address และ Storage
- 2) Service ทำหน้าที่เชื่อมต่อ Pods และให้การเข้าถึงจากภายนอก Cluster พร้อมทั้งทำ Load Balancing
- 3) Ingress จัดการการเข้าถึงทรัพยากรจากภายนอก Cluster ไปยังบริการภายใน Cluster ผ่าน URL Routing

- 4) Namespace ใช้สำหรับแยกส่วนการทำงานภายใน Cluster เพื่อการจัดการที่ง่ายและลดความซับซ้อน

ข้อดีของ Kubernetes

- 1) เพิ่มประสิทธิภาพการจัดการระบบที่ซับซ้อน Kubernetes ช่วยจัดการระบบที่ประกอบด้วยหลายคอนเทนเนอร์ได้ง่ายขึ้น
- 2) รองรับการปรับขนาดได้ดี สามารถปรับขนาดแอปพลิเคชันแบบอัตโนมัติตามความต้องการของทรัพยากร
- 3) รองรับการทำงานแบบกระจาย Kubernetes ช่วยกระจายงานและจัดการทรัพยากรในระบบที่มีหลาย Node
- 4) เสถียรภาพและความพร้อมใช้งานสูง Kubernetes ช่วยตรวจสอบและกู้คืน Pods ที่ล้มเหลวโดยอัตโนมัติ

ข้อจำกัดของ Kubernetes

- 1) ความซับซ้อนในการตั้งค่า การติดตั้งและกำหนดค่าระบบ Kubernetes ต้องอาศัยความรู้ด้านเทคนิคสูง
- 2) การจัดการทรัพยากร Kubernetes อาจใช้ทรัพยากรมากเกินไปในระบบขนาดเล็ก
- 3) ต้องการการดูแลที่ต่อเนื่อง ระบบ Kubernetes ต้องการการตรวจสอบและอัปเดตซอฟต์แวร์อย่างสม่ำเสมอ

Kubernetes เป็นเครื่องมือที่สำคัญสำหรับการจัดการแอปพลิเคชันในคอนเทนเนอร์ โดยช่วยให้นักพัฒนาและผู้ดูแลระบบสามารถควบคุมและจัดการระบบขนาดใหญ่ที่ซับซ้อนได้อย่างมีประสิทธิภาพ อย่างไรก็ตาม การใช้งาน Kubernetes ควรพิจารณาถึงความซับซ้อนของระบบและการจัดการทรัพยากรเพื่อให้สามารถใช้งานได้เต็มประสิทธิภาพและเหมาะสมกับลักษณะของแอปพลิเคชัน

2.1.5 SSL

SSL (Secure Socket Layer) [24] [25] เป็นโปรโตคอลที่ได้รับการยอมรับมาเพื่อรักษาความปลอดภัยในการสื่อสารระหว่าง ผู้ใช้ (Client) และ เซิร์ฟเวอร์ (Server) โดยการเข้ารหัสข้อมูลที่ส่งผ่านเครือข่ายอินเทอร์เน็ต SSL มีบทบาทสำคัญในการปกป้องข้อมูลที่ละเอียดอ่อน เช่น ข้อมูลส่วนบุคคล และข้อมูลทางการเงิน เพื่อป้องกันการดักจับและการโจมตีจากบุคคลที่สาม ด้วยเหตุนี้ SSL จึงกลายเป็นมาตรฐานสำคัญสำหรับการสร้างความน่าเชื่อถือและความมั่นใจในโลกดิจิทัล

หน้าที่ของ SSL

- 1) การเข้ารหัสข้อมูล (Data Encryption) SSL ช่วยปกป้องข้อมูลที่ส่งผ่านเครือข่ายอินเทอร์เน็ต โดยการเข้ารหัสข้อมูลเพื่อป้องกันไม่ให้บุคคลที่สามสามารถดักจับและอ่านข้อมูลระหว่างทางได้
- 2) การตรวจสอบตัวตน (Authentication) เว็บไซต์ที่ใช้ SSL จะมีใบรับรอง SSL Certificate ซึ่งทำหน้าที่ตรวจสอบและยืนยันตัวตนของเว็บไซต์นั้น ๆ เพื่อป้องกันการแอบอ้างตัวตน
- 3) การรักษาความสมบูรณ์ของข้อมูล (Data Integrity) SSL ช่วยตรวจสอบว่าข้อมูลที่ถูกส่งระหว่างต้นทางและปลายทางยังคงความสมบูรณ์ไม่ถูกเปลี่ยนแปลงหรือตัดแปลงในระหว่างการส่ง

ประเภทของ SSL Certificates

- 1) Domain Validated (DV) เป็นใบรับรองที่ตรวจสอบความถูกต้องของโดเมนเท่านั้น โดยมักจะมาสำหรับเว็บไซต์ส่วนบุคคลหรือบล็อกที่ไม่ได้มีข้อมูลสำคัญ
- 2) Organization Validated (OV) เป็นใบรับรองที่ตรวจสอบทั้งโดเมนและข้อมูลองค์กร ที่ยื่นขอใบรับรอง ซึ่งช่วยเพิ่มความมั่นใจให้กับผู้ใช้งาน
- 3) Extended Validation (EV) เป็นใบรับรองที่ให้ความมั่นใจสูงสุด เนื่องจากมีการตรวจสอบอย่างละเอียดเกี่ยวกับองค์กร เช่น การยืนยันตัวตนทางกฎหมายและสถานะขององค์กร

ตัวอย่างการใช้งาน SSL

เว็บไซต์ที่ใช้ SSL จะมี URL เริ่มต้นด้วย "https://" แทนที่จะเป็น "http://" และมักแสดงโฉมหนูรูปกุญแจในแถบที่อยู่ (Address Bar) ของเบราว์เซอร์ ตัวอย่างการใช้งาน SSL ได้แก่

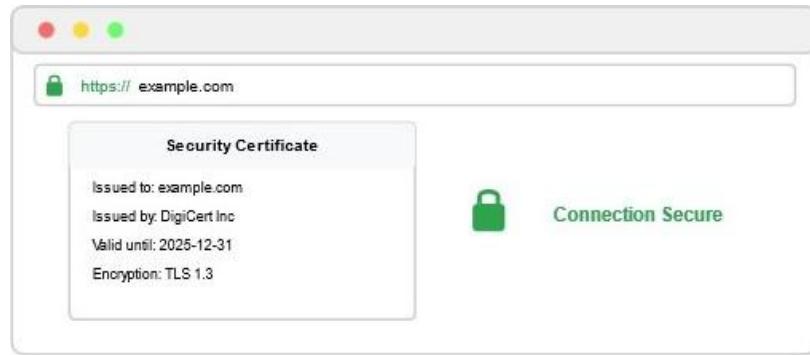
- 1) ธนาคารออนไลน์ ธนาคารใช้งาน SSL เพื่อปกป้องข้อมูลทางการเงิน เช่น เลขบัญชีและข้อมูลการทำธุรกรรม
- 2) ร้านค้าออนไลน์ แพลตฟอร์ม e-commerce ใช้ SSL เพื่อรักษาความปลอดภัยในการชำระเงิน ผ่านบัตรเครดิตและข้อมูลการสั่งซื้อ
- 3) แพลตฟอร์มโซเชียลมีเดีย แพลตฟอร์มโซเชียลมีเดียใช้ SSL เพื่อปกป้องข้อมูลส่วนบุคคลของผู้ใช้งาน เช่น ข้อมูลโปรไฟล์และข้อมูลส่วนตัว

ข้อดีของ SSL

- 1) เพิ่มความปลอดภัย ช่วยป้องกันการโจมตีแบบ Man-in-the-Middle (MITM) และการดักจับข้อมูล
- 2) สร้างความน่าเชื่อถือ เว็บไซต์ที่ใช้ SSL ช่วยเพิ่มความมั่นใจให้กับผู้ใช้ เนื่องจากแสดงถึงความตั้งใจในการปกป้องข้อมูล
- 3) ปรับปรุง SEO เว็บไซต์ที่มี SSL ได้รับการจัดอันดับที่ดีขึ้นในผลการค้นหาของ Google

ข้อจำกัดของ SSL

- 1) ค่าใช้จ่าย การติดตั้ง SSL Certificate โดยเฉพาะประเภท OV และ EV อาจมีค่าใช้จ่ายสูง
- 2) การตั้งค่าซับซ้อน การติดตั้งและจัดการ SSL อาจต้องการความเชี่ยวชาญด้านเทคนิค
- 3) ผลกระทบต่อประสิทธิภาพ การเข้ารหัสและถอดรหัสข้อมูลอาจเพิ่มเวลาในการโหลดหน้าเว็บ

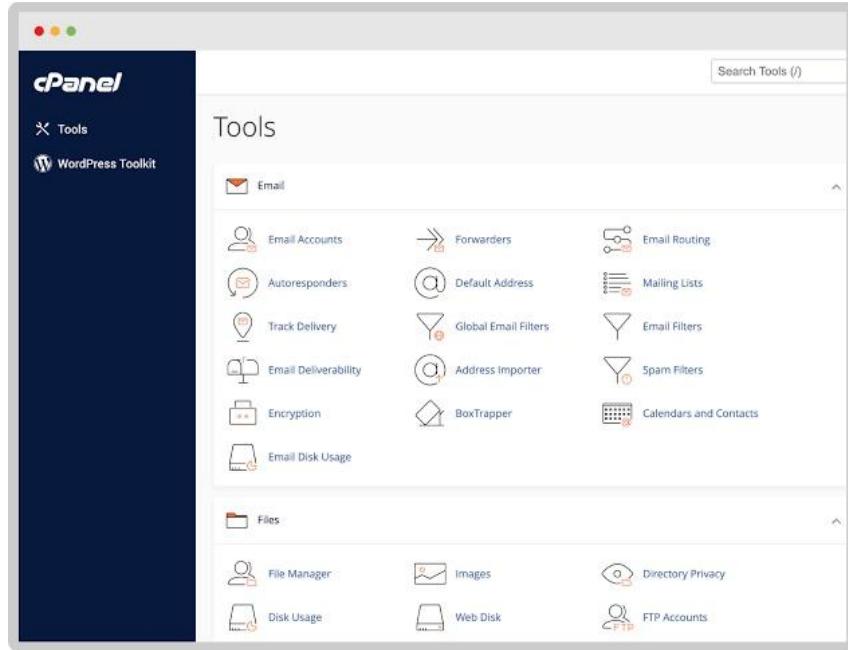


รูปที่ 2.11 องค์ประกอบทั้งหมดที่เกี่ยวข้องกับการแสดงผล SSL Certificate ในเบราว์เซอร์ทั่วไป

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 cPanel

cPanel [26] เป็นระบบควบคุมการจัดการเว็บไซต์ที่ได้รับความนิยมสูงสุดในปัจจุบัน โดยได้รับการออกแบบมาเพื่ออำนวยความสะดวกแก่ผู้ใช้งานในการบริหารจัดการเว็บไซต์ผ่านอินเทอร์เฟซแบบกราฟิกที่ใช้งานง่าย ด้วยฟีเจอร์ที่ครบถ้วนและครอบคลุมทุกความต้องการในการจัดการเว็บไซต์ ทำให้ cPanel กลายเป็นตัวเลือกที่ได้รับความนิยมอย่างแพร่หลายในอุตสาหกรรมไซต์



รูปที่ 2.12 cPanel Control Panel

การทำงานของ cPanel

cPanel แบ่งการทำงานออกเป็น 3 ส่วนหลัก ได้แก่

- 1) Web Hosting Manager (WHM) สำหรับผู้ดูแลระบบ (Administrator) ใช้ในการจัดการเซิร์ฟเวอร์ และบัญชีผู้ใช้งานทั้งหมด เช่น การตั้งค่าเซิร์ฟเวอร์ การสร้างบัญชีไฮสติ๊กใหม่ และการกำหนดนโยบายด้านความปลอดภัยของเซิร์ฟเวอร์
- 2) cPanel Control Panel เป็นอินเทอร์เฟซที่ผู้ใช้ทั่วไปสามารถใช้ในการบริหารจัดการเว็บไซต์ เช่น การจัดการไฟล์ การตั้งค่าโดเมน อีเมล และฐานข้อมูล
- 3) Web Hosting Manager Reseller (WHM Reseller) สำหรับผู้ให้บริการไฮสติ๊กรายย่อย (Reseller) ที่ต้องการแบ่งทรัพยากรและจัดการบัญชีลูกค้าผ่านระบบที่ง่ายต่อการใช้งาน

ความสามารถของ cPanel

cPanel มาพร้อมกับฟีเจอร์ที่ช่วยให้ผู้ใช้งานสามารถบริหารจัดการเว็บไซต์ได้อย่างครอบคลุม โดยมีความสามารถที่สำคัญดังต่อไปนี้

1) การจัดการไฟล์ (File Management)

- ผู้ใช้สามารถอัปโหลด ดาวน์โหลด และแก้ไขไฟล์ผ่าน File Manager ได้โดยไม่จำเป็นต้องใช้โปรแกรม FTP
- รองรับการจัดการลิขิตรการเข้าถึงไฟล์และการกำหนดโครงสร้างไดเรกทอรี

2) การจัดการฐานข้อมูล (Database Management)

- รองรับการจัดการฐานข้อมูล MySQL และ MariaDB ผ่าน phpMyAdmin
- สามารถสร้างและแก้ไขฐานข้อมูลได้โดยง่ายผ่านอินเทอร์เฟซที่เป็นมิตรต่อผู้ใช้

3) การจัดการโดเมน (Domain Management)

- การเพิ่มและตั้งค่าชื่อโดเมนหลัก ชับโดเมน และโดเมนเสริม
- การกำหนดการเปลี่ยนเส้นทาง (Redirects) และจัดการ DNS ได้อย่างละเอียด

4) การจัดการอีเมล (Email Management)

- สามารถสร้างบัญชีอีเมลภายในไดเมนของตนเอง พร้อมตั้งค่า Forwarders และ Auto-responders
- ฟีเจอร์ป้องกันสแปมอีเมล และระบบกรองอีเมลอัตโนมัติ

5) การติดตั้งแอปพลิเคชัน (Application Installation)

- รองรับการติดตั้ง CMS ยอดนิยม เช่น WordPress, Joomla, Drupal ผ่านตัวติดตั้งอัตโนมัติ เช่น Softaculous หรือ Installatron

6) การวิเคราะห์และตรวจสอบเว็บไซต์ (Website Analytics & Monitoring)

- ให้ข้อมูลเชิงลึกเกี่ยวกับปริมาณการเข้าชมเว็บไซต์ การใช้ทรัพยากร และประสิทธิภาพของเซิร์ฟเวอร์

จุดเด่นของ cPanel

- ใช้งานง่าย (User-friendly Interface) อินเทอร์เฟซแบบกราฟิกที่ออกแบบมาให้ใช้งานง่าย โดยไม่จำเป็นต้องมีความรู้ทางเทคนิคขั้นสูง
- ฟีเจอร์ครบครัน (Comprehensive Features) รองรับฟังก์ชันการจัดการไซต์ที่หลากหลาย ช่วยลดความยุ่งยากในการดูแลเว็บไซต์

3) รองรับหลายภาษา (Multi-language Support) รองรับภาษาต่าง ๆ รวมถึงภาษาไทย ทำให้ผู้ใช้งานสามารถใช้งานได้ง่ายขึ้น

4) ระบบรักษาความปลอดภัยสูง (Security Features) มีเครื่องมือสำหรับการป้องกันข้อมูล เช่น SSL, Firewall และการป้องกันการโจมตีจากมัลแวร์

ข้อดีด้อยของ cPanel

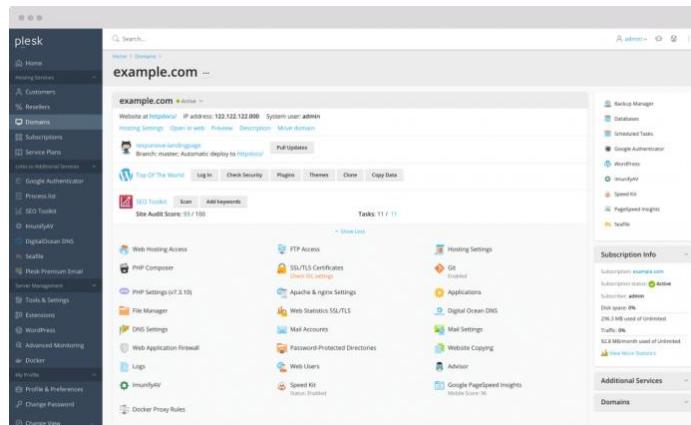
1) ค่าใช้จ่ายสูง (High Cost) ในการซื้อและติดตั้ง cPanel มีค่าใช้จ่ายที่สูงกว่าแพลตฟอร์มอื่น ๆ ซึ่งอาจเป็นภาระสำหรับผู้ประกอบการรายย่อย

2) ความซับซ้อนในการตั้งค่าเริ่มต้น (Complex Initial Setup) ผู้ใช้งานที่ไม่มีประสบการณ์อาจพบความซับซ้อนในขั้นตอนการติดตั้งและกำหนดค่าครั้งแรก

3) ข้อจำกัดด้านทรัพยากร (Resource-intensive) อาจไม่เหมาะสมกับเว็บไซต์ขนาดใหญ่ที่ต้องการทรัพยากรเช่นฟลีตเวอร์สูง เนื่องจากอาจทำให้เซิร์ฟเวอร์ทำงานช้าลง

2.2.2 Plesk

Plesk [27] เป็นซอฟต์แวร์สำหรับการจัดการเว็บไซต์ที่ได้รับความนิยมอย่างแพร่หลายในหมู่ผู้ให้บริการไซต์และนักพัฒนาเว็บไซต์ ด้วยความสามารถที่ครอบคลุมและการออกแบบที่รองรับการใช้งานบนแพลตฟอร์มต่าง ๆ ทั้ง Windows และ Linux ทำให้ Plesk เป็นเครื่องมือที่เหมาะสมสำหรับการบริหารจัดการเว็บไซต์และเซิร์ฟเวอร์อย่างมืออาชีพ



รูปที่ 2.13 Plesk Control Panel

การทำงานของ Plesk

Plesk ทำงานโดยให้ผู้ใช้งานสามารถเข้าถึงและควบคุมการบริหารจัดการเซิร์ฟเวอร์ผ่านอินเทอร์เฟซแบบกราฟิก (Graphical User Interface หรือ GUI) ที่ใช้งานง่าย โดยมีฟีเจอร์หลักที่ช่วยอำนวยความสะดวกในการจัดการโดเมน ไฟล์ ฐานข้อมูล อีเมล และการติดตั้งแอปพลิเคชันต่าง ๆ จากศูนย์ควบคุมเดียวทั้งหมด

ความสามารถของ Plesk ครอบคลุมการทำงานทั้งในระดับผู้ให้บริการไฮสติง (Hosting Providers) และผู้ดูแลระบบ (Administrators) โดยสามารถแบ่งการใช้งานออกเป็นระดับต่าง ๆ ได้แก่

- 1) Administrator Level สำหรับการบริหารจัดการเซิร์ฟเวอร์ทั้งหมด รวมถึงการสร้างบัญชีผู้ใช้งาน
- 2) Reseller Level สำหรับผู้ให้บริการไฮสติงย่อยที่สามารถจัดการลูกค้าของตนเองได้
- 3) User Level สำหรับเจ้าของเว็บไซต์ที่สามารถจัดการเนื้อหาเว็บไซต์และอีเมลของตนเอง

ความสามารถของ Plesk

Plesk มาพร้อมฟีเจอร์ที่ช่วยให้ผู้ใช้งานสามารถบริหารจัดการเว็บไฮสติงได้อย่างครบถ้วน โดยมีความสามารถสำคัญดังนี้

- 1) การจัดการโดเมนและไฮสติง (Domain & Hosting Management)
 - เพิ่ม ลบ และแก้ไขเว็บไซต์
 - รองรับการติดตั้งระบบจัดการเนื้อหา (CMS) เช่น WordPress, Joomla และ Drupal
 - ตั้งค่าโดเมนย่อย (Subdomains) และกำหนดค่า DNS ได้โดยง่าย
- 2) การจัดการไฟล์ (File Management)
 - มีตัวจัดการไฟล์ (File Manager) ที่ใช้งานง่าย
 - รองรับการอัปโหลดและดาวน์โหลดไฟล์ผ่านอินเทอร์เฟซกราฟิกหรือโปรโตคอล FTP
- 3) การจัดการฐานข้อมูล (Database Management)
 - รองรับ MySQL และ MariaDB รวมถึงการเข้าถึง phpMyAdmin
 - ฟีเจอร์การสำรองข้อมูลและคืนฐานข้อมูลอัตโนมัติ
- 4) การจัดการอีเมล (Email Management)
 - สร้างและจัดการบัญชีอีเมลภายใต้โดเมนของตนเอง
 - ฟีเจอร์ป้องกันสแปม (Spam Filtering) และการตั้งค่าการส่งต่ออีเมล (Email Forwarding)

5) การติดตั้งแอปพลิเคชัน (Application Installation)

- รองรับการติดตั้งแอปพลิเคชันยอดนิยมผ่าน Plesk Application Vault
- ติดตั้งเครื่องมือสำหรับนักพัฒนา เช่น Git, Node.js และ Docker

6) ระบบรักษาความปลอดภัย (Security Features)

- รองรับการติดตั้งในรับรองความปลอดภัย SSL/TLS
- ระบบ Firewall และเครื่องมือป้องกันการโจมตีจากมัลแวร์
- รองรับการตั้งค่าการเข้าถึงโดยใช้ Two-Factor Authentication (2FA)

จุดเด่นของ Plesk

- อินเทอร์เฟซที่ใช้งานง่าย (User-friendly Interface) รูปแบบการจัดการที่เป็นระบบและเข้าถึงได้ง่าย ทำให้สามารถบริหารจัดการเว็บไซต์ได้อย่างรวดเร็วและสะดวก
- ฟีเจอร์ที่ครบครัน (Comprehensive Features) รองรับการจัดการทุกด้านของไซต์ในที่เดียว ตั้งแต่ไฟล์ไปจนถึงความปลอดภัยและการสำรองข้อมูล
- รองรับหลายแพลตฟอร์ม (Cross-platform Support) ทำงานได้ทั้งบนระบบปฏิบัติการ Windows และ Linux ช่วยเพิ่มความยืดหยุ่นให้กับผู้ใช้งาน
- ความปลอดภัยสูง (Enhanced Security) มีฟีเจอร์ด้านความปลอดภัยที่ครบถ้วน เช่น Web Application Firewall (WAF), Fail2Ban และเครื่องมือป้องกันการโจมตี DDoS

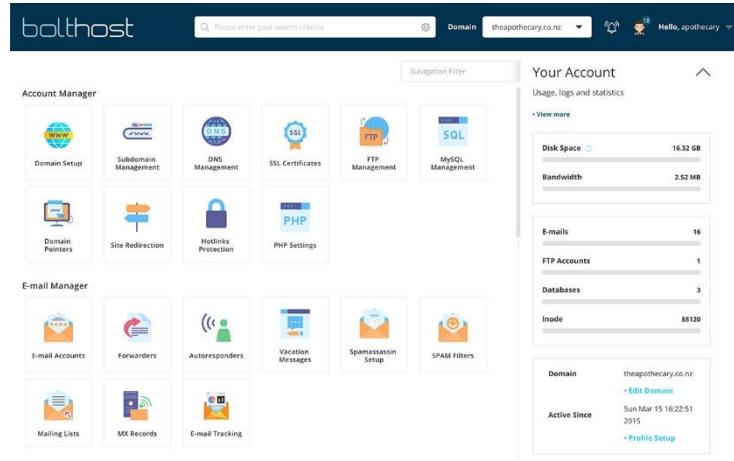
จุดด้อยของ Plesk

- ค่าใช้จ่ายสูง (High Cost) ใบอนุญาต Plesk มีราคาค่อนข้างสูง โดยแพ็คเกจเริ่มต้นอยู่ที่ประมาณ \$10 – \$30 ต่อเดือน ซึ่งอาจเป็นภาระสำหรับธุรกิจขนาดเล็ก
- ความซับซ้อนสำหรับผู้เริ่มต้น (Steep Learning Curve) แม้จะมีอินเทอร์เฟซที่ใช้งานง่าย แต่สำหรับผู้ที่ไม่มีประสบการณ์ด้านการจัดการไซต์อาจต้องใช้เวลาเรียนรู้ระบบ
- ข้อจำกัดในการขยายตัว (Scalability Limitations) แม้จะเหมาะสมสำหรับการใช้งานทั่วไป แต่สำหรับเว็บไซต์ที่มีปริมาณการเข้าชมสูงมากๆ อาจต้องใช้การกำหนดค่าที่ซับซ้อนเพิ่มเติม

2.2.3 DirectAdmin

DirectAdmin [28] เป็นหนึ่งในระบบควบคุมการจัดการเว็บไซต์ที่ได้รับความนิยม เนื่องจากมีฟีเจอร์ที่ครอบคลุมและราคาที่ต่ำค่า ทำให้เหมาะสมสำหรับผู้ให้บริการเว็บไซต์ นักพัฒนาเว็บไซต์ และ

ผู้ดูแลระบบ โดย DirectAdmin ออกแบบมาเพื่อให้สามารถบริหารจัดการเซิร์ฟเวอร์และเว็บไซต์ได้อย่างสะดวกและมีประสิทธิภาพ



รูปที่ 2.14 DirectAdmin Control Panel

การทำงานของ DirectAdmin

DirectAdmin ทำงานผ่านอินเทอร์เฟซแบบกราฟิก (Graphical User Interface หรือ GUI) ที่ช่วยให้ผู้ใช้สามารถดำเนินการต่าง ๆ ได้โดยไม่ต้องมีความรู้เชิงลึกเกี่ยวกับการบริหารเซิร์ฟเวอร์ ผู้ใช้งานสามารถดำเนินการที่เกี่ยวข้องกับการจัดการเว็บไซต์ เช่น การกำหนดค่าโดเมน อีเมล ไฟล์เว็บไซต์ ฐานข้อมูล และระบบสำรองข้อมูลผ่านหน้าจอเดียว โดย DirectAdmin รองรับโครงสร้างการบริหารใน 3 ระดับ ได้แก่

- 1) Administrator Level สำหรับผู้ดูแลระบบที่สามารถจัดการเซิร์ฟเวอร์ทั้งหมด รวมถึงการตั้งค่าระบบและบัญชีผู้ใช้
- 2) Reseller Level สำหรับผู้ให้บริการไซต์ที่สามารถบริหารลูกค้าและการใช้งานทรัพยากรได้
- 3) User Level สำหรับเจ้าของเว็บไซต์ที่สามารถจัดการเนื้อหาเว็บไซต์และบริการอื่น ๆ

ความสามารถของ DirectAdmin

DirectAdmin มาพร้อมกับฟีเจอร์สำคัญที่ช่วยอำนวยความสะดวกในการบริหารจัดการเว็บไซต์ ดังต่อไปนี้

1) การจัดการโดเมน (Domain Management)

- รองรับการเพิ่ม ลบ และแก้ไขโดเมน รวมถึงการตั้งค่าระบบ DNS และชับโดเมนได้อย่างง่ายดาย
- รองรับการตั้งค่า URL Forwarding และการใช้ชื่อโดเมนร่วมกัน (Domain Pointer)

2) การจัดการบัญชีผู้ใช้ (User Management)

- สามารถสร้างและกำหนดลิทธิ์การเข้าถึงของบัญชีผู้ใช้ได้ตามระดับการบริหารที่กำหนด
- ระบบจัดการโควต้าและการจำกัดทรัพยากรต่อผู้ใช้

3) การติดตั้งแอปพลิเคชัน (Application Installation)

- มีเครื่องมือสำหรับติดตั้งแอปพลิเคชันยอดนิยม เช่น WordPress, Joomla และ Drupal ผ่านระบบอัตโนมัติ

4) การจัดการฐานข้อมูล (Database Management)

- รองรับการสร้างและจัดการฐานข้อมูล MySQL ผ่าน phpMyAdmin
- ฟีเจอร์สำรองและกู้คืนข้อมูลฐานข้อมูลได้อย่างง่ายดาย

5) ระบบสำรองข้อมูล (Backup and Restore System)

- รองรับการสร้างข้อมูลสำรองแบบกำหนดเอง (Manual Backup) และการสำรองข้อมูลอัตโนมัติ (Automated Backup)
- สามารถกู้คืนข้อมูลผ่านอินเทอร์เฟซโดยไม่ต้องใช้คำสั่งผ่าน SSH

6) ระบบความปลอดภัย (Security Features)

- รองรับการติดตั้งในรับรอง SSL/TLS เพื่อเพิ่มความปลอดภัยให้กับเว็บไซต์
- มีระบบ Two-Step Authentication และการป้องกันการโจมตีแบบ Brute Force

จุดเด่นของ DirectAdmin

- ใช้งานง่าย (User-Friendly Interface) อินเทอร์เฟซที่เรียบง่ายและเข้าใจง่าย ช่วยให้ผู้ใช้งานสามารถบริหารจัดการเว็บไซต์ได้สะดวก แม้ไม่มีความรู้เชิงเทคนิคมากนัก
- ค่าใช้จ่ายที่คุ้มค่า (Cost-Effective Solution) DirectAdmin มีราคาค่าใช้จ่ายที่ถูกกว่าเมื่อเปรียบเทียบกับระบบควบคุมอื่น ๆ เช่น cPanel โดยมีค่าใช้จ่ายเฉลี่ยอยู่ที่ประมาณ \$5 – \$20 ต่อเดือน
- ฟีเจอร์ครบครัน (Comprehensive Features) แม้ว่า DirectAdmin จะมีต้นทุนต่ำ แต่ยังคงมีฟีเจอร์ที่เพียงพอต่อความต้องการของผู้ใช้งานทั่วไปและธุรกิจขนาดเล็กถึงกลาง

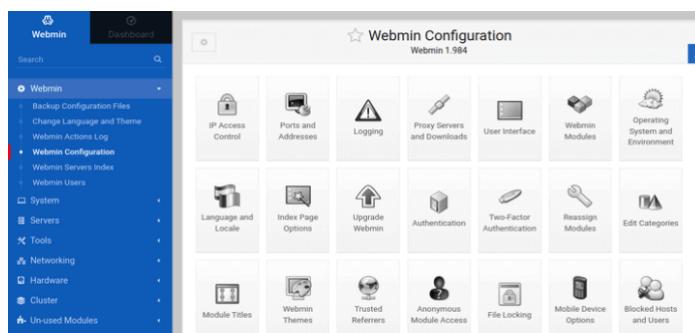
- 4) รองรับหลายภาษา (Multi-Language Support) DirectAdmin รองรับการใช้งานหลายภาษา รวมถึงภาษาไทย ทำให้เข้าถึงกลุ่มผู้ใช้งานในประเทศไทยได้ง่ายขึ้น
- 5) ประสิทธิภาพสูง (Lightweight and Efficient) DirectAdmin มีโครงสร้างที่เบาและไม่ใช้ทรัพยากร ระบบมีการทำงานที่ดีอย่างราบรื่นแม้บนเซิร์ฟเวอร์ที่มีทรัพยากรจำกัด

จุดด้อยของ DirectAdmin

- 1) ฟีเจอร์บางตัวอาจไม่ครบถ้วนเท่ากับ cPanel (Limited Features Compared to cPanel) DirectAdmin มีฟีเจอร์พื้นฐานที่เพียงพอสำหรับการใช้งานทั่วไป แต่เมื่อเปรียบเทียบกับ cPanel อาจพบว่าบางฟีเจอร์ขั้นสูงยังไม่มีหรือมีข้อจำกัดในการใช้งาน
- 2) ตัวเลือกการปรับแต่งน้อยกว่า (Less Customization Options) DirectAdmin อาจมีตัวเลือกการปรับแต่งและส่วนเสริมที่น้อยกว่าเมื่อเปรียบเทียบกับแพลตฟอร์มคู่แข่ง เช่น Plesk หรือ cPanel
- 3) การสนับสนุนจากชุมชนยังจำกัด (Limited Community Support) แม้ว่าจะมีเอกสารช่วยเหลืออย่างเป็นทางการ แต่ชุมชนผู้ใช้งาน DirectAdmin อาจไม่กว้างขวางเท่ากับแพลตฟอร์มอื่น ๆ ซึ่งอาจทำให้การแก้ปัญหาทำได้ช้ากว่า

2.2.4 Webmin

Webmin [29] เป็นซอฟต์แวร์สำหรับบริหารจัดการเซิร์ฟเวอร์ผ่านเว็บเบราว์เซอร์ โดยมีเป้าหมายเพื่ออำนวยความสะดวกให้กับผู้ดูแลระบบในการจัดการและกำหนดค่าต่าง ๆ ของเซิร์ฟเวอร์โดยไม่จำเป็นต้องพิมพ์ Command Line Interface (CLI) โดยตรง Webmin ช่วยลดความซับซ้อนในการดำเนินงาน และเหมาะสมสำหรับห้องผู้ดูแลระบบที่มีประสบการณ์และผู้เริ่มต้น



รูปที่ 2.15 Webmin Control Panel

การทำงานของ Webmin

Webmin ทำงานผ่านโปรโตคอล HTTP หรือ HTTPS บนพอร์ตเริ่มต้น 10000 โดยผู้ใช้สามารถเข้าถึงและบริหารเซิร์ฟเวอร์จากรายละเอียดผ่านเว็บเบราว์เซอร์ โดย Webmin สามารถติดตั้งและทำงานบนระบบปฏิบัติการ Unix/Linux เป็นหลัก แต่ยังสามารถใช้งานบนระบบปฏิบัติการ Windows ได้ด้วยฟีเจอร์ที่รองรับในบางกรณี

Webmin ประกอบด้วยโครงสร้างการบริหารจัดการที่แยกเป็นโมดูลต่าง ๆ ทำให้ผู้ใช้สามารถควบคุมและกำหนดค่าการทำงานของเซิร์ฟเวอร์ได้ในลักษณะเฉพาะด้าน เช่น การจัดการบัญชีผู้ใช้ การบริหารฐานข้อมูล และการตั้งค่าเครือข่าย โดยไม่ต้องแก้ไขไฟล์คอนฟิกเรซั่นด้วยตนเอง

ความสามารถของ Webmin

Webmin มีฟีเจอร์ที่หลากหลาย ครอบคลุมการบริหารจัดการเซิร์ฟเวอร์ในหลายด้าน ดังนี้

1) การจัดการผู้ใช้และกลุ่ม (User and Group Management)

- สามารถสร้าง ลบ และแก้ไขบัญชีผู้ใช้ รวมถึงการกำหนดสิทธิ์การเข้าถึง
- รองรับการตั้งค่ารหัสผ่านและการกำหนดชีดจำกัตทรัพยากรของผู้ใช้

2) การจัดการฐานข้อมูล (Database Management)

- รองรับระบบฐานข้อมูล MySQL และ PostgreSQL โดยสามารถสร้าง แก้ไข และสำรองข้อมูลฐานข้อมูลได้จากยินเทอร์เฟช

3) การตั้งค่าเครือข่าย (Network Configuration)

- สามารถกำหนดค่า IP Address, DNS, Gateway และ Firewall ผ่าน Webmin เพื่อเพิ่มความปลอดภัยและความยืดหยุ่นให้กับเครือข่าย

4) การจัดการเว็บเซิร์ฟเวอร์ (Web Server Management)

- รองรับการบริหารจัดการเซิร์ฟเวอร์เว็บยอดนิยม เช่น Apache และ Nginx รวมถึงการตั้งค่า Virtual Host และการกำหนดค่าการเข้าถึงไฟล์ต่าง ๆ

5) การบริหารระบบไฟล์และคอนฟิกเรซั่น (File System and Configuration Management)

- ผู้ใช้สามารถเข้าถึงและแก้ไขไฟล์คอนฟิกเรซั่นเซิร์ฟเวอร์ได้โดยตรงผ่าน Webmin โดยไม่ต้องใช้คำสั่งใน CLI

6) ระบบสำรองและกู้คืนข้อมูล (Backup and Restore System)

- รองรับการสำรองข้อมูลของระบบและบริการสำคัญ เช่น ฐานข้อมูลและไฟล์ระบบ

7) โมดูลที่หลากหลาย (Extensive Modules)

- Webmin มีโมดูลมากมายสำหรับการจัดการบริการต่าง ๆ เช่น BIND DNS, Postfix, และ FTP Server ช่วยให้สามารถบริหารบริการได้จากส่วนกลาง

จุดเด่นของ Webmin

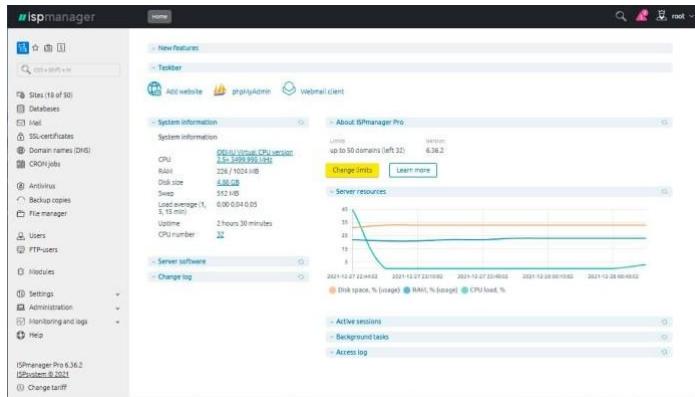
- ใช้งานง่าย (User-Friendly Interface) อินเทอร์เฟซที่เรียบง่ายและเข้าใจง่าย ช่วยลดความซับซ้อนของการบริหารเซิร์ฟเวอร์ และเหมาะสมสำหรับผู้เริ่มต้น
- เป็นซอฟต์แวร์โอเพนซอร์ส (Open Source Software) Webmin ให้บริการฟรีภายใต้สัญญาอนุญาตแบบ BSD ซึ่งช่วยลดต้นทุนในการบริหารจัดการเซิร์ฟเวอร์
- ความยืดหยุ่นในการปรับแต่ง (High Flexibility) สามารถติดตั้งโมดูลเพิ่มเติมและปรับแต่งให้เหมาะสมกับความต้องการเฉพาะด้านขององค์กรได้
- รองรับหลายระบบปฏิบัติการ (Cross-Platform Compatibility) รองรับการทำงานบนระบบปฏิบัติการ Unix/Linux เป็นหลัก และสามารถปรับใช้บน Windows ได้
- การจัดการระยะไกล (Remote Management) ผู้ดูแลระบบสามารถเข้าถึงและจัดการเซิร์ฟเวอร์ได้จากทุกที่ผ่านเว็บเบราว์เซอร์

จุดด้อยของ Webmin

- ความเสี่ยงด้านความปลอดภัย (Security Risks) การตั้งค่า Webmin ที่ไม่เหมาะสมอาจทำให้เกิดช่องโหว่ด้านความปลอดภัย เนื่องจาก Webmin ทำงานในระดับสิทธิ์ root ซึ่งอาจถูกโจมตีได้หากไม่มีการตั้งค่าการเข้าถึงที่เหมาะสม
- ความซับซ้อนสำหรับผู้เริ่มต้น (Complexity for Beginners) แม้ว่าอินเทอร์เฟซจะใช้งานง่ายแต่สำหรับผู้ที่ไม่คุ้นเคยกับระบบ Unix/Linux อาจต้องเรียนรู้แนวทางการตั้งค่าเพิ่มเติม
- ความต้องการโมดูลเพิ่มเติม (Module Dependency) การใช้งาน Webmin ให้มีประสิทธิภาพสูงสุดในบางกรณีอาจต้องติดตั้งโมดูลเสริมเพิ่มเติม ซึ่งอาจเพิ่มความยุ่งยากสำหรับผู้ใช้ใหม่
- ข้อจำกัดด้านการสนับสนุน (Limited Official Support) แม้ว่า Webmin จะมีชุมชนผู้ใช้งานที่ช่วยเหลือกัน แต่การสนับสนุนอย่างเป็นทางการอาจไม่ครอบคลุมทุกปัญหาและต้องอาศัยความรู้จากเอกสารหรือชุมชนออนไลน์

2.2.5 ISPConfig

ISPConfig [30] เป็นซอฟต์แวร์ควบคุมการจัดการเว็บไซต์แบบโอลูนชอร์สที่ให้บริการโดยไม่มีค่าใช้จ่าย ช่วยให้ผู้ดูแลระบบสามารถจัดการเซิร์ฟเวอร์และบริการต่าง ๆ ได้จากศูนย์กลางเดียว รองรับการจัดการเซิร์ฟเวอร์หลายเครื่องพร้อมกัน และมีคุณสมบัติที่ครอบคลุมความต้องการของผู้ให้บริการเว็บไซต์และองค์กรที่ต้องการบริหารระบบเซิร์ฟเวอร์อย่างมีประสิทธิภาพ



รูปที่ 2.16 ISPConfig Control Panel

การทำงานของ ISPConfig

ISPConfig ทำงานผ่านอินเทอร์เฟซบนเว็บที่ช่วยให้ผู้ใช้สามารถเข้าถึงและจัดการเซิร์ฟเวอร์จากระยะไกล รองรับการจัดการเซิร์ฟเวอร์หลายเครื่องจากแงบคุมเดียว (multi-server environment) ทำให้เหมาะสมสำหรับผู้ให้บริการไซต์ที่ต้องการบริหารจัดการหลายเซิร์ฟเวอร์พร้อมกัน ISPConfig รองรับระบบปฏิบัติการ Linux หลายรุ่น เช่น Debian, Ubuntu, และ CentOS ซึ่งสามารถติดตั้งและใช้งานได้บนระบบที่รองรับซอฟต์แวร์โอลูนชอร์สต่าง ๆ เช่น Apache, Nginx, MySQL/MariaDB และ Postfix

ความสามารถของ ISPConfig

ISPConfig มีฟังก์ชันการทำงานที่หลากหลายเพื่อรองรับการจัดการบริการบนเซิร์ฟเวอร์ ดังนี้

1) การจัดการโดเมน (Domain Management)

- รองรับการเพิ่ม ลบ และแก้ไขโดเมนและชับโดเมน
- สามารถจัดการการตั้งค่า DNS และตั้งค่าไซต์แบบเสมือน (Virtual Hosts)

2) การจัดการผู้ใช้ (User Management)

- รองรับการสร้างและบริหารบัญชีผู้ใช้ด้วยลิทธีการเข้าถึงที่กำหนดได้
- สามารถกำหนดระดับผู้ใช้ เช่น ผู้ดูแลระบบ (Administrator) ผู้ให้บริการไฮสติง (Reseller) และผู้ใช้ทั่วไป (Client)

3) การบริหารอีเมล (Email Services Management)

- รองรับการตั้งค่าและจัดการบัญชีอีเมลผ่าน Postfix และ Dovecot
- สนับสนุนโปรโตคอล SMTP, POP3 และ IMAP รวมถึงการตั้งค่าการป้องกันสแปมและไวรัส

4) การจัดการฐานข้อมูล (Database Management)

- รองรับระบบฐานข้อมูล MySQL และ MariaDB
- ให้ลิทธีผู้ใช้ในการเข้าถึงฐานข้อมูลผ่าน phpMyAdmin

5) การจัดการบัญชี FTP (FTP Management)

- สามารถสร้างและควบคุมบัญชี FTP เพื่อการอัปโหลดและดาวน์โหลดไฟล์ระหว่างเซิร์ฟเวอร์และผู้ใช้

6) การตั้งค่าไฟร์วอลล์และ DNS (Firewall and DNS Management)

- รองรับการกำหนดค่า Firewall และการจัดการ DNS Records เช่น A, CNAME, MX และ TXT

7) การสำรองข้อมูลและการกู้คืน (Backup and Restore)

- มีฟังก์ชันสำรองข้อมูลอัตโนมัติและการกู้คืนระบบในกรณีฉุกเฉิน

8) การรองรับหลายเซิร์ฟเวอร์ (Multi-Server Support)

- ISPCConfig สามารถบริหารจัดการหลายเซิร์ฟเวอร์จากแพลตฟอร์มเดียว ทำให้สะดวกในการบริหารคุณภาพข้อมูลขนาดใหญ่

จุดเด่นของ ISPCConfig

- เป็นโอเพนซอร์สและไม่มีค่าใช้จ่าย (Open Source and Free of Charge) สามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่ายเพิ่มเติม ลดต้นทุนในการบริหารจัดการเว็บไฮสติง
- รองรับหลายภาษา (Multi-Language Support) มีอินเทอร์เฟซที่รองรับหลายภาษา รวมถึงภาษาไทย ช่วยให้ผู้ใช้สามารถเข้าใจและใช้งานระบบได้ง่ายขึ้น

- 3) ความยืดหยุ่นและขยายชีดความสามารถได้สูง (Highly Flexible and Scalable) สามารถปรับแต่งและขยายฟังก์ชันการทำงานเพิ่มเติมเพื่อรองรับความต้องการขององค์กรที่มีความซับซ้อน
- 4) รองรับหลายระบบปฏิบัติการ (Compatible with Various Linux Distributions) สามารถติดตั้งบนระบบปฏิบัติการ Linux หลายรูปแบบ เช่น Debian, Ubuntu และ CentOS
- 5) การสนับสนุนจากชุมชน (Large Community Support) มีเอกสาร คู่มือ และฟอรั่มสนับสนุนที่ช่วยแก้ไขปัญหาการใช้งานอย่างกว้างขวาง

จุดด้อยของ ISPConfig

- 1) อินเทอร์เฟซผู้ใช้งานไม่ใช้งานง่ายเท่ากับ cPanel หรือ Plesk (Less Intuitive Interface) อินเทอร์เฟซของ ISPConfig อาจดูซับซ้อนสำหรับผู้ใช้มือใหม่เมื่อเปรียบเทียบกับเครื่องมือที่มีค่าใช้จ่ายอย่าง cPanel และ Plesk
 - 2) ฟีเจอร์บางส่วนอาจมีข้อจำกัด (Limited Features Compared to Commercial Solutions) แม้ว่า ISPConfig จะมีฟีเจอร์พื้นฐานที่ครบถ้วน แต่บางฟีเจอร์ขั้นสูง เช่น ระบบบูรณาการแบบอัตโนมัติอาจไม่รองรับเต็มรูปแบบ
 - 3) ต้องมีความรู้ด้าน Linux (Requires Linux Expertise) การติดตั้งและตั้งค่า ISPConfig อาจต้องอาศัยความรู้เกี่ยวกับระบบปฏิบัติการ Linux และการบริหารเซิร์ฟเวอร์
 - 4) การตั้งค่าเริ่มต้นอาจซับซ้อน (Complex Initial Setup) การติดตั้งและกำหนดค่าครั้งแรกอาจใช้เวลาและต้องการความเข้าใจเกี่ยวกับโครงสร้างของซอฟต์แวร์
- จากการที่เกี่ยวข้องดังที่กล่าวมา สามารถนำจุดเด่นของแต่ละงาน เสริมกับจุดด้อยที่ควรจะเกิดขึ้น ทำให้เกิดงาน ADDMS โดยการนำจุดเด่นของระบบอื่นมาประยุกต์ใช้และพัฒนาจุดด้อยที่มีอยู่ จะทำให้ ADDMS เป็นระบบที่มีความสมบูรณ์มากขึ้น พร้อมทั้งรักษาจุดเด่นด้านการรองรับเทคโนโลยีล่าสุดใหม่อย่าง GoFiber ที่เป็นจุดแข็งสำคัญของระบบ แสดงข้อมูลตั้งต่างๆ ที่ 2.1 และต่างๆ ที่ 2.2

ตารางที่ 2.1 ตารางเปรียบเทียบความสามารถของงานที่เกี่ยวข้องและโปรเจกต์ ADDMS

คุณสมบัติ/ Control panel	cPanel	Plesk	DirectAdmin	Webmin	ISPConfig	ADDMS
ประเภท	เชิงพาณิชย์	เชิงพาณิชย์	เชิงพาณิชย์	โอเพนซอร์ส	โอเพนซอร์ส	โอเพนซอร์ส
ระบบปฏิบัติการ ที่รองรับ	Linux	Windows, Linux	Linux	Unix, Linux, Windows (บาง พักร์ชั่น)	Linux	Linux
การจัดการ โดเมน	✓	✓	✓	✓	✓	✓
การจัดการ ฐานข้อมูล	MySQL, PostgreSQL	MySQL, PostgreSQL	MySQL, PostgreSQL	MySQL, PostgreSQL	MySQL, MariaDB	MySQL, MariaDB
การจัดการอีเมล	✓	✓	✓	✓	✓	✓
ความสามารถในการติดตั้งแอป พลัตฟอร์ม	ใช้งานง่าย (Softaculous)	ใช้งานง่าย (Plesk Installer)	ใช้งานง่าย (Installatron)	ต้องติดตั้งด้วย ตนเอง	ต้องติดตั้งด้วย ตนเอง	สามารถทำงาน ผ่านเว็บไซต์
การสนับสนุน SSL	ฟรี SSL ผ่าน AutoSSL	ฟรี SSL ผ่าน Let's Encrypt	ฟรี SSL ผ่าน Let's Encrypt	ต้องติดตั้งด้วย ตนเอง	ต้องติดตั้งด้วย ตนเอง	ฟรี SSL ผ่าน Let's Encrypt
การจัดการผู้ใช้	✓	✓	✓	✓	✓	✓

ตารางที่ 2.1 ตารางเปรียบเทียบความสามารถของงานที่เกี่ยวข้องและโปรเจกต์ ADDMS (ต่อ)

คุณสมบัติ/ Control panel	cPanel	Plesk	DirectAdmin	Webmin	ISPConfig	ADDMS
ความสามารถในการสำรองข้อมูล	✓	✓	✓	✓	✓	✗
อินเทอร์เฟซผู้ใช้	ใช้งานง่ายและเป็นมิตรกับผู้ใช้	ใช้งานง่ายและสวยงาม	ใช้งานง่ายแต่เรียบง่าย	ไม่ค่อยเป็นมิตรกับผู้ใช้แต่ปรับแต่งได้สูง	อินเทอร์เฟซที่เข้าใจง่ายแต่ไม่สวยงามเท่า	ใช้งานง่ายแต่เรียบง่าย

ตารางที่ 2.2 ตารางเปรียบเทียบภาษาที่รองรับของงานที่เกี่ยวข้องและโปรเจกต์ ADDMS

คุณสมบัติ	ISPConfig	cPanel	Plesk	DirectAdmin	Webmin	ADDMS
PHP Support						
Multiple PHP Versions	✓	✓	✓	✓	✓	✓
Easy PHP Configuration	✓	✓	✓	✓	!	✓
PHP Extensions Manager	!	✓	✓	!	!	✓
Node.js Support						
Built-in Support	✗	✓	✓	✗	✗	✓
Easy Installation	!	✓	✓	!	!	✓
Version Manager	✗	✓	✓	✗	✗	✓
Go/Gofiber Support						
Native Support	✗	✗	✗	✗	!	✓
Manual Installation	!	!	!	!	✓	✓
Static HTML						
File Manager	✓	✓	✓	✓	✓	✓
FTP Support	✓	✓	✓	✓	✓	✓
General Features						
GUI Interface	✓	✓	✓	✓	!	✓
Easy to Use	!	✓	✓	✓	!	✓
Customization	✓	!	✓	!	✓	✓
Auto-installer Scripts	!	✓	✓	✓	✗	✓

คำอธิบายสัญลักษณ์

✓ = รองรับเต็มรูปแบบ/ใช้งานง่าย

! = รองรับบางส่วน/ต้องตั้งค่าเพิ่มเติม

✗ = ไม่รองรับ/ต้องติดตั้งเอง

บทที่ 3

วิธีการดำเนินงาน

3.1 การวิเคราะห์ความต้องการของระบบ

การวิเคราะห์ความต้องการของระบบเป็นขั้นตอนสำคัญในการพัฒนาเว็บคอนโทรลพาแนลในรูปแบบเว็บแอปพลิเคชันที่ลดขั้นตอนและความซับซ้อนในการสร้างเว็บไซต์ ในขั้นตอนนี้ คณะกรรมการตัดทำได้ดำเนินการศึกษาและรับทราบข้อมูลจากแหล่งต่าง ๆ เพื่อให้เข้าใจความต้องการของผู้ใช้และผู้มีส่วนได้ส่วนเสียอย่างครอบคลุม การวิเคราะห์นี้จะช่วยให้สามารถออกแบบและสร้างเว็บแอปพลิเคชันที่ตอบสนองความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพคำนึงการวิเคราะห์นี้จะช่วยให้สามารถออกแบบและพัฒนาระบบที่ตอบสนองความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพ คำนึงถึงความต้องการที่หลากหลายของผู้ใช้ในแต่ละกลุ่มตั้งแต่ผู้ใช้ที่มีความรู้พื้นฐานจนถึงมืออาชีพในการสร้างเว็บไซต์ ในหัวข้ออยู่ต่อไปนี้จะนำเสนอหัวข้อย่อยต่อไปนี้จะนำเสนอผลการวิเคราะห์ทั้งผู้ใช้งานรวมถึงพัฒนาระบบที่ได้กำหนดขึ้นจากการวิเคราะห์นี้

3.1.1 การกำหนดพัฒนาการทำงานหลักของระบบ

จากการวิเคราะห์ความต้องการของผู้ใช้และผู้มีส่วนได้ส่วนเสียคณะกรรมการตัดทำได้กำหนดจากการวิเคราะห์ความต้องการของผู้ใช้และผู้มีส่วนได้ส่วนเสียคณะผู้จัดทำได้กำหนดพัฒนาการทำงานหลักของเว็บแอปพลิเคชันที่เป็นเครื่องมือช่วยสร้างโครงสร้างพื้นฐานที่จะขับเคลื่อนและความซับซ้อนในการสร้างเว็บไซต์ดังนี้

1) การจัดการผู้ใช้และสิทธิ์

- ผู้ใช้สามารถสร้างบัญชีใหม่และเข้าสู่ระบบได้
- การกำหนดการกำหนดบทบาทและสิทธิ์การเข้าถึงระบบสามารถกำหนดบทบาทต่าง ๆ เช่นผู้ดูแลระบบผู้ใช้งานระบบโดยแต่ละบทบาทจะมีสิทธิ์การเข้าถึงและการดำเนินการที่ต่างกัน

2) การสร้างเว็บไซต์

- พัฒนาระบบที่เป็นการช่วยผู้ใช้งานสร้างเว็บไซต์ได้อย่างรวดเร็วทั้งการเลือกเฟรมเวิร์กและโปรแกรมภาษาต่างๆ
- ผู้ใช้สามารถสร้างเว็บไซต์ได้โดยไม่ต้องมีความรู้ด้านการกำหนดข้อมูลและระบบ

3) การจัดการไฟล์และโฟลเดอร์

- พัฒนาการจัดการไฟล์จะต้องรองรับไฟล์ที่เกี่ยวข้องกับการพัฒนาเว็บในเทคโนโลยีต่าง ๆ ทั้งไฟล์ HTML, CSS, JavaScript, รวมถึงไฟล์ที่เกี่ยวข้องกับ Node.js, Go และ PHP
- เพิ่มพัฒนาการจัดการไฟล์ที่รองรับการอัปโหลดไฟล์ที่เป็นส่วนประกอบของแอปพลิเคชัน และส่งออกไฟล์เป็นอัตโนมัติ
- พัฒนานี้ช่วยให้ผู้ใช้จัดการเว็บไซต์ทั้งการเพิ่ม แก้ไข ลบ ได้ตลอดเวลา

4) การติดตั้งซอฟต์แวร์และส่วนเซอร์วิส

- รองรับการติดตั้งส่วนเซอร์วิสต่างๆ เช่น MySQL 1-2 ชั้นตอน โดยไม่ต้องมีการตั้งค่าซับซ้อน
- พัฒนานี้จะช่วยให้ผู้ใช้สามารถเพิ่มฟีเจอร์ที่ต้องการได้อย่างง่ายดายผ่านการติดตั้งส่วนเซอร์วิสที่รองรับทั้ง Node.js, Go, PHP
- ผู้ใช้สามารถเลือกจากรายการส่วนเซอร์วิสที่เหมาะสมสำหรับเว็บไซต์ที่พัฒนาในแต่ละเทคโนโลยีได้ทันที

5) การจัดการโดเมน

- เพิ่มการรองรับโดเมนที่ใช้ในเว็บแอปพลิเคชันที่พัฒนาโดย HTML, CSS, JavaScript, Node.js, Go, PHP เพื่อให้ง่ายต่อการเชื่อมต่อกับเว็บไซต์ที่พัฒนาขึ้น
- รองรับการตั้งค่าโดเมนหลายตัวจากแอปพลิเคชันเดียวในระบบ

6) การตั้งค่าเว็บเซิร์ฟเวอร์

- รองรับการตั้งค่าเว็บเซิร์ฟเวอร์ที่รองรับเว็บไซต์ที่พัฒนาโดย HTML, CSS, JavaScript, Node.js, Go, PHP
- เพิ่มพัฒนาการตั้งค่าเซิร์ฟเวอร์ที่สามารถรองรับทั้ง Static และ Dynamic content ที่เขียนด้วยเทคโนโลยีเหล่านี้

3.2 การออกแบบสถาปัตยกรรมระบบ

ในการพัฒนาเว็บแอปพลิเคชัน การออกแบบระบบถือเป็นขั้นตอนที่มีความสำคัญอย่างยิ่ง เนื่องจากเป็นการวางแผนสำหรับการพัฒนาทั้งระบบ การออกแบบที่ดีจะช่วยให้การพัฒนาเป็นไปอย่างมีประสิทธิภาพ ลดความซับซ้อนและง่ายต่อการบำรุงรักษาในอนาคต ในบทนี้จะอธิบายถึงการออกแบบระบบในด้านต่าง ๆ ประกอบด้วย แผนภาพการออกแบบ พังก์ชันการทำงานหลักที่แสดงให้เห็นภาพรวมของการทำงานทั้งระบบ ความสัมพันธ์และการเชื่อมโยงระหว่างส่วนประกอบต่าง ๆ การออกแบบส่วนหน้าบ้าน การออกแบบหลังบ้าน และการออกแบบฐานข้อมูล การออกแบบในแต่ละส่วนนี้จะต้องคำนึงถึงความสอดคล้องและการทำงานร่วมกันอย่างมีประสิทธิภาพ เพื่อให้ได้ระบบที่มีความสมบูรณ์และตอบสนองความต้องการของผู้ใช้งานได้อย่างดีที่สุด

3.2.1 แผนภาพการออกแบบพังก์ชันการในด้านของผู้ใช้งานและด้านของผู้ดูแลระบบ

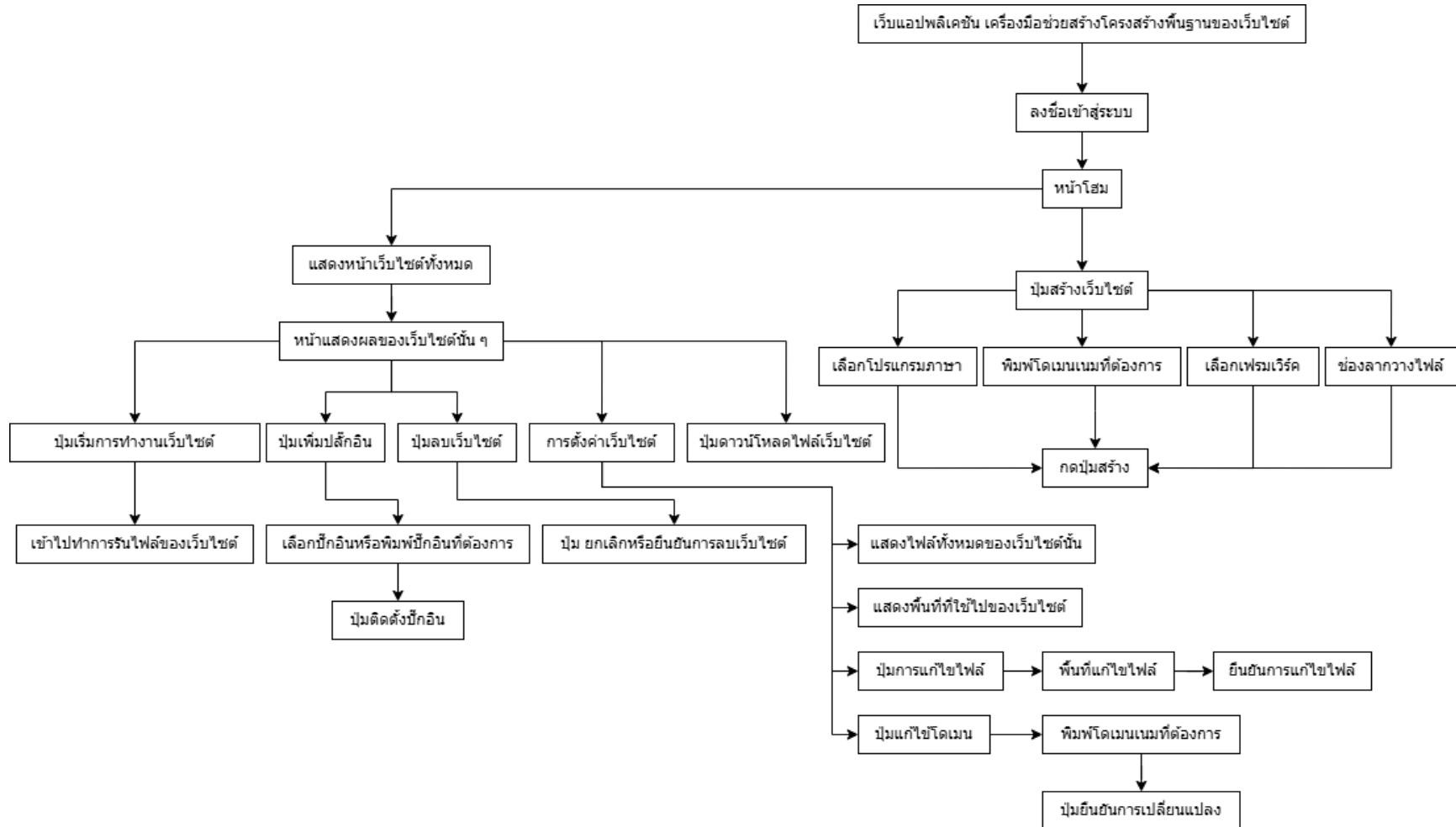
แผนภาพการออกแบบพังก์ชันการทำงานหลักของผู้ใช้งานเว็บแอปพลิเคชันสำหรับแสดงเว็บคอนโทรลพาเนลในรูปแบบเว็บแอปพลิเคชัน สำหรับสร้างสิ่งแวดล้อมในการพัฒนาเว็บไซต์ ประกอบด้วย 6 ส่วนสำคัญ ได้แก่ ส่วนของหน้าโฮมเพจ หน้าเข้าสู่ระบบ หน้าการสร้างเว็บไซต์ หน้าแสดงเว็บไซต์ของผู้ใช้งานทั้งหมด หน้าการตั้งค่าเว็บไซต์ และหน้าเพิ่มส่วนเสริม ดังรูปที่ 3.1 โดยมีรายละเอียดดังต่อไปนี้

แผนภาพแสดงการทำงานของเว็บแอปพลิเคชัน โดยเริ่มจากการลงชื่อเข้าใช้งานซึ่งผู้ใช้งานสามารถเลือกดำเนินการได้ 6 ส่วนหลัก หากมีเว็บไซต์อยู่ก่อนหน้าแล้ว ในกรณีที่ไม่มีเว็บไซต์อยู่ก่อนหน้า ผู้ใช้สามารถทำได้โดยการเริ่มต้นสร้างเว็บไซต์เพียงเท่านั้น โดยเริ่มจากเลือกโปรแกรมภาษาเพรอมเวิร์ก และตั้งชื่อโดเมนของเว็บไซต์ หรืออาจทำการลากวางไฟล์ของเว็บไซต์ได้ ในกรณีที่มีเว็บไซต์อยู่แล้ว ผู้ใช้สามารถเลือกดำเนินขั้นตอนที่เหลือได้ตามต้องการ เช่น เริ่มต้นการทำงานเว็บไซต์ การแก้ไขเว็บไซต์ การดาวน์โหลดไฟล์เว็บไซต์ การสร้างเว็บไซต์เพิ่มเติม (กรณีพื้นที่ของผู้ใช้งานยังคงเหลือจากการซึ่งกันและกัน) การเพิ่มส่วนเสริม และการลบเว็บไซต์ที่ไม่ต้องการ

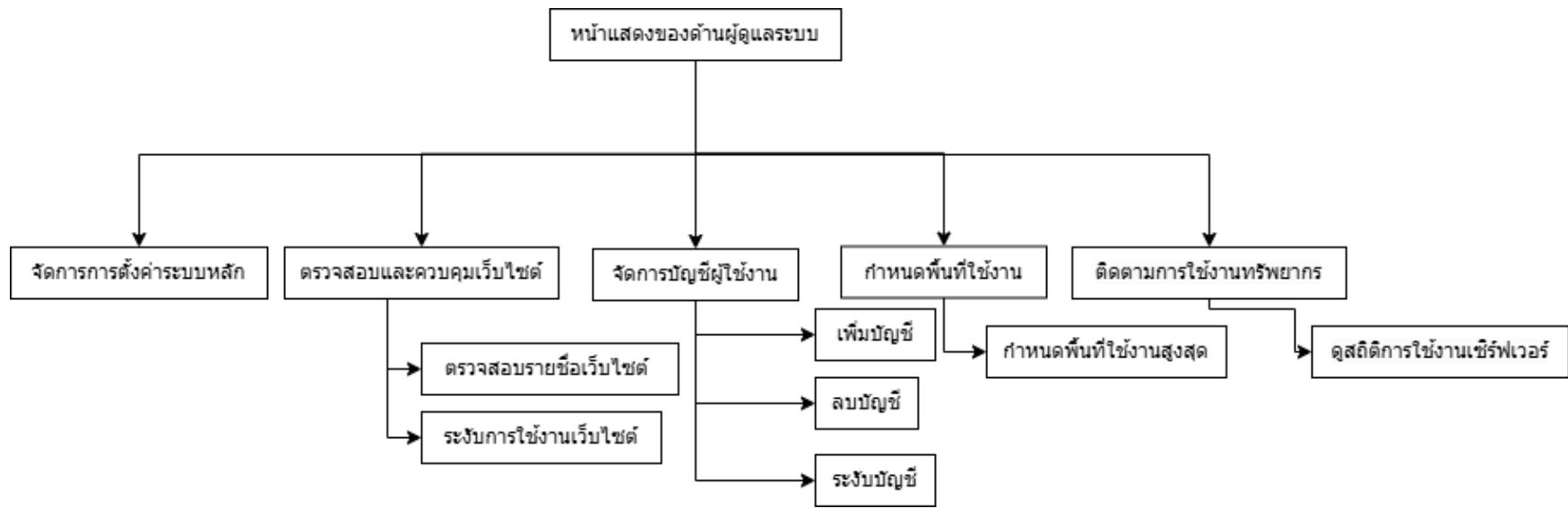
ในส่วนของ ผู้ดูแลระบบ จะสามารถจัดการผู้ใช้งานได้ทั้งการจัดการเว็บไซต์ของผู้ใช้งาน การกำหนดพื้นที่ที่สามารถใช้งานได้ และการจัดสรรทรัพยากรของระบบให้เหมาะสม โดยผู้ดูแลระบบสามารถดำเนินการ กำหนดพื้นที่สูงสุดที่ผู้ใช้แต่ละคนสามารถใช้งานได้ ตรวจสอบและจัดการเว็บไซต์ ของผู้ใช้งาน เช่น รับรองการใช้งานเว็บไซต์ หรืออนุญาตให้เพิ่มโดเมน ติดตามการใช้งานทรัพยากร

เชิร์ฟเวอร์และบริหารจัดการการตั้งค่าระบบหลัก ลบหรือรังับัญชีผู้ใช้งานที่ลงทะเบียนข้อมูล
ดังแสดงในรูปที่ 3.2 โดยมีรายละเอียดดังต่อไปนี้

แผนภาพการทำงานของผู้ดูแลระบบประกอบด้วย 5 ส่วนหลัก ได้แก่ การจัดการบัญชีผู้ใช้งาน
(เพิ่ม ลบ หรือรังับัญชีที่ลงทะเบียนข้อมูล) การกำหนดพื้นที่การใช้งาน (กำหนดพื้นที่สูงสุดตาม
ข้อกำหนดของระบบ) การตรวจสอบและควบคุมเว็บไซต์ (รังับการใช้งาน อนุญาตเพิ่มเติม หรือ
บังคับลบเว็บไซต์ที่ลงทะเบียนเข้า) การจัดการการตั้งค่าระบบหลัก และการติดตามการใช้งาน
ทรัพยากร



รูปที่ 3.1 แผนภาพการออกแบบพื้นที่ชั้นการทำงานหลักในด้านของผู้ใช้งาน



รูปที่ 3.2 แผนภาพการออกแบบแบบฟังก์ชันการทำงานของผู้ดูแลระบบ

3.2.2 แผนภาพรวมของระบบ.

แผนภาพนี้แสดงการทำงานของระบบเว็บแอปพลิเคชันสำหรับการแสดงผลข้อมูลเว็บไซต์ให้กับผู้ใช้งานโดยแบ่งได้ตามส่วนประกอบดังนี้

1) กระบวนการเริ่มต้น (HTTP)

ผู้ใช้งานเริ่มต้นการทำงานของระบบด้วยการส่งคำขอผ่านโปรโตคอล HTTP ซึ่งอาจเกิดจากการเข้าถึงเว็บไซต์ หรือทำงานผ่าน API หลังจากนั้นคำขอเหล่านี้จะถูกส่งไปยัง Nginx ซึ่งเป็นตัวกลางที่ทำหน้าที่สำคัญในการกระจายคำขอไปยังปลายทางที่เหมาะสม

2) บทบาทของ Nginx

Nginx ทำหน้าที่เป็น Reverse Proxy Server โดยมีหน้าที่หลักในการจัดการและกระจายคำขอ HTTP ไปยังระบบต่าง ๆ ที่เหมาะสม เช่น หากคำขอเกี่ยวข้องกับการแสดงผลส่วนหน้าของเว็บไซต์ คำขอจะถูกส่งไปยัง Next.js หากเป็นคำขอสำหรับ API หรือการทำงานของ Backend คำขอจะถูกส่งไปยัง Go Fiber หากคำขอต้องการทำงานที่เกี่ยวข้องกับ PHP หรือ Node.js Nginx จะส่งต่อคำขอไปยังระบบที่รองรับภาษาเหล่านั้น Nginx ยังช่วยในด้านประสิทธิภาพการทำงานของระบบ เช่น การแคชข้อมูลและการลดภาระโหลดเซิร์ฟเวอร์

3) ส่วนหน้าของระบบ (Frontend)

ส่วนหน้าบ้านหรือ Frontend ของระบบถูกพัฒนาโดยใช้ Next.js ซึ่งเป็น Framework ของ React.js ที่ช่วยในการสร้างส่วนติดต่อผู้ใช้ (User Interface) ที่มีความรวดเร็วและทันสมัย Next.js ทำงานร่วมกับเทคโนโลยีพื้นฐาน เช่น HTML, CSS, และ JavaScript เพื่อแสดงผลหน้าเว็บให้แก่ผู้ใช้งาน ส่วนหน้าบ้านนี้มีหน้าที่ในการรับคำสั่งจากผู้ใช้และส่งข้อมูลคำขอไปยัง Backend ผ่าน API

4) ส่วนหลังของระบบ (Backend)

ระบบหลังบ้านถูกพัฒนาโดยใช้ Go Fiber ซึ่งเป็น Framework ของภาษา Go ที่มีความเร็วและเหมาะสมสำหรับการสร้าง API Fiber จะรับคำขอจาก Nginx และประมวลผลคำขอ เช่น การจัดการข้อมูล การคำนวณ หรือการตรวจสอบลิขสิทธิ์ Fiber เชื่อมต่อรับฐานข้อมูล MariaDB ผ่าน GORM (Go Object-Relational Mapping) เพื่อจัดการข้อมูล เช่น การเพิ่ม ลบ แก้ไข และเรียกดูข้อมูล ผลลัพธ์ของการประมวลผลจะถูกส่งกลับไปยังผู้ใช้งานหรือระบบ Frontend ผ่าน Nginx

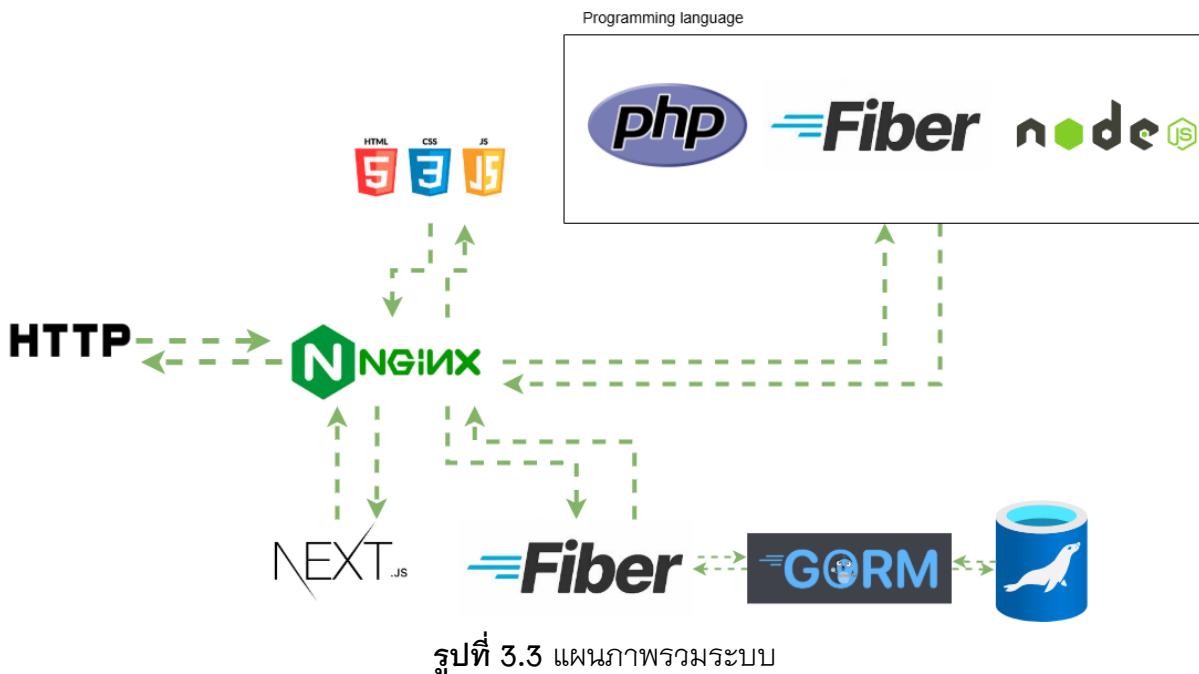
5) PHP และ Node.js

นอกจาก Go Fiber ระบบยังรองรับการทำงานของ PHP และ Node.js เพื่อรองรับคำขอหรือฟีเจอร์เฉพาะบางอย่างที่พัฒนาด้วยภาษาเหล่านี้ PHP มากใช้สำหรับแอปพลิเคชันหรือส่วนที่ยังคงใช้งานระบบเดิม Node.js

6) ฐานข้อมูล (MariaDB)

ระบบจัดเก็บข้อมูลทั้งหมดไว้ใน MariaDB ซึ่งเป็นระบบจัดการฐานข้อมูลแบบ Open Source ที่มีประสิทธิภาพและรองรับการขยายตัวในอนาคต Fiber จะเชื่อมต่อกับ MariaDB ผ่าน GORM ซึ่งช่วยให้การเขียนและอ่านข้อมูลในฐานข้อมูลเป็นไปได้อย่างสะดวกและเป็นระเบียบ MariaDB มีความสามารถในการจัดการข้อมูลจำนวนมากอย่างมีประสิทธิภาพ

ข้างต้นเป็นการทำงานตามส่วนประกอบของการทำงานหลักที่ช่วยทำให้เห็นการทำงานของเว็บแอปพลิเคชัน เครื่องมือช่วยสร้างโครงสร้างพื้นของเว็บไซต์ได้เป็นอย่างดี และแสดงแผนภาพรวมดังรูปที่ 3.3



3.3 การออกแบบแผนภาระบบ

การออกแบบแพนกว่าเป็นส่วนสำคัญของการสร้างระบบเพื่อแสดงกระบวนการทำงานทั้งหมด ในส่วนนี้ ผู้จัดทำได้นำเสนอแพนกว่า 2 ประเภท ได้แก่ แพนกวากิจกรรม (Activity Diagram) และ แพนกวายูสเคส (Use Case Diagram) เพื่อแสดงรายละเอียดการทำงานและความสัมพันธ์ขององค์ประกอบในระบบ โดยแต่ละประเภทมีรายละเอียดดังต่อไปนี้

3.3.1 แผนภาพกิจกรรม (Active Diagram)

แผนภาพกิจกรรมใช้ในการแสดงลำดับขั้นตอนการทำงานของระบบอย่างชัดเจนและเข้าใจง่าย โดยแสดงให้เห็นถึงการเชื่อมโยงระหว่างกิจกรรมต่าง ๆ ภายในกระบวนการทำงานของระบบ รวมถึง จุดเริ่มต้นและสิ้นสุดของการบวนการ นอกจากนี้ แผนภาพกิจกรรมยังช่วยในการวิเคราะห์และ ออกแบบระบบ ทำให้สามารถระบุปัญหาหรือข้อปรับปรุงในขั้นตอนได้อย่างแม่นยำ ถ้าทั้งยังใช้สื่อสาร กับทีมงานหรือผู้เกี่ยวข้องเพื่อให้มีความเข้าใจที่ตรงกันเกี่ยวกับกระบวนการทำงานของระบบในหัวข้อนี้ ผู้จัดทำได้นำเสนอแผนภาพกิจกรรมของระบบสามารถแบ่งออกเป็น 4 กระบวนการ ดังนี้

3.3.1.1 แผนภาพของกิจกรรมของหน้าเข้าสู่ระบบ

กระบวนการเข้าสู่ระบบต้องใช้ อีเมล (email) และรหัสผ่านที่ได้ลงทะเบียนไว้ก่อนหน้าแล้ว โดยรายชื่นตอนการทำงานแสดงดังรูปที่ 3.4

- 1) ผู้ใช้งานเข้าสู่ระบบโดยการกรอก อีเมลและรหัสผ่าน
 - 2) ระบบจะนำไปเช็คข้อมูลในฐานข้อมูล
 - 3) ตรวจสอบ อีเมลและรหัสผ่าน ในฐานข้อมูล ซึ่งจะแบ่งออกเป็น 2 กรณีดังนี้
 - 3.1) กรณีที่ตรงกับฐานข้อมูลทั้ง อีเมลและรหัสผ่านของกันและกัน
 - 3.2) กรณีที่ไม่พบข้อมูลจะแจ้งเตือนว่า “invalid user or password” และกับไปสู่หน้าเข้าสู่ระบบใหม่อีกครั้ง
 - 4) ตรวจสอบผู้ใช้งานที่ ตรงกับการกรอกข้อมูลจะภาไปสู่หน้า แสดงข้อมูลผู้ใช้แสดงเว็บไซต์ของผู้ใช้ แสดงพื้นที่ของผู้ใช้ที่อยู่ใช้ได้ และแสดงพื้นที่ของผู้ใช้งานที่ใช้ไปแล้ว

3.3.1.2 แผนกิจกรรมของหน้าสร้างเว็บไซต์ใหม่

กระบวนการสร้างหน้าเว็บไซต์ใหม่ของผู้ใช้งานผู้ใช้งานจะต้องเลือกโปรแกรมภาษา เลือกเฟรมเวิร์ก หรือ ลากวางไฟล์แล้วตั้งชื่อโดเมนหลังจากที่ผู้ใช้กดปุ่มสร้างระบบจะทำการติดตั้ง โครงสร้างตามที่ผู้ใช้งานได้เลือกไว้จากนั้นจะเก็บข้อมูลเส้นของไฟล์ไปที่ฐานข้อมูล โดยมีรายละเอียด และขั้นตอนการทำงานดังรูปที่ 3.5

- 1) ผู้ใช้เข้าไปหน้าสร้างเว็บไซต์
- 2) เลือกโปรแกรมภาษา เลือกเฟรมเวิร์กและตั้งชื่อโดเมน
- 3) กดปุ่มสร้างเว็บไซต์

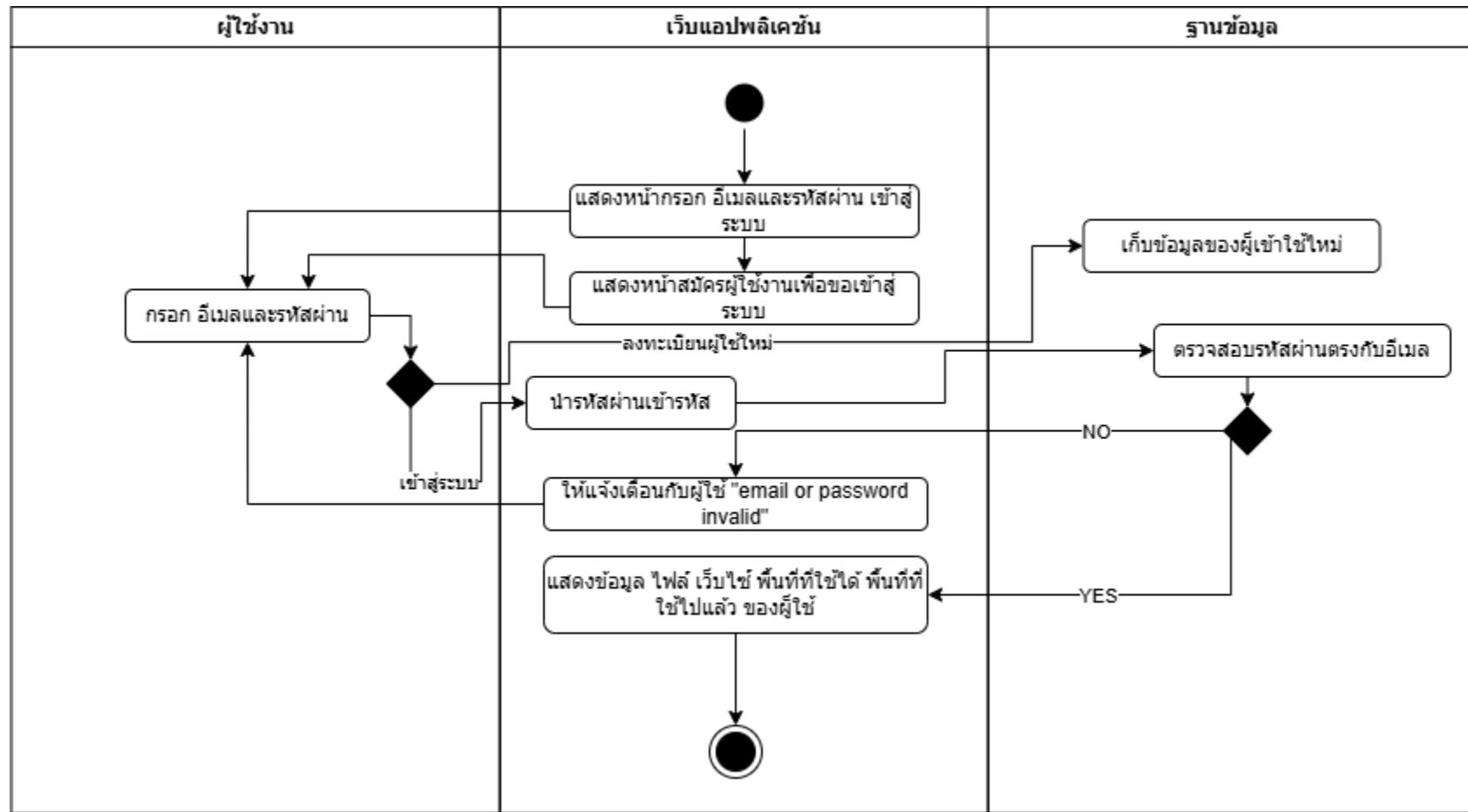
3.3.1.3 แผนภาพกิจกรรมของหน้าแสดงเว็บไซต์ที่มีอยู่ของผู้ใช้

กระบวนการแสดงหน้าเว็บไซต์ทั้งหมดของผู้ใช้งานจะเป็นการดึงมาจากฐานข้อมูลที่เป็นที่อยู่ของไฟล์ที่ได้บันทึกไว้ในฐานข้อมูลผู้ใช้สามารถเข้าถึงการตั้งค่าเว็บไซต์ เพิ่มส่วนเสริม ล้างเว็บไซต์ เริ่มการทำงาน ดาวน์โหลดไฟล์บีบอัด หรือ ลบเว็บไซต์นั้น ๆ ได้ โดยมีรายละเอียดและขั้นตอน การทำงานดังรูปที่ 3.6

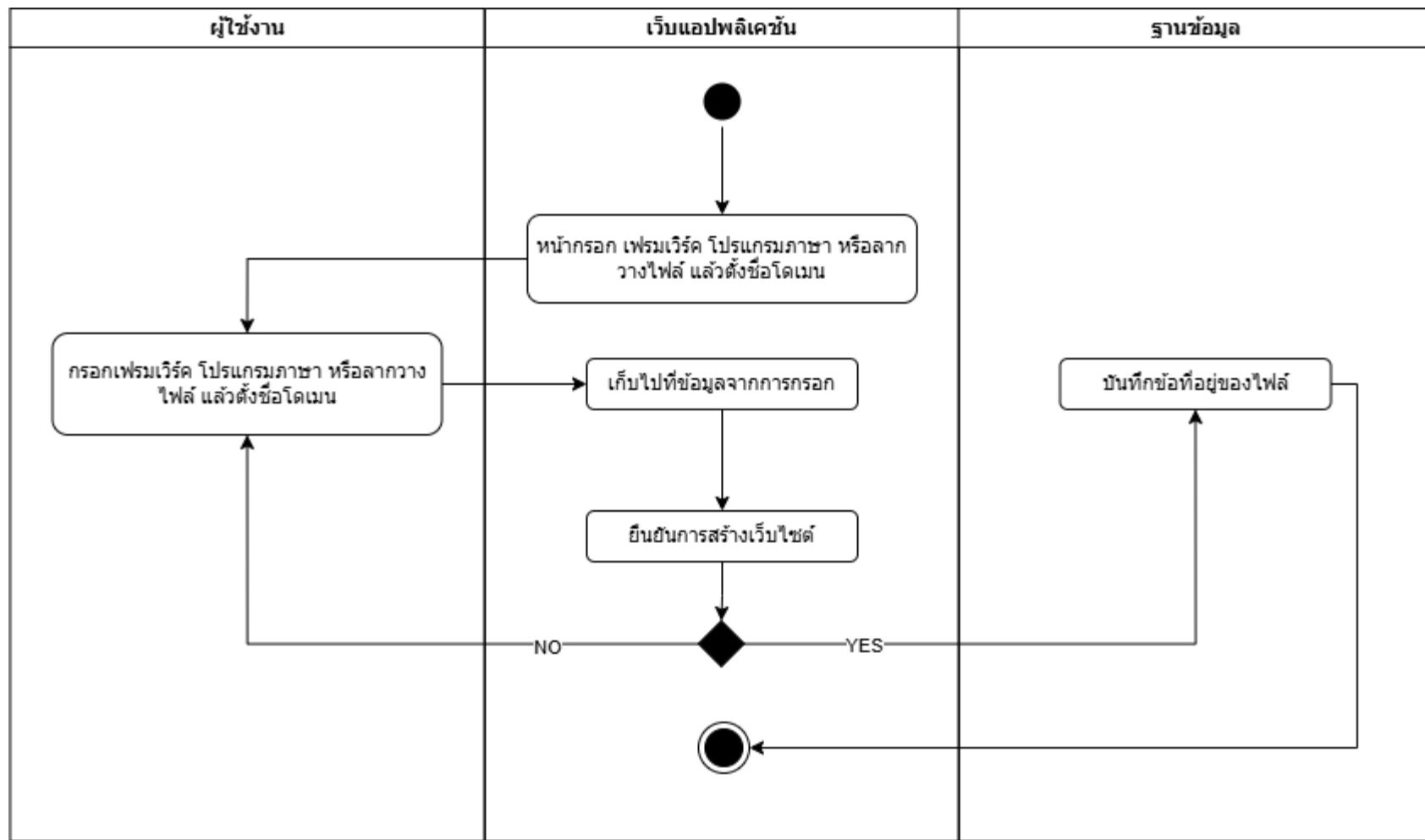
- 1) แสดงเว็บไซต์จากการดึงที่อยู่ของไฟล์จากฐานข้อมูลของผู้ใช้นี้มาแสดงผล
- 2) แสดงการเข้าถึงและการจัดการของไฟล์
 - 2.1) การตั้งค่าเว็บไซต์
 - 2.2) เพิ่มส่วนเสริม
 - 2.3) ล้างเว็บไซต์เริ่มการทำงาน
 - 2.4) ดาวน์โหลดไฟล์บีบอัด
 - 2.5) ลบเว็บไซต์

3.3.1.4 แผนกิจกรรมของหน้าแสดงการตั้งค่าเว็บไซต์

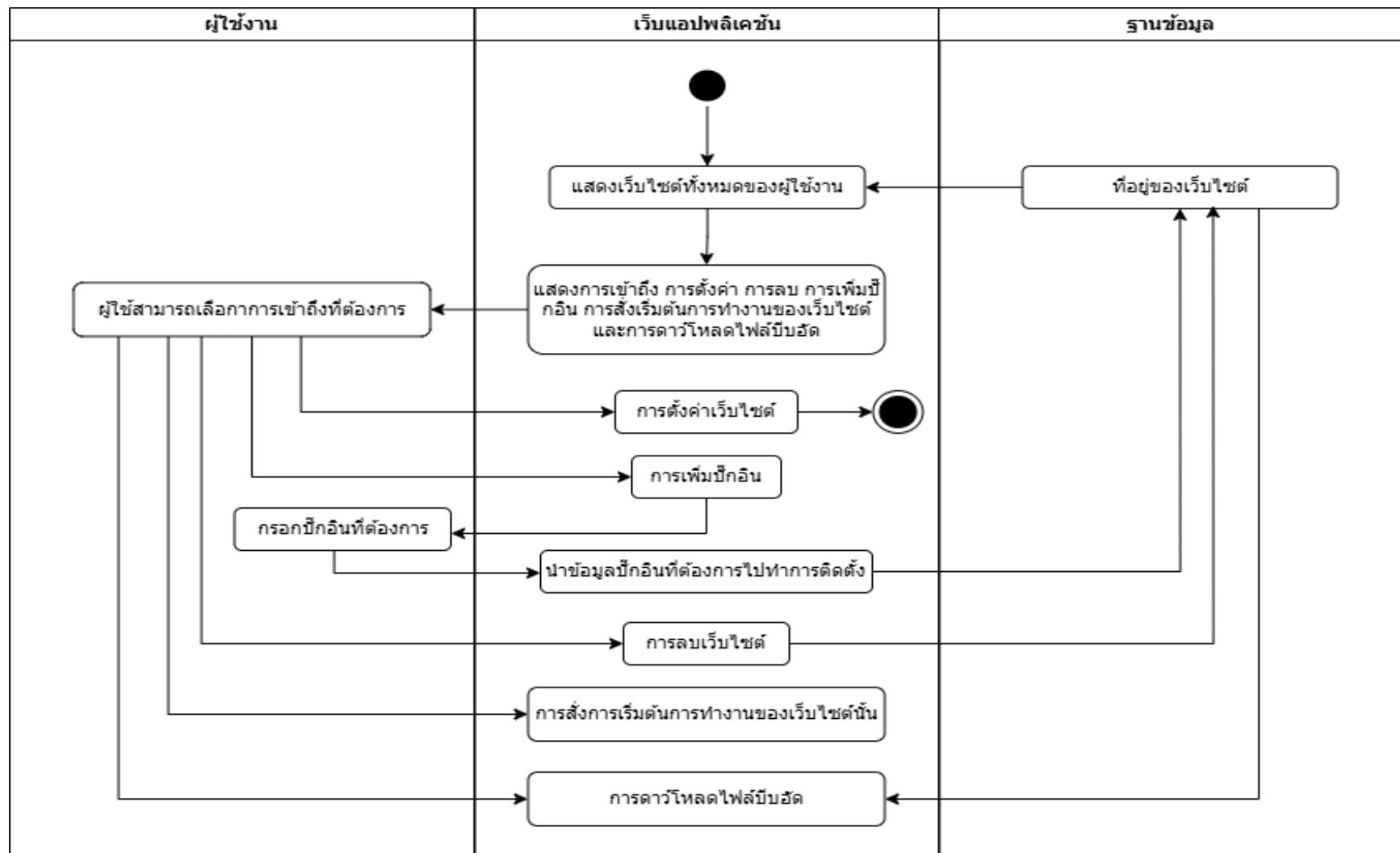
กระบวนการแสดงหน้าการตั้งค่าของเว็บไซต์จะทำให้ผู้ใช้สามารถเข้าถึงการจัดการต่าง ๆ เช่น การแสดงไฟล์ของเว็บไซต์ทั้งหมด การเปลี่ยนชื่อโดเมน การแก้ไขไฟล์ การแสดงผลของพื้นที่ที่ใช้ไปของเว็บไซต์ โดยมีรายละเอียด และขั้นตอนการทำงานดังรูปที่ 3.7



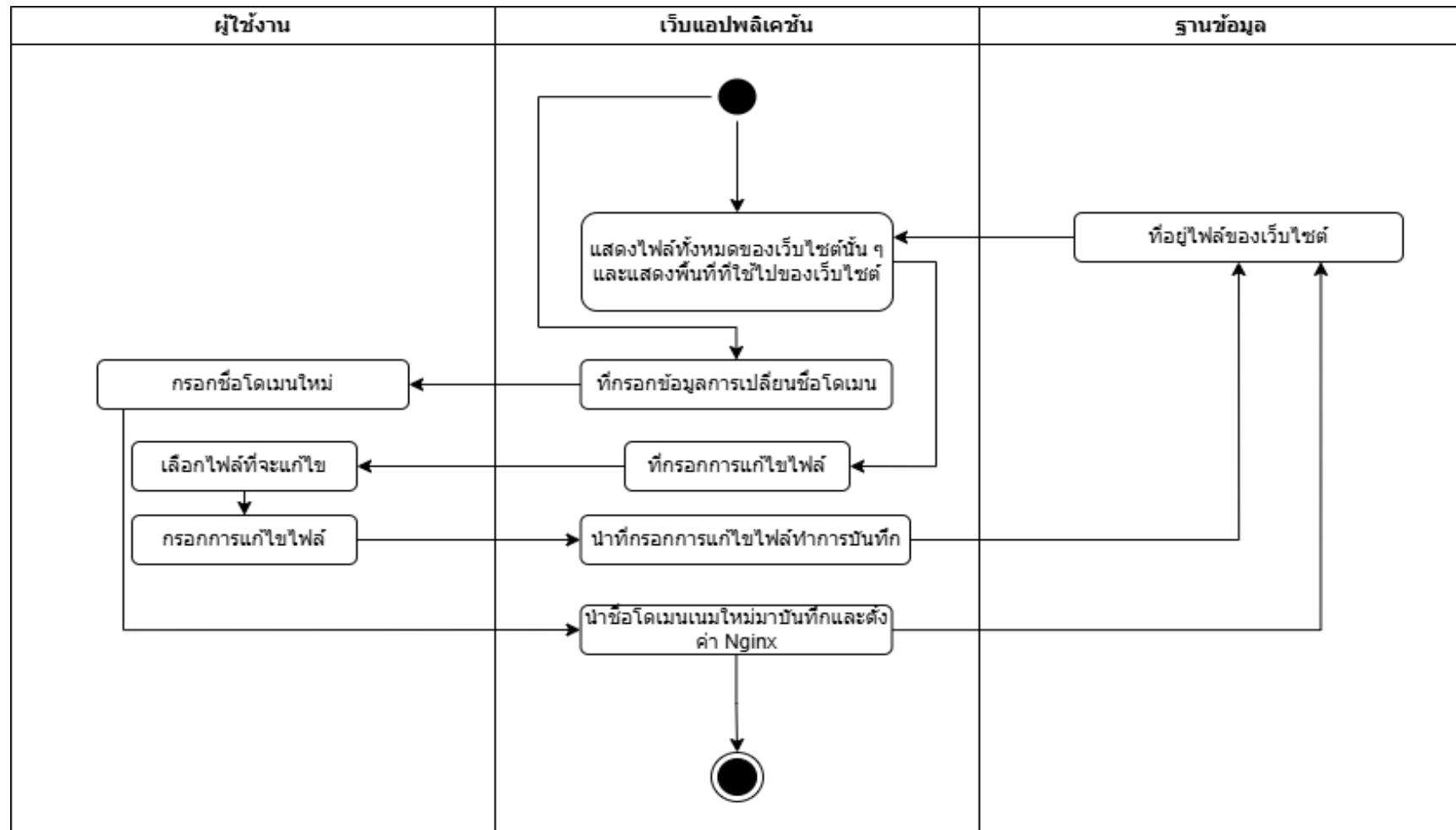
รูปที่ 3.4 แผนภาพของกิจกรรมของหน้าเข้าสู่ระบบ



รูปที่ 3.5 แผนกิจกรรมของหน้าสร้างเว็บไซต์ใหม่



รูปที่ 3.6 แผนภาพกิจกรรมของหน้าแสดงเว็บไซต์ที่มีอยู่ของผู้ใช้



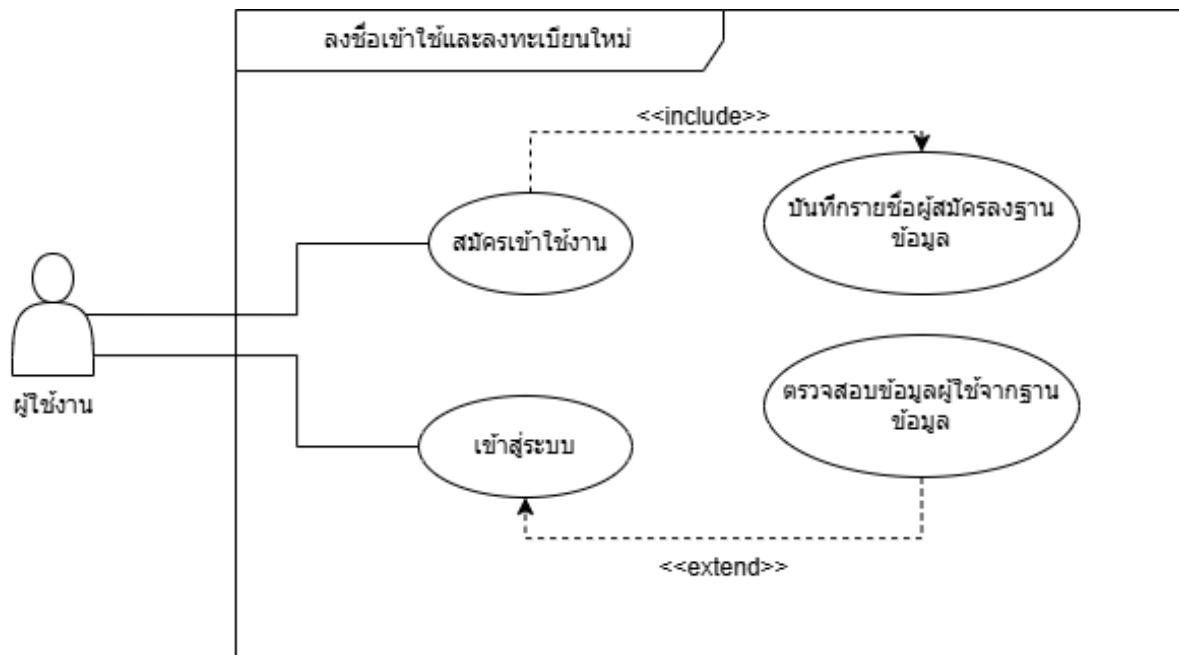
รูปที่ 3.7 แผนกิจกรรมของหน้าแสดงการตั้งค่าเว็บไซต์

- 1) ผู้ใช้เข้ามาสู่หน้าการตั้งจ้างการเข้าถึงจากหน้าเว็บไซต์นั้น ๆ ของผู้ใช้งานตามขั้น
กระบวนการทำงานที่ 3.3.1.3
- 2) แสดงข้อมูลเพลสทั้งหมดของเว็บไซต์นั้น
- 3) สามารถแก้ไขพายที่ต้องการได้
- 4) การเปลี่ยนชื่อโดเมนจะทำการให้ผู้ใช้กรอกชื่อโดเมนที่ต้องการจะเปลี่ยน
- 5) แสดงพื้นที่ที่ใช้ไปของเว็บไซต์

3.3.2 แผนภาพมูสเดส (Use Case Diagram)

แผนภาพมูสเดสเป็นทบทวนสำคัญในการแสดงความสัมพันธ์ระหว่างผู้ใช้งาน (Actor) กับ พังก์ชัน
การทำงานต่าง ๆ ของระบบ (Use Case) โดยแผนภาพนี้ช่วยให้มองเห็นภาพรวมของระบบได้
อย่างชัดเจนว่ามีผู้ใช้งานประเภทใด และแต่ละประเภทสามารถดำเนินการได้บ้าง นอกจากนี้แผนภาพ
มูสเดสยังมีประโยชน์ในการสนับสนุนการวางแผนและการออกแบบระบบเพื่อให้สามารถตอบสนอง
ความต้องการของผู้ใช้งานได้อย่างมีประสิทธิภาพ คือทั้งยังช่วยส่งเสริมความเข้าใจและการสื่อสาร
ที่สอดคล้องกันระหว่างทีมพัฒนาและผู้เกี่ยวข้องในกระบวนการทำงาน สามารถแบ่งออกเป็น 4 ส่วน
ดังรูปที่ 3.8 แผนภาพมูสเดสนี้อธิบายถึงระบบการลงชื่อเข้าใช้และการลงทะเบียนใหม่ หากต้องการ
ลงทะเบียนใหม่ ผู้ใช้จำเป็นต้องกรอกแบบฟอร์มจากเว็บแอปพลิเคชัน เพื่อบันทึกข้อมูลลงในฐานข้อมูล
และในกรณีที่ผู้ใช้ต้องการลงชื่อเข้าใช้ จำเป็นต้องมีข้อมูลในฐานข้อมูลอยู่แล้ว ระบบจึงจะสามารถ
ตรวจสอบและให้สิทธิ์การเข้าถึงได้ โดยผู้ใช้จะถูกนำเข้าสู่หน้าหลักของระบบ

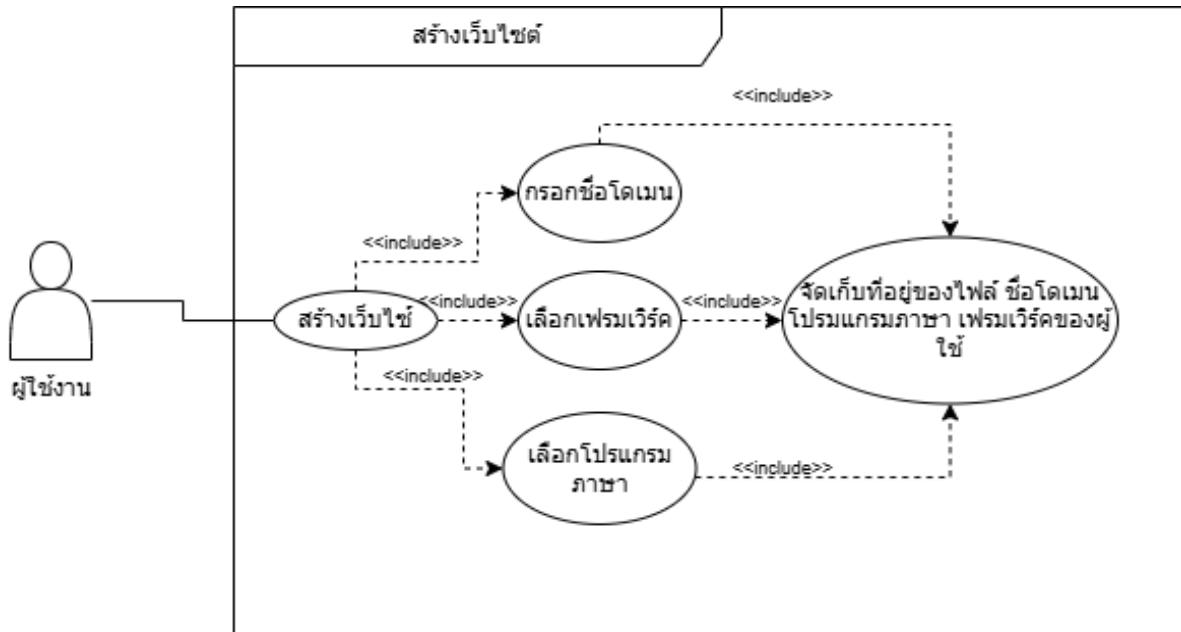
3.3.2.1 แผนภาพข้อมูลเดสก์การลงชื่อเข้าใช้และลงทะเบียนใหม่



รูปที่ 3.8 แผนภาพข้อมูลเดสก์การลงชื่อเข้าใช้และลงทะเบียนใหม่

3.3.2.2 แผนภาพข้อมูลเดสส์ร้างเว็บไซต์

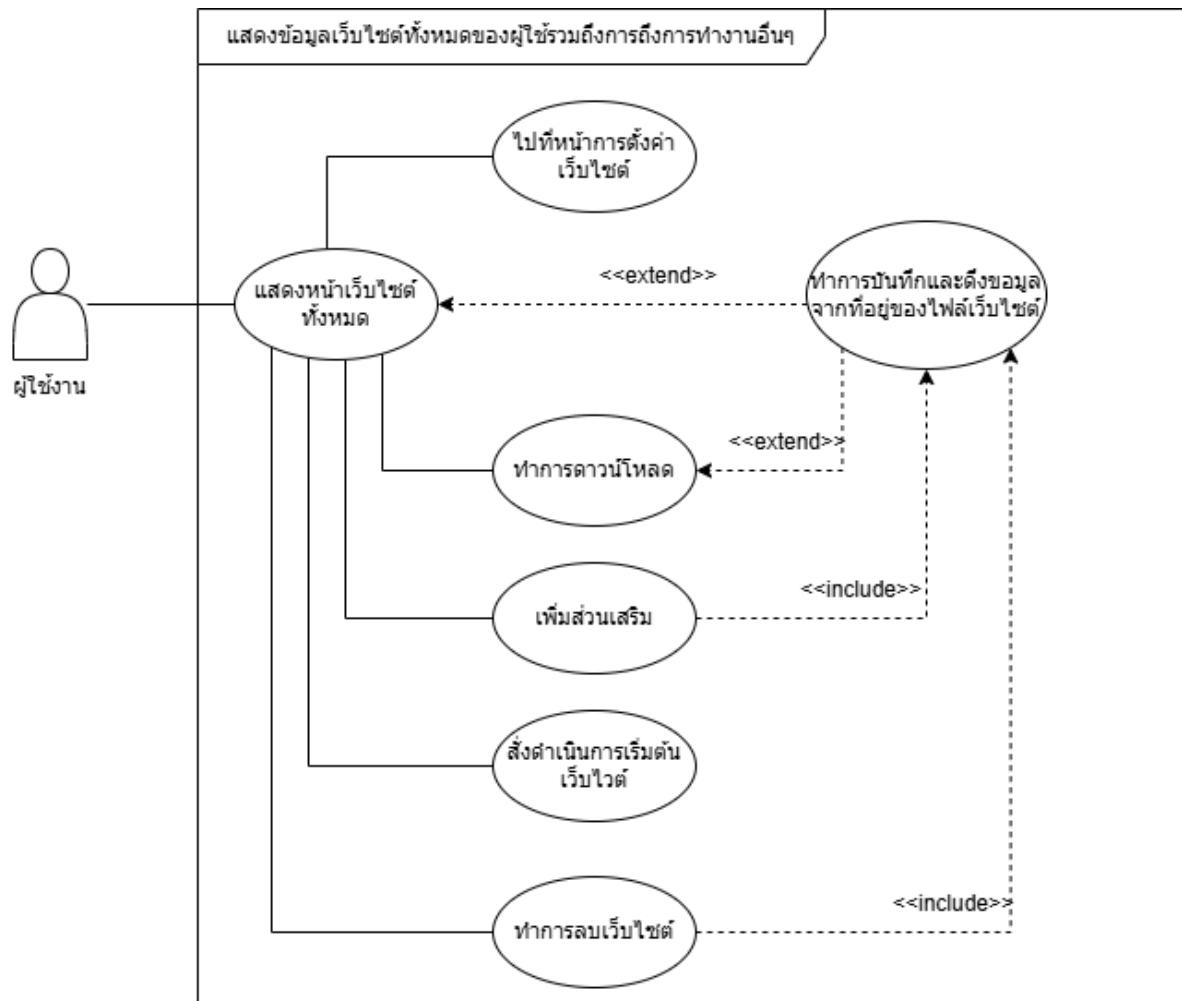
จากรูปที่ 3.9 แผนภาพข้อมูลเดสก์นี้อธิบายถึงกระบวนการสร้างเว็บไซต์ใหม่ของผู้ใช้ โดยผู้ใช้สามารถกรอกข้อมูลที่จำเป็น เช่น ชื่อเว็บไซต์ ชื่อโดเมน และเลือก Framework ที่ต้องการใช้งาน ระบบจะตรวจสอบความถูกต้องของข้อมูลและสร้างโครงสร้างพื้นฐาน (เฟลเดอร์และไฟล์ต้น) สำหรับเว็บไซต์นั้น พร้อมบันทึกข้อมูลลงในฐานข้อมูลเพื่อจัดเก็บและใช้งานในอนาคต



รูปที่ 3.9 แผนภาพพยุสเคสสร้างเว็บไซต์

3.3.2.3 แผนภาพพยุสเคสแสดงข้อมูลเว็บไซต์ทั้งหมดของผู้ใช้รวมถึงการเข้าถึง

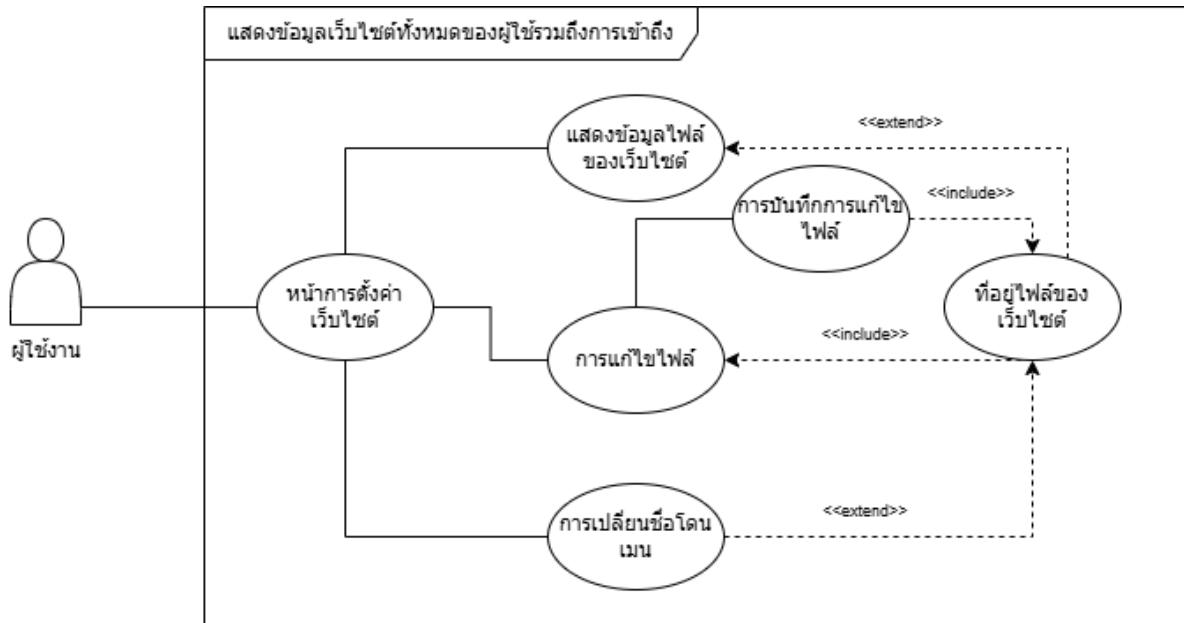
แผนภาพพยุสเคสนี้อธิบายถึงพัฒนาการแสดงข้อมูลเว็บไซต์ทั้งหมดของผู้ใช้ ผู้ใช้สามารถเข้าถึงรายการเว็บไซต์ที่สร้างไว้ในระบบ ซึ่งจะแสดงข้อมูล เช่น ชื่อเว็บไซต์ ชื่อโดเมน และสถานะของเว็บไซต์ ผู้ใช้สามารถกดเพื่อเข้าดูรายละเอียดของเว็บไซต์ หรือทำการเปิดเว็บไซต์ผ่าน Browser ได้โดยตรง รายละเอียดสามารถดูได้ดังรูปที่ 3.10



รูปที่ 3.10 แผนภาพพยุลเดสแสดงข้อมูลเว็บไซต์ทั้งหมดของผู้ใช้รวมถึงการทำงานอีกฯ

3.3.2.4 แผนภาพพยุลเดสของ การตั้งค่าเว็บไซต์

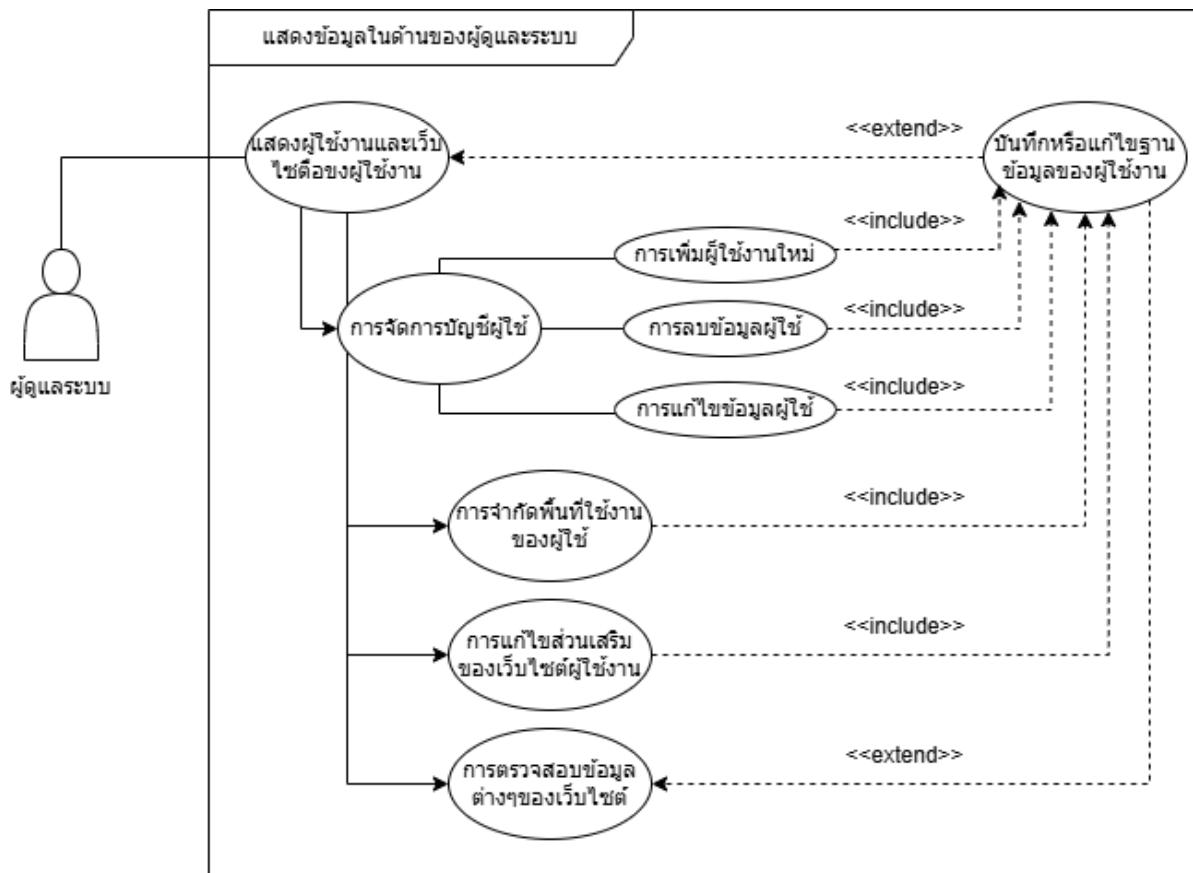
แผนภาพพยุลเดสนี้ อธิบายถึงกระบวนการตั้งค่าต่าง ๆ การแก้ไขไฟล์ในเว็บไซต์ของผู้ใช้ ผู้ใช้สามารถเข้าถึงโครงสร้างไฟล์ผ่าน GUI ของระบบ เพื่อคูไฟล์เดอร์หรือไฟล์ต่าง ๆ จากนั้นสามารถเลือกไฟล์ที่ต้องการเปิดและแก้ไขเนื้อหาใน Text Editor ได้โดยตรง และยังสามารถเปลี่ยนชื่อโดเมนได้โดยการเปลี่ยนแปลงทั้งหมดจะถูกบันทึกลงในระบบ รายละเอียดแสดงดังรูปที่ 3.11



รูปที่ 3.11 แผนภาพผู้ใช้งาน

3.3.2.5 แผนภาพผู้ใช้งาน

แผนภาพผู้ใช้งานนี้ชี้ให้เห็นถึงกระบวนการจัดการระบบของผู้ดูแลระบบ ผู้ดูแลระบบสามารถดำเนินการจัดการบัญชีผู้ใช้งาน รวมถึงการสร้าง แก้ไข ลบ และรังับบัญชีผู้ใช้ที่ลงทะเบียนไว้ นอกเหนือจากนี้ ผู้ดูแลระบบสามารถตรวจสอบเว็บไซต์ของผู้ใช้ ดำเนินการลบหรือรังับเว็บไซต์ที่ลงทะเบียนขึ้นมา รวมถึงการกำหนดพื้นที่ใช้งานสูงสุดให้กับผู้ใช้แต่ละราย ผู้ดูแลระบบยังสามารถเข้าถึงการตั้งค่าระบบเพื่อตรวจสอบสถานะเซิร์ฟเวอร์ ดูปริมาณการใช้งานทรัพยากร รวมถึงจัดการการติดตั้งส่วนเสริม โดยการเปลี่ยนแปลงทั้งหมดจะถูกบันทึกลงในระบบ รายละเอียดดังรูปที่ 3.12

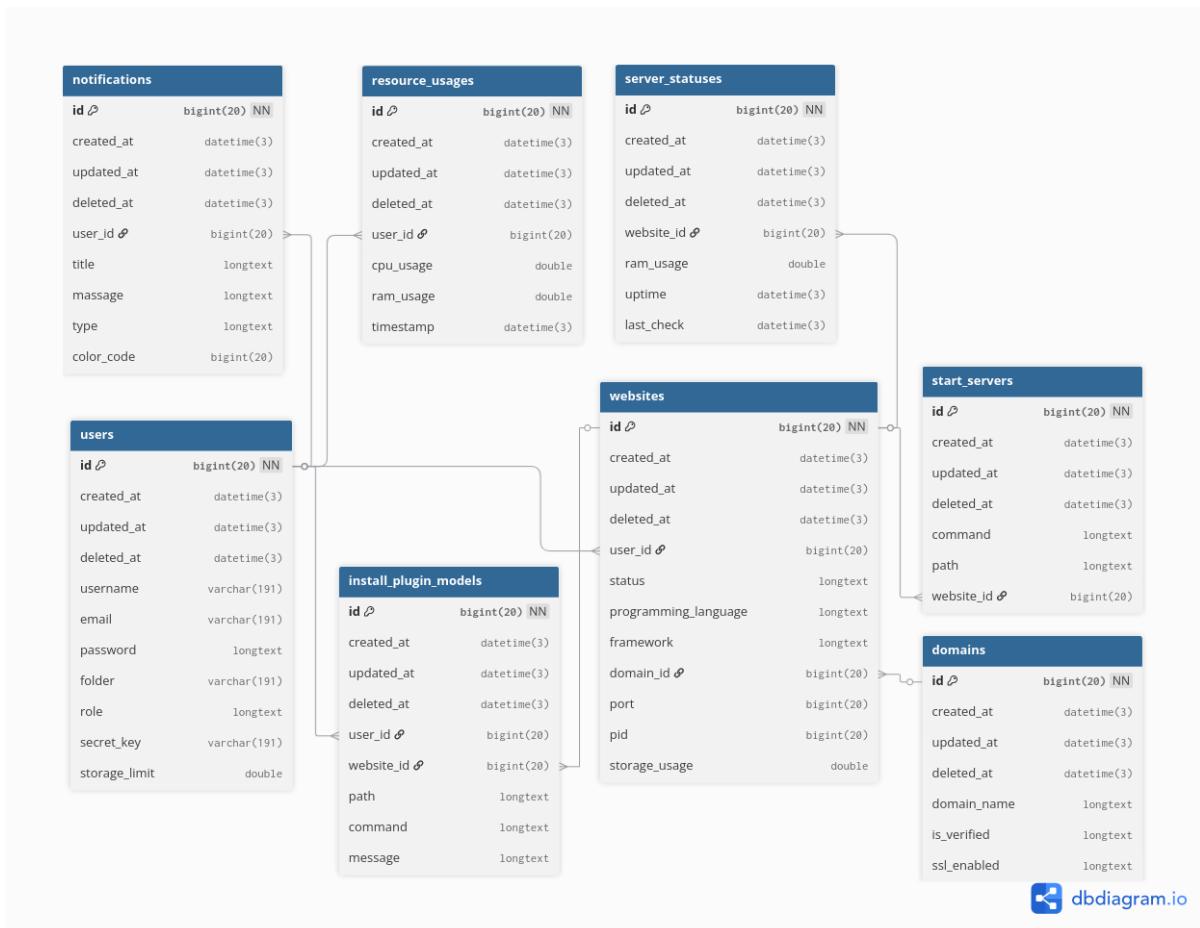


รูปที่ 3.12 แผนภาพพยัญชนะของผู้ดูแลระบบ

3.4 การออกแบบฐานข้อมูล

3.4.1 แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล

ฐานข้อมูลนี้ถูกออกแบบให้รองรับการทำงานของเว็บแอปพลิเคชันเครื่องมือช่วยสร้างโครงสร้างพื้นฐานของเว็บไซต์โดยมีการออกแบบตารางและความสัมพันธ์ที่สอดคล้องกับการทำงานของระบบในภาพรวม ทั้งการจัดเก็บข้อมูลผู้ใช้งาน การสร้างเว็บไซต์ การจัดการส่วนเสริม การจัดการไฟล์และการตั้งค่าต่าง ๆ ของเว็บไซต์ โดยมีความสัมพันธ์และแผนภาพรวมตามแสดงดังรูปที่ 3.13



รูปที่ 3.13 แผนภาพแสดงความสัมพันธ์ระหว่างข้อมูล

จากรูปที่ 3.13 รายละเอียดและความสัมพันธ์ของข้อมูลที่ทำการเก็บข้อมูลของผู้ใช้เก็บใช้ต่อไปนี้
ส่วนเสริมมีความสัมพันธ์ดังตารางด้านล่างต่อไปนี้

ตารางที่ 3.1 ตารางผู้ใช้ (users)

ชื่อตาราง	users		
คำอธิบาย	ข้อมูลผู้ใช้ระบบ		
Primary Key (PK)	id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัสผู้ใช้
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)
username	VARCHAR	191	ชื่อผู้ใช้
email	VARCHAR	191	อีเมล
password	LONGTEXT		รหัสผ่านที่เข้ารหัส
folder	VARCHAR	191	โฟลเดอร์ของผู้ใช้
role	LONGTEXT		บทบาท
secret_key	VARCHAR	191	คีย์ลับสำหรับผู้ใช้
storage_limit	DOUBLE		ขีดจำกัดพื้นที่เก็บข้อมูล

ตารางที่ 3.2 ตารางเว็บไซต์ (websites)

ชื่อตาราง	websites		
คำอธิบาย	ข้อมูลเว็บไซต์ของผู้ใช้		
Primary Key (PK)	id		
Foreign Key (FK)	user_id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัสเว็บไซต์
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต

ตารางที่ 3.2 ตารางเว็บไซต์ (websites) (ต่อ)

ชื่อตาราง	websites		
คำอธิบาย	ข้อมูลเว็บไซต์ของผู้ใช้		
Primary Key (PK)	id		
Foreign Key (FK)	user_id, domain_id		
Field	Datatype	Size	Description
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)
user_id	BIGINT	20	รหัสผู้ใช้ (FK)
status	LONGTEXT		สถานะเว็บไซต์
programming_language	LONGTEXT		ภาษาโปรแกรมที่ใช้
framework	LONGTEXT		Framework ที่ใช้
domain_id	BIGINT	20	รหัสโดเมน (FK)
port	BIGINT	20	พอร์ตที่ใช้
pid	BIGINT	20	Process ID
storage_usage	DOUBLE		พื้นที่เก็บข้อมูลที่ใช้

ตารางที่ 3.3 ตารางข้อมูลโดเมน (domain)

ชื่อตาราง	domain		
คำอธิบาย	ข้อมูลโดเมน		
Primary Key (PK)	id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัสโดเมน
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)
domain_name	LONGTEXT		ชื่อดomen

ตารางที่ 3.3 ตารางชื่อ miền (domain) (ต่อ)

ชื่อตาราง	domain		
คำอธิบาย	ชื่อ miền		
Primary Key (PK)	id		
Field	Datatype	Size	Description
is_verified	LONGTEXT		สถานะการยืนยัน
ssl_enabled	LONGTEXT		สถานะ SSL

ตารางที่ 3.4 ตารางคำสั่งเริ่มต้นเซิร์ฟเวอร์ (start servers)

ชื่อตาราง	start_servers		
คำอธิบาย	คำสั่งเริ่มต้นเซิร์ฟเวอร์		
Primary Key (PK)	id		
Foreign Key (FK)	website_id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัส
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)
command	LONGTEXT		คำสั่ง
path	LONGTEXT		path
website_id	BIGINT	20	รหัสเว็บไซต์ (FK)

ตารางที่ 3.5 ตารางการติดตั้งปลั๊กอินและคำสั่ง (install plugin models)

ชื่อตาราง	install_plugin_models		
คำอธิบาย	การติดตั้งปลั๊กอินและคำสั่ง		
Primary Key (PK)	id		
Foreign Key (FK)	user_id, website_id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัส
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)
user_id	BIGINT	20	รหัสผู้ใช้ (FK)
website_id	BIGINT	20	รหัสเว็บไซต์ (FK)
path	LONGTEXT		path
command	LONGTEXT		คำสั่ง
message	LONGTEXT		ข้อความ

ตารางที่ 3.6 ตารางสถานะเซิร์ฟเวอร์ (server statuses)

ชื่อตาราง	server_statuses		
คำอธิบาย	สถานะเซิร์ฟเวอร์		
Primary Key (PK)	id		
Foreign Key (FK)	website_id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัส
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)

ตารางที่ 3.6 ตารางสถานะเซิร์ฟเวอร์ (server statuses) (ต่อ)

ชื่อตาราง	server_statuses		
คำอธิบาย	สถานะเซิร์ฟเวอร์		
Primary Key (PK)	id		
Foreign Key (FK)	website_id		
Field	Datatype	Size	Description
website_id	BIGINT	20	รหัสเว็บไซต์ (FK)
ram_usage	DOUBLE		การใช้ RAM
uptime	DATETIME	3	เวลาการทำงาน
last_check	DATETIME	3	ครั้งสุดท้ายที่ตรวจสอบ

ตารางที่ 3.7 ตารางการใช้ทรัพยากระบบ (resource usages)

ชื่อตาราง	resource_usages		
คำอธิบาย	การใช้ทรัพยากระบบ		
Primary Key (PK)	id		
Foreign Key (FK)	user_id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัส
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)
user_id	BIGINT	20	รหัสผู้ใช้ (FK)
cpu_usage	DOUBLE		การใช้ CPU
ram_usage	DOUBLE		การใช้ RAM
timestamp	DATETIME	3	เวลาที่บันทึกข้อมูล

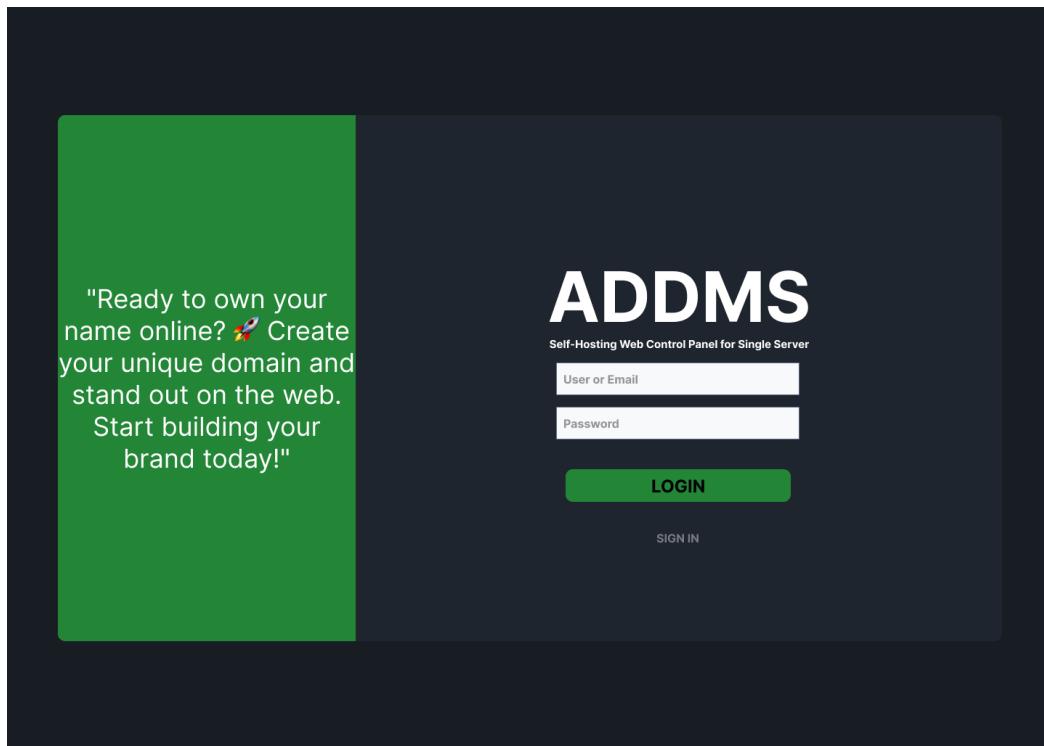
ตารางที่ 3.8 ตารางการแจ้งเตือนผู้ใช้ (notifications)

ชื่อตาราง	notifications		
คำอธิบาย	การแจ้งเตือนผู้ใช้		
Primary Key (PK)	id		
Foreign Key (FK)	user_id		
Field	Datatype	Size	Description
id	BIGINT	20	รหัส
created_at	DATETIME	3	เวลาที่สร้าง
updated_at	DATETIME	3	เวลาที่อัปเดต
deleted_at	DATETIME	3	เวลาที่ลบ (Soft Delete)
user_id	BIGINT	20	รหัสผู้ใช้ (FK)
title	LONGTEXT		หัวข้อการแจ้งเตือน
massage	LONGTEXT		ข้อความการแจ้งเตือน
type	LONGTEXT		ประเภทการแจ้งเตือน
color_code	BIGINT	20	รหัสสี

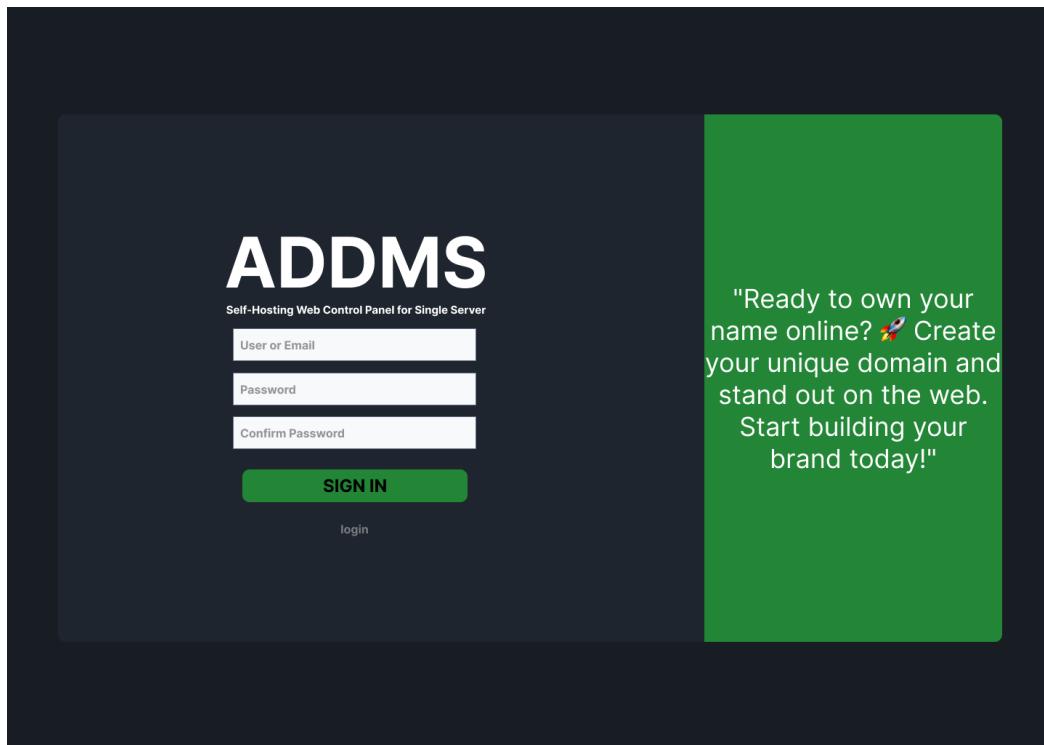
3.5 ส่วนอุปแบบการติดต่อประสานงานผู้ใช้งาน

3.5.1 หน้าเข้าสู่ระบบและลงทะเบียน

เมื่อผู้ใช้เข้าสู่หน้าแรกของระบบ จะพบกับหน้าจอที่มีฟอร์มสำหรับการเข้าสู่ระบบ ซึ่งประกอบด้วยช่องให้กรอกอีเมลหรือชื่อผู้ใช้ และรหัสผ่าน หลังจากผู้ใช้กรอกข้อมูลและกดปุ่ม "เข้าสู่ระบบ" ระบบจะทำการตรวจสอบข้อมูลที่กรอกกว่าตรงกับฐานข้อมูลหรือไม่ หากข้อมูลถูกต้อง ระบบจะนำผู้ใช้เข้าสู่หน้าหลักที่แสดงรายการเว็บไซต์ของพวกรเข้า แต่หากข้อมูลไม่ถูกต้อง ระบบจะแสดงข้อความแจ้งเตือน เช่น "Username or Password invalid" นอกจากนี้ยังมีลิงก์สำหรับการลงทะเบียนใหม่ หากผู้ใช้งานไม่มีบัญชี แสดงรายละเอียดดังรูปที่ 3.14 และ รูปที่ 3.15



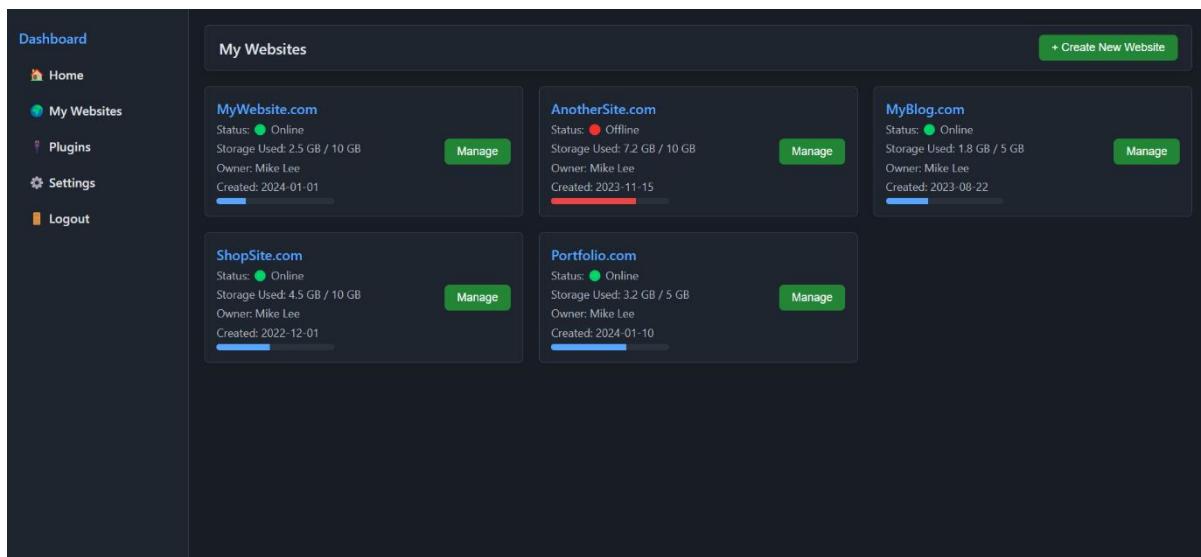
ຮູບທີ 3.14 ມານາຂອງການເຂົ້າສູ່ຮະບບ



ຮູບທີ 3.15 ມານາຂອງການເຂົ້າສູ່ຮະບບຜູ້ໃຊ້ງານໃໝ່

3.5.2 หน้าแสดงเว็บไซต์ของผู้ใช้ทั้งหมด

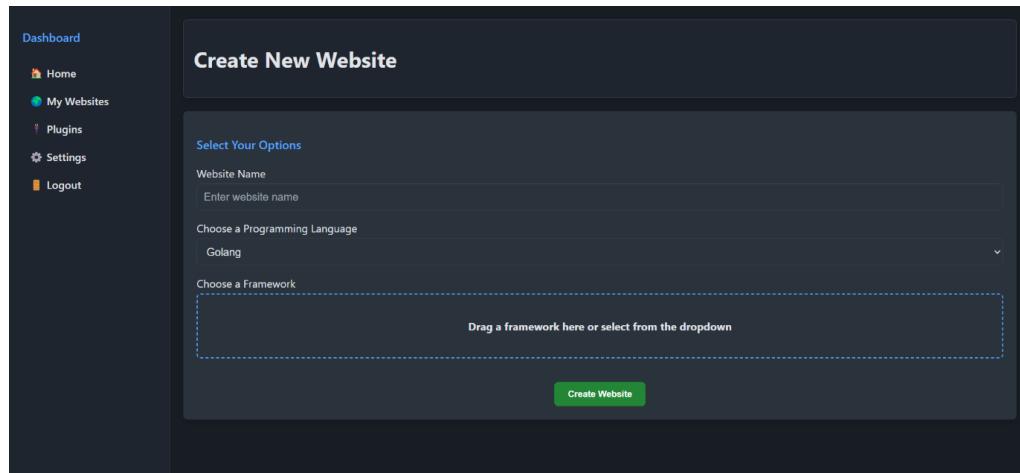
หลังจากเข้าสู่ระบบสำหรับผู้ใช้จะพบหน้าหลักซึ่งจะแสดงรายการเว็บไซต์ทั้งหมดที่ผู้ใช้สร้างไว้ เว็บไซต์เหล่านี้จะแสดงในรูปแบบรายการหรือการ์ด โดยแต่ละรายการจะมีข้อมูลสำคัญ เช่น ชื่อเว็บไซต์ ชื่อโดเมน สถานะ (Running/Stop) ผู้ใช้สามารถคลิกที่ปุ่ม "แก้ไข" หรือ "ดูเว็บไซต์" เพื่อเข้าสู่การจัดการเว็บไซต์นั้น ๆ ได้ทันที หากผู้ใช้มีเว็บไซต์จำนวนมาก ระบบจะมีช่องค้นหาและตัวกรอง เช่น การค้นหาตามชื่อ นอกจากนี้ยังมีปุ่ม "สร้างเว็บไซต์ใหม่" สำหรับเพิ่มเว็บไซต์ใหม่ และตัวเลือก สำหรับการลบเว็บไซต์หรือสั่งดำเนินการเว็บไซต์ แสดงรายละเอียดดังรูปที่ 3.16



รูปที่ 3.16 หน้าแสดงเว็บไซต์ของผู้ใช้ทั้งหมด

3.5.3 หน้าสร้างเว็บไซต์

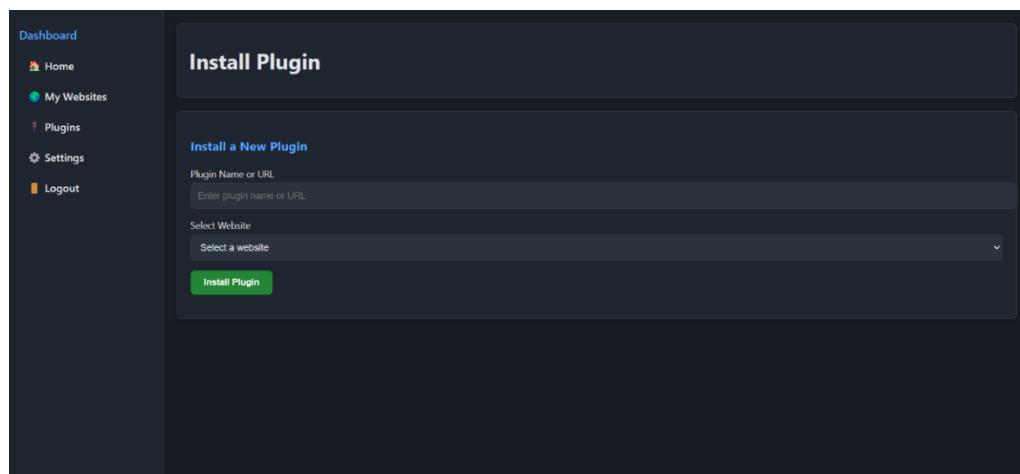
เมื่อผู้ใช้คลิกปุ่ม "สร้างเว็บไซต์ใหม่" ระบบจะนำผู้ใช้มายังหน้าสำหรับสร้างเว็บไซต์ โดยขั้นตอนเริ่มต้นคือการกรอกชื่อเว็บไซต์ เลือกเฟรมเวิร์กที่ต้องการใช้ (HTML, PHP, Node.js, Golang หรือ Static) และกำหนดโดเมน ระบบจะแนะนำโดเมนฟรี (เช่น TEST.server.1) ให้ หากผู้ใช้ยังไม่มีโดเมนเป็นของตัวเอง เมื่อผู้ใช้กรอกข้อมูลครบถ้วนและกดปุ่ม "สร้าง" ระบบจะทำการตั้งค่าโครงสร้างพื้นฐานสำหรับเว็บไซต์ เช่น การสร้างโฟลเดอร์ไฟล์ การกำหนดค่า Nginx และสร้างข้อมูลในฐานข้อมูล หลังจากการสร้างเสร็จสิ้น ผู้ใช้จะถูกนำไปยังหน้าจอสำหรับจัดการเว็บไซต์นั้นทันที แสดงรายละเอียดดังรูปที่ 3.17



รูปที่ 3.17 หน้าสร้างเว็บไซต์

3.5.4 หน้าติดตั้งส่วนเสริมเสริม

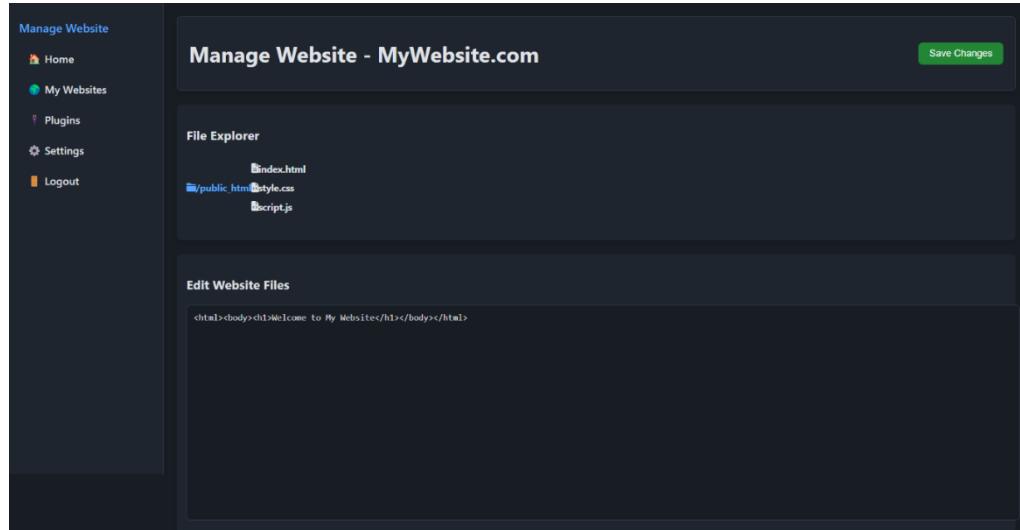
เมื่อผู้ใช้ต้องการเพิ่มฟังก์ชันการทำงานให้กับเว็บไซต์ เช่น การเพิ่มระบบจัดการไฟล์หรือการปรับแต่ง SEO พวกรเข้าสามารถเข้าไปยังหน้าติดตั้งส่วนเสริม ในหน้าจอนี้จะแสดงรายการส่วนเสริมที่มีอยู่ในระบบ โดยแต่ละส่วนเสริมจะมีคำอธิบาย เวอร์ชัน และปุ่ม "ติดตั้ง" เมื่อผู้ใช้กดติดตั้งส่วนเสริมระบบจะทำการเพิ่มข้อมูลส่วนเสริมลงในฐานข้อมูล พร้อมอัปเดตไฟล์หรือโฟลเดอร์ที่เกี่ยวข้องในเซิร์ฟเวอร์ เช่น การเพิ่มตัวดูไฟล์ PHP หรือการเพิ่มฟังก์ชันใน Nginx เพื่อรับส่วนเสริม หลังจากติดตั้งสำเร็จ ระบบจะแสดงสถานะว่าส่วนเสริมติดตั้งเสร็จสมบูรณ์ แสดงรายละเอียดดังรูปที่ 3.18



รูปที่ 3.18 หน้าติดตั้งส่วนเสริมเสริม

3.5.5 หน้าตั้งค่าเว็บไซต์

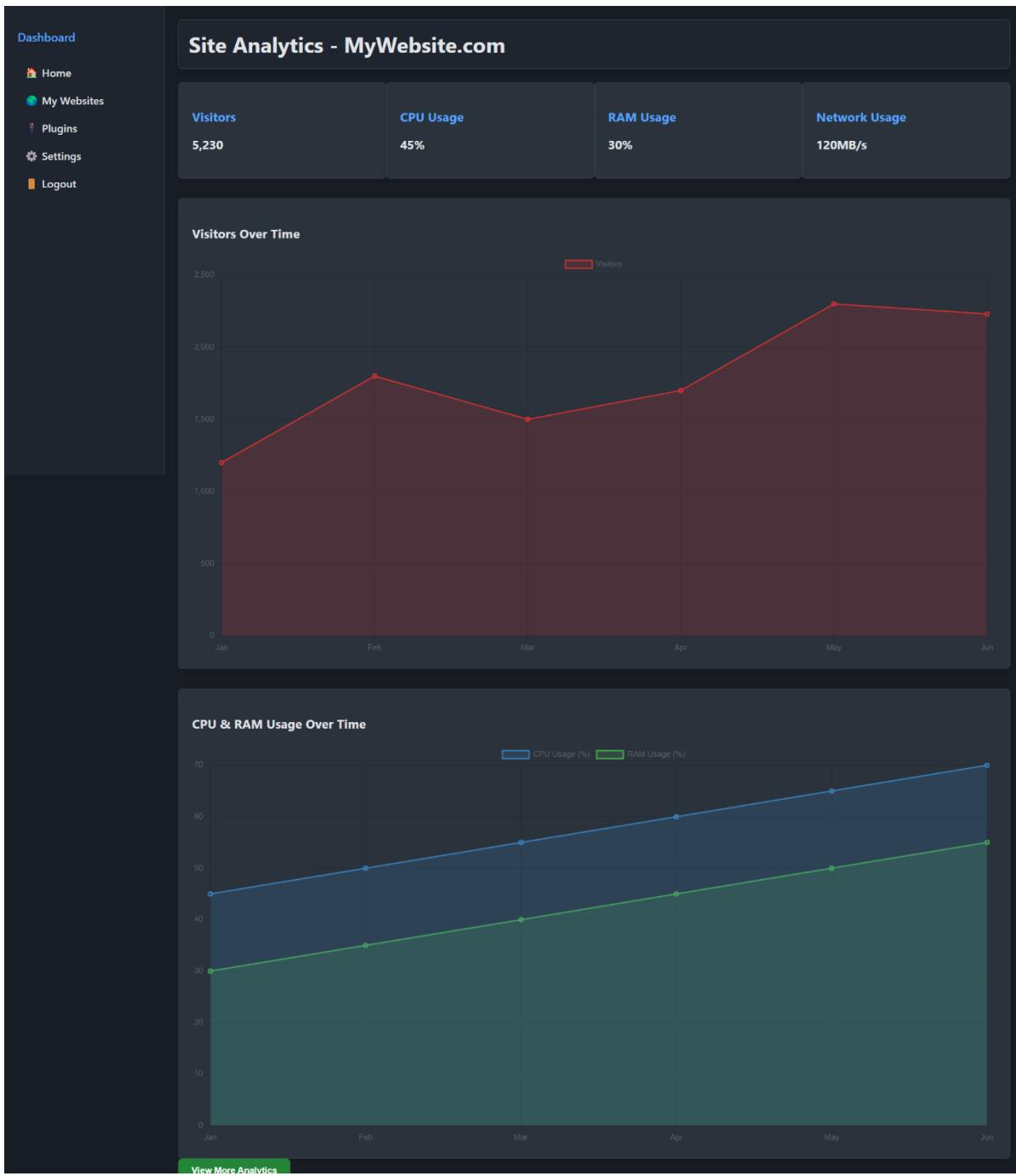
หน้าตั้งค่าเว็บไซต์ออกแบบมาเพื่อให้ผู้ใช้สามารถปรับแต่งเว็บไซต์ได้ตามความต้องการ โดยเมื่อผู้ใช้เข้าสู่หน้าตั้งค่า จะพบกับตัวเลือกต่าง ๆ เช่น การเปลี่ยนโดเมนหลัก (Primary Domain) หรือการกำหนดค่าความปลอดภัย (เช่น HTTPS) เมื่อผู้ใช้ทำการแก้ไขค่าต่าง ๆ และกด "บันทึกการตั้งค่า" ระบบจะทำการอัปเดตข้อมูลในฐานข้อมูล หลังจากนั้น ระบบจะแจ้งให้ผู้ใช้ทราบว่า การเปลี่ยนแปลงถูกบันทึกสำเร็จแล้ว แสดงรายละเอียดดังรูปที่ 3.19



รูปที่ 3.19 หน้าตั้งค่าเว็บไซต์

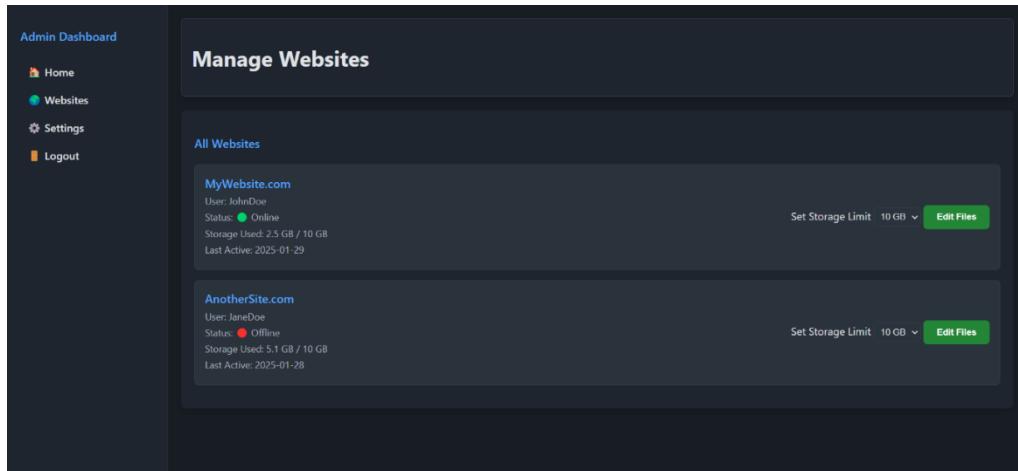
3.5.6 หน้าวิเคราะห์ข้อมูลเว็บไซต์

ออกแบบมาเพื่อให้ผู้ใช้สามารถดูข้อมูลเชิงลึกเกี่ยวกับการใช้งานเว็บไซต์ของตน ซึ่งจะช่วยให้ผู้ใช้สามารถติดตามประสิทธิภาพและพฤติกรรมของผู้เข้าชมเว็บไซต์ได้ง่าย ๆ โดยหน้าวิเคราะห์นี้จะแสดงข้อมูลต่าง ๆ ที่เกี่ยวข้องกับการใช้งานเว็บไซต์ เช่น จำนวนผู้เยี่ยมชม เวลาเฉลี่ยที่ใช้บนเว็บไซต์ จำนวนหน้าเพจที่ดู และข้อมูลการเข้าถึงจากแหล่งที่มาที่แตกต่างกัน (เช่น การเข้าถึงจากเครื่องมือค้นหาหรือการเข้าถึงโดยตรง) ซึ่งจะช่วยให้ผู้ใช้สามารถปรับปรุงเว็บไซต์ตามความต้องการและแนวโน้มของผู้ใช้งาน เมื่อผู้ใช้เข้าสู่หน้าวิเคราะห์เว็บไซต์ (Site Analytics) จะพบกับข้อมูลที่แสดงอยู่ในรูปแบบกราฟ หรือแผนภูมิที่เข้าใจง่าย รวมถึงตัวเลขที่แสดงการเติบโตและการมีปฏิสัมพันธ์ของผู้ใช้ที่เข้ามาใช้งานเว็บไซต์ ซึ่งสามารถดูได้ในกรอบเวลา เช่น สัปดาห์ เดือน หรือปี โดยสามารถเลือกรอบเวลาได้ตามต้องการ แสดงรายละเอียดดังรูปที่ 3.20



3.5.7 หน้าของผู้ดูแลระบบ

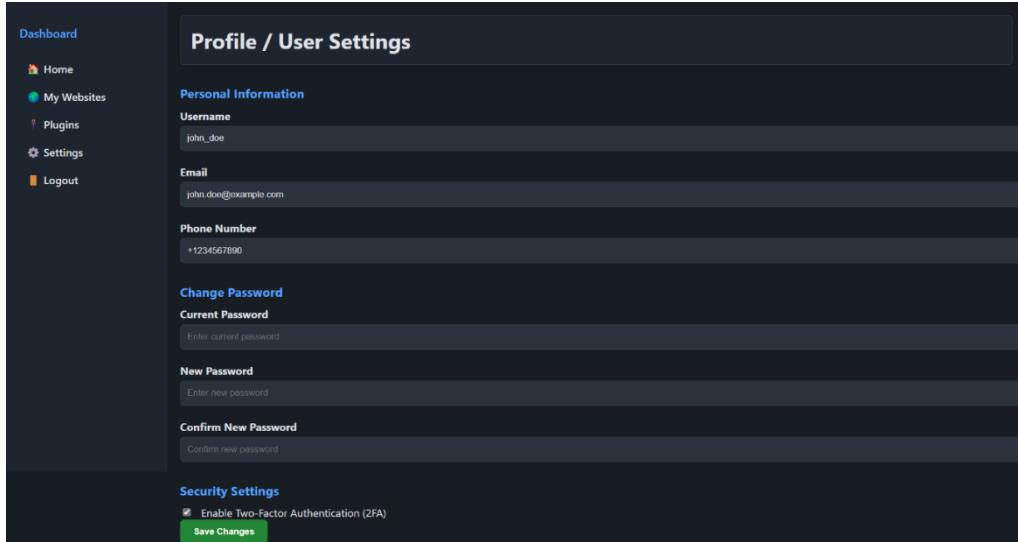
ออกแบบเพื่อให้ผู้ดูแลระบบสามารถจัดการเว็บไซต์ต่าง ๆ ได้สะดวก โดยแสดงข้อมูลสำคัญของแต่ละเว็บไซต์ เช่น ชื่อเว็บไซต์ สถานะออนไลน์/ออฟไลน์ การใช้งานพื้นที่เก็บข้อมูลและข้อมูลการใช้งานล่าสุด ในรูปแบบการ์ดที่เข้าใจง่าย ผู้ดูแลระบบสามารถตั้งค่าพื้นที่เก็บข้อมูล (Storage Limit) และเข้าสู่หน้าจัดการไฟล์ได้โดยตรงผ่านปุ่ม "Edit Files" ซึ่งช่วยให้การจัดการเว็บไซต์เป็นไปอย่างมีประสิทธิภาพและสะดวกต่อการควบคุมหลายเว็บไซต์พร้อมกัน แสดงรายละเอียดดังรูปที่ 3.21



รูปที่ 3.21 หน้าของผู้ดูแลระบบ

3.5.8 หน้าของการตั้งค่าข้อมูลส่วนตัวของผู้ใช้งาน

ออกแบบเพื่อให้ผู้ใช้สามารถจัดการข้อมูลส่วนตัวและการตั้งค่าบัญชีของตนเองได้อย่างสะดวกโดยในหน้านี้ผู้ใช้สามารถอัปเดตข้อมูลส่วนตัว เช่น ชื่อผู้ใช้ อีเมล และหมายเลขโทรศัพท์รวมถึงสามารถเปลี่ยนรหัสผ่านได้ง่าย ๆ ผ่านช่องกรอกข้อมูลที่จัดเตรียมไว้ให้ นอกจากนี้ยังมีการตั้งค่าความปลอดภัยเพิ่มเติม เช่น การเปิดใช้งานการยืนยันตัวตนด้วยสองปัจจัย (2FA) เพื่อเพิ่มความปลอดภัยให้กับบัญชีผู้ใช้ เมื่อผู้ใช้เข้าสู่หน้านี้จะพบกับการแบ่งหมวดหมู่การตั้งค่าที่ชัดเจน เช่น ข้อมูลส่วนตัว การเปลี่ยนรหัสผ่าน และการตั้งค่าความปลอดภัย โดยการเปลี่ยนแปลงทุกอย่างสามารถทำได้โดยการกรอกข้อมูลในฟอร์มและคลิกปุ่ม "Save Changes" เพื่อบันทึกการเปลี่ยนแปลงทั้งหมด รายละเอียดดังรูปที่ 3.22



รูปที่ 3.22 หน้าของการตั้งค่าข้อมูลส่วนตัวของผู้ใช้งาน

3.6 บทสรุป

ในบทที่ 3 ได้อธิบายถึงการออกแบบระบบ ทั้งในส่วนของโครงสร้างฐานข้อมูลและการทำงานของส่วนต่าง ๆ ของระบบ เช่น การจัดการเว็บไซต์ การติดตั้งส่วนเสริม และการตั้งค่าผ่าน GUI ซึ่งเป็นหัวใจสำคัญของการพัฒนาโครงการนี้ โดยการออกแบบดังกล่าวถูกวางแผนอย่างรอบคอบเพื่อรองรับการทำงานที่หลากหลายและตอบสนองต่อความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพ ข้อมูลทั้งหมดในบทนี้จะถูกนำไปใช้ใน บทที่ 4 การพัฒนาระบบ ซึ่งจะกล่าวถึงกระบวนการพัฒนาโครงการตามการออกแบบที่ได้วางไว้ เช่น การพัฒนาโค้ดเบื้องต้น การเชื่อมต่อฐานข้อมูล การเขียน API สำหรับจัดการข้อมูล และการพัฒนา GUI เพื่อให้ระบบสามารถทำงานได้ตามฟังก์ชันที่ระบุไว้ รวมถึงการปรับปรุงระบบให้เหมาะสมกับการใช้งานจริง

บทที่ 4

ผลการทดสอบ

บทนี้นำเสนอผลการพัฒนาและการทดสอบระบบติดตั้งแอปพลิเคชันและเชื่อมโยงโดเมนอัตโนมัติ ที่นำเสนอในบทที่ 3 ซึ่งครอบคลุมถึงการพัฒนาระบบ การทดสอบฟังก์ชัน การประเมินประสิทธิภาพ และความปลอดภัย การทดสอบใช้งานจริง การทดสอบแสดงให้เห็นถึงประสิทธิภาพ ความสามารถ และข้อจำกัดของระบบ รวมถึงแนวทางการพัฒนาระบบในอนาคต โดยการนำเสนอแบ่งเป็นหัวข้ออย่างต่อๆ กัน ตามที่ระบุไว้ในบทที่ 3 โดยการนำเสนอแบ่งเป็นหัวข้ออย่างต่อๆ กัน

4.1 ผลการพัฒนาระบบ

4.1.1 ส่วนของผู้ใช้งานระบบ

ระบบได้รับการพัฒนาตามการออกแบบที่วางไว้ในบทที่ 3 โดยมุ่งเน้นการสร้างในส่วนของผู้ใช้งาน ที่ใช้ง่าย สวยงาม และมีประสิทธิภาพ

● หน้า LandingPage หรือ โฮมเพจ

หน้า LandingPage หรือ โฮมเพจเป็นหน้าแรกที่ผู้ใช้งานจะพบเมื่อเข้าสู่เว็บไซต์ ได้รับการออกแบบให้ใช้งานง่าย มีคำอธิบายล้วนๆ ที่เกี่ยวกับระบบติดตั้งแอปพลิเคชันและเชื่อมโยงโดเมนอัตโนมัติ โดยมีองค์ประกอบสำคัญ ดังนี้

- 1) โลโก้สำหรับแสดงอัตลักษณ์ของระบบ
- 2) ปุ่ม “Get Started” เพื่อนำผู้ใช้งานเข้าสู่หน้าการทำงานหลัก
- 3) คำอธิบายคุณสมบัติหลักของระบบอย่างกระชับ
- 4) ส่วนที่แสดงฟีเจอร์สำคัญของระบบ

หน้าโฮมเพจได้รับการออกแบบมาโดยคำนึงถึงความสวยงาม การสร้างความประทับใจแรกและ การใช้งานที่เรียบง่าย ดังรูปที่ 4.1 หน้าโฮมเพจการออกแบบโดยรวมใช้โทนสีขาวเป็นหลักผสมผสานกับสีส้มในจุดเด่น ทำให้หน้าเว็บไซต์ดูสะอาดตา มีความเป็นมืออาชีพ และสามารถสื่อสารข้อมูลเกี่ยวกับระบบได้อย่างมีประสิทธิภาพ ผู้ใช้สามารถทำความเข้าใจเกี่ยวกับฟังก์ชันและขั้นตอนการทำงานของระบบได้ในทันทีที่เข้าชมหน้าเว็บไซต์

4.1.2 พั้งก์ชันการทำงานหลัก

ระบบได้รับการพัฒนาฟังก์ชันการทำงานหลักตามที่ออกแบบไว้ในบทที่ 3 โดยมีรายละเอียดดังนี้

- หน้าสำหรับการลงชื่อเข้าใช้งานระบบ

หน้าสำหรับการลงชื่อเข้าใช้งานระบบเป็นส่วนที่สำคัญที่ใช้ในการยืนยันตัวตนของผู้ใช้งาน ก่อนที่จะเข้าใช้งานระบบ ดังรูปที่ 4.2 หน้าการลงชื่อเข้าใช้ โดยมีองค์ประกอบหลัก ดังนี้

- 1) โลโก้และชื่อระบบ แสดงอยู่ในส่วนบนสุดเพื่อยืนยันว่าผู้ใช้กำลังเข้าสู่ระบบที่ถูกต้อง
 - 2) ชื่อผู้ใช้ (Username) สำหรับระบุตัวตนของผู้ใช้งาน
 - 3) รหัสผ่าน (Password) สำหรับยืนยันตัวตนของผู้ใช้งาน
 - 4) ปุ่ม “Login” หากกรอกข้อมูลครบถ้วน และถูกต้อง สามารถคลิกปุ่มนี้เพื่อดำเนินการเข้าสู่ระบบ
 - 5) Don’t have an account? Register สำหรับผู้ใช้งานที่ยังไม่มีบัญชีและต้องการสร้างบัญชีผู้ใช้ใหม่
 - 6) By creating an account. You agree to our Terms of Service and Privacy Policy สำหรับอ่านข้อกำหนดและนโยบายความเป็นส่วนตัว
 - 7) หากผู้ใช้ลืมรหัสผ่านสามารถกดที่ ลืมรหัสผ่าน(forget password) เพื่อทำการขอเปลี่ยนรหัสผ่านใหม่จากหน้าลืมรหัสผ่าน(forgot password) ดังรูปที่ 4.7 โดยมีขั้นตอนนี้

หน่วยซึ่งเข้าใช้งานระบบนี้ได้ออกแบบให้มีความเรียบง่าย สะอาดตา และใช้งานง่าย การจัดวางองค์ประกอบต่าง ๆ เป็นไปอย่างเป็นระเบียบ ผู้ใช้สามารถกรอกข้อมูลและเข้าสู่ระบบได้อย่างรวดเร็ว และไม่ซับซ้อน

- หน้าสำหรับระบบการลงทะเบียนผู้ใช้ใหม่

หน้าสำหรับระบบการลงทะเบียนผู้ใช้ใหม่ ดังรูปที่ 4.3 เป็นส่วนในการสร้างบัญชีผู้ใช้งานใหม่ สำหรับระบบ โดยมีองค์ประกอบหลัก ดังนี้

- 1) โลโก้และชื่อระบบ และคงอยู่ในส่วนบนสุดเพื่อยืนยันว่าผู้ใช้กำลังลงท่าเบียนผู้ใช้ใหม่กับระบบที่ถูกต้อง
 - 2) ชื่อผู้ใช้ (Username) ต้องเป็นชื่อผู้ใช้ที่ไม่ซ้ำในระบบ
 - 3) อีเมล (Email) ต้องเป็นรูปแบบอีเมลที่ถูกต้อง (example@domain.com)
 - 4) รหัสผ่าน (Password) ต้องมีความยาวอย่างน้อย 8 ตัวอักษร
 - 5) ยืนยันรหัสผ่าน (Confirm Password) ต้องตรงกับช่องรหัสผ่าน
 - 6) ปุ่ม “Create Account” หากกรอกข้อมูลครบถ้วน และถูกต้อง สามารถคลิกปุ่มนี้เพื่อดำเนินการสร้างบัญชีผู้ใช้ใหม่
 - 8) Already have an account? Login here สำหรับผู้ใช้ที่มีบัญชีอยู่แล้ว และต้องการไปยังหน้าเข้าสู่ระบบ

9) By creating an account. You agree to our Terms of Service and Privacy Policy ສໍາຮັບອ່ານ
ຂໍອກຳນົດແລະນົມຍາຍຄວາມເປັນສ່ວນຕົວ
ຂໍອກຳນົດແຕ່ລະຫ່ວງການຈະປາກງູມເມື່ອຜູ້ໃຊ້ຄລິກໃນທີ່ຂ່ອງນັ້ນ ຈະເພື່ອໃຫ້ຂໍອມຸລແລກຮອກຂໍອມຸລໄດ້
ອ່າງຄູກຕ້ອງ ຜ່າຍລດຄວາມພິດພາດໃນກາລົງທະເປີຍນິ້ນໃໝ່

● ນໍາແທນບອ້ານ (Dashboard)

ໜ້າ Dashboard ເປັນໜ້າຫລັກຂອງຮບບກາຣເວັບໄຟ້ຕໍ່ອັນຸມື້ງໃໝ່ຈະປາກງູມເມື່ອຜູ້ໃຊ້ຈະປາກງູມເມື່ອຜູ້ໃຊ້ໄໝ່
ສໍາຄັນດັ່ງຕ້ອນໄປນີ້

1) ສ່ວນຫັວ (Header)

- ດ້ານໜ້າແສດງຂໍອັນຸມື້ງໃໝ່ແລະອື່ນເມລ
- ດ້ານໜ້າແສດງຂໍອມຸລຂອງເຄື່ອງແມ່ຂ່າຍ

2) System Overview

- ແສດງເວັບໄຟ້ຕໍ່ເປີດໃໝ່ຈະນ ເຊີ່ງພວກເຮົາທີ່ກຳລັງທຳການ ແລະພື້ນທີ່ໜ່າຍຄວາມຈຳກັດໃໝ່ຈະນໃໝ່
ສ້າງເວັບໄຟ້

3) Your Websites & System Alerts

- Your Websites ແສດງເວັບໄຟ້ຕໍ່ຜູ້ໃຊ້ຈະສ້າງທີ່ໜ່າຍ ແລະປຸ່ມ “+ New Deploy” ສໍາຮັບສ້າງ
ເວັບໄຟ້ເພີ່ມ
- System Alerts ແສດງກາຣເຈັງເຕືອນຕ່າງ ຈະຂອງເວັບໄຟ້

4) Sidebar

- ໂລໂກ້ແລະໜ້ອຍ່ອຂອງຮບບກາ
- ປຸ່ມ “Dashboard” ດລິກເພື່ອໄປຢັ້ງໜ້າ Dashboard
- ປຸ່ມ “Settings” ດລິກເພື່ອໄປຢັ້ງໜ້າ Settings
- ປຸ່ມ “Logout” ດລິກເພື່ອອອກຈາກຮບບກາ

ໜ້າແທນບອ້ານນີ້ໄດ້ຮັບກາຣອອກແບນນາໃໝ່ໃໝ່ຈະນ ເປັນມີຕຽກກັບຜູ້ໃຊ້ຈະນ ກາຣຈັດເຮີຍເປັນຮະເປີຍ
ໃໝ່ໂທນສີຂາວ-ສົມເພື່ອໃຫ້ອ່ານຈ່າຍແລະສບາຍຕາ ພວ້ອມບອກຂໍອມຸລທີ່ຈຳເປັນສໍາຮັບຜູ້ໃຊ້ ເພື່ອໃຫ້ຜູ້ໃຊ້ຈັດກາຣ
ເວັບໄຟ້ຂອງຕົນໄດ້ມີປະລິກທິກາພສູງທີ່ສຸດ

● หน้าการติดตั้งเว็บไซต์ใหม่ (New Deploy)

หน้าการติดตั้งเว็บไซต์ใหม่เป็นส่วนสำคัญของระบบที่ช่วยให้ผู้ใช้สามารถสร้างและปรับใช้เว็บไซต์ใหม่บนแพลตฟอร์มได้อย่างมีประสิทธิภาพ ระบบได้รับการออกแบบให้รองรับการติดตั้งเว็บไซต์ด้วยเฟรมเวิร์กและภาษาโปรแกรมต่าง ๆ ดังรูปที่ 4.5 องค์ประกอบหลักของหน้าติดตั้งเว็บไซต์ใหม่ประกอบด้วย

1) ช่องการตั้งค่าพื้นฐาน (Basic Configuration)

- แสดงหน้าต่างที่ให้ผู้ใช้กรอกชื่อเว็บไซต์โดยจะสามารถเลือกได้ว่าจะใช้เว็บไซต์ที่อยู่ในเครือของเครื่องแม่ข่าย หรือ ใช้ตัวโดเมนที่จดด้วยตนเองที่จดบนเว็บไซต์ของ เช่น z.com โดยที่จะต้องใช้ที่อยู่อพีลารณะเดียวกับตัวของเครื่องแม่ข่าย
- แสดงช่องให้ผู้ใช้เลือกตัวของโปรแกรมภาษาจะประกอบด้วย nodejs, go, php, html
- แสดงช่องให้ผู้ใช้เลือกตัวของเครื่องมือพัฒนาต่าง ๆ ของ โปรแกรมภาษาที่เลือก

2) ช่องแหล่งที่มาของการติดตั้ง (Deployment source)

- แสดงเครื่องกำกับให้ผู้ใช้ทราบถึงเครื่องที่รองรับให้การอ้างแหล่งที่มาโดย ADDMS ใช้เครื่องมือที่เก็บโค้ดออนไลน์ (GitHub)
- แสดงช่องใส่ลิงค์ของแหล่งที่มาของการติดตั้ง

3) ช่องทรัพยากรและความปลอดภัย (Resources & Security)

- แสดงให้ผู้ใช้ตัดสินใจที่ต้องการทำ โปรโตคอลการเข้ารหัสแบบ SSL

4) ปุ่มของการติดตั้งเว็บไซต์ของผู้ใช้งาน

- หน้าติดตั้งออกแบบให้มีความเข้าใจง่าย เพื่อให้สามารถติดตั้งตัวเว็บได้อย่างรวดเร็ว และสะดวกสบาย

● หน้าการติดตั้งส่วนเสริมและแก้ไขไฟล์

หน้าการติดตั้งส่วนเสริมและแก้ไขไฟล์ เป็นหน้าที่ให้ผู้ใช้งานสามารถจัดการไฟล์และเพิ่มส่วนเสริมให้กับเว็บไซต์ของตนได้อย่างสะดวก หน้านี้ยังมีตัวแก้ไข เพิ่ม ลบ และอัปโหลดไฟล์ โดยออกแบบให้ใช้งานง่าย แม้สำหรับผู้ใช้ที่ไม่ต้องการแก้ไขไฟล์จากภายนอก ดังรูปที่ 4.6 องค์ประกอบหลักของหน้าประกอบด้วย

1) เส้นทางไฟล์ (File Path Navigation)

- แสดงตำแหน่งปัจจุบันของผู้ใช้งานภายใต้ระบบไฟล์ของเว็บไซต์
- ผู้ใช้สามารถกดเพื่อย้อนกลับไปยังตำแหน่งก่อนหน้าได้ง่าย

2) ปุ่มกลับสู่แดชบอร์ด (Dashboard)

- ปุ่มสำหรับกลับไปหน้าหลักของระบบ เพื่อดูภาพรวมของเว็บไซต์ทั้งหมด

3) ชื่อเว็บไซต์และการเข้าชม

- แสดงชื่อเว็บไซต์ที่กำลังแก้ไข

- มีไอคอนรูปโลก () อยู่ด้านหน้าชื่อเว็บไซต์ เพื่อกดเข้าไปเยี่ยมชมเว็บไซต์ได้ทันที

4) หน้าต่างคำสั่งและการติดตั้งส่วนเสริม

- หน้าต่างการสั่งให้เว็บไซต์ทำงาน (Run Command)

- ช่องเลือกไฟล์หรือสคริปต์ที่ต้องการใช้เป็นตัวสั่งงานเว็บไซต์

- ช่องให้ผู้ใช้พิมพ์คำสั่งที่ได้รับอนุญาต

- ปุ่ม ดำเนินการ เพื่อให้ระบบเริ่มทำงานตามไฟล์และคำสั่งที่กำหนด

- แสดงตัวอย่างคำสั่งที่จะถูกประมวลผลบนเซิร์ฟเวอร์ เพื่อความมั่นใจก่อนสั่งรันจริง

- หน้าต่างการติดตั้งส่วนเสริม (Install Plugins)

- ช่องเลือกไฟล์หรือดำเนินการที่ใช้ในการติดตั้งส่วนเสริม

- ช่องให้ผู้ใช้พิมพ์คำสั่งที่เกี่ยวข้องกับการติดตั้ง (ที่ระบบอนุญาตเท่านั้น)

- ปุ่ม ติดตั้ง เพื่อดำเนินการติดตั้งส่วนเสริม

- แสดงตัวอย่างคำสั่งที่จะประมวลผลบนเซิร์ฟเวอร์

5) เมนูจัดการไฟล์และโฟลเดอร์

เมื่อกดปุ่ม “การจัดการไฟล์” ระบบจะแสดงเมนูหลัก 4 รายการ

- สร้างไฟล์หรือโฟลเดอร์ (Create File/Folder)

- เลือกได้ว่าจะสร้างไฟล์ใหม่หรือโฟลเดอร์ใหม่

- มีช่องสำหรับกรอกชื่อไฟล์/โฟลเดอร์

- ปุ่ม สร้าง เพื่อทำการสร้างจริงตามที่กรอก

- อัปโหลดไฟล์ (Upload File)

- มีปุ่มให้เลือกไฟล์จากเครื่องของผู้ใช้

- ปุ่ม อัปโหลด เพื่อส่งไฟล์เข้าสู่ระบบ

- ดาวน์โหลดไฟล์ (Download File/Folder)

- ใช้งานได้เมื่อออยู่ที่ตำแหน่งแรกสุดของลิสต์

- ระบบจะดาวน์โหลดเป็นไฟล์บีบอัด

- ลบไฟล์หรือโฟลเดอร์ (Delete File/Folder)
 - แสดงหน้าต่างยืนยันก่อนลบทุกครั้ง เพื่อป้องกันการลบโดยไม่ตั้งใจ
- 6) รายการไฟล์และโฟลเดอร์ของเว็บไซต์
- หากกดเข้า โฟลเดอร์ จะเปิดรายการไฟล์/โฟลเดอร์รายใน
 - หากกดเข้า ไฟล์ จะเปิดหน้าต่างແริ่ฟ์ไฟล์ซึ่งมีฟังก์ชัน
 - แก้ไขข้อมูลความต้องการ
 - บุํมันทึกเพื่อเขียนกลับไปยังเซิร์ฟเวอร์
 - รองรับสำหรับภาษาโปรแกรมที่ใช้บ่อย เช่น PHP, HTML, JS, Go ฯลฯ
- หน้าแดเนเชอร์บคุณผู้ดูแลระบบ (Admin Dashboard)

หน้าแดเนเชอร์บคุณนี้สามารถเข้าถึงได้เฉพาะผู้ใช้งานที่มีสิทธิ์เป็นผู้ดูแลระบบ (Admin) เท่านั้น โดยมีหน้าที่หลักคือ จัดการผู้ใช้งานและจัดการเว็บไซต์บนเครื่องแม่ข่าย ซึ่งช่วยให้ผู้ดูแลสามารถตรวจสอบแก้ไข และควบคุมข้อมูลต่าง ๆ ได้สะดวก ดังรูปที่ 4.8 หน้านี้ประกอบด้วย บุํมันและฟังก์ชันหลัก ดังนี้

1. การจัดการผู้ใช้งาน (User Management) เมื่อผู้ใช้กดบุํมัน “จัดการผู้ใช้งาน” ระบบจะแสดงหน้าจัดการผู้ใช้งานทั้งหมด ดังรูปที่ 4.9

- ส่วนหัวของหน้า (Header) แสดงชื่อหัวข้อของหน้า บุํม Refresh สำหรับอัปเดตข้อมูลผู้ใช้งานล่าสุด
- แท็บสรุปข้อมูลผู้ใช้งาน (Summary Bar) จำนวนผู้ใช้งานทั้งหมด จำนวนผู้ใช้งานที่เป็นผู้ดูแลระบบ จำนวนผู้ใช้งานที่เป็นผู้ใช้ทั่วไป
- ช่องค้นหาผู้ใช้งาน (Search Bar) สามารถค้นหาผู้ใช้งานตามชื่อหรืออีเมล
- รายการผู้ใช้งาน (User List) ลำดับผู้ใช้งาน ชื่อผู้ใช้งานและอีเมล
- สิทธิ์ของผู้ใช้งาน (Admin / User)
- พื้นที่จัดเก็บของผู้ใช้งาน (Storage Quota) วันที่สร้างผู้ใช้งาน วันที่มีการแก้ไขล่าสุด
- บุํมเครื่องมือ (Action Buttons) แก้ไขผู้ใช้งาน ลบผู้ใช้งาน

2. การจัดการเว็บไซต์ (Website Manager) เมื่อผู้ใช้กดบุํมัน “จัดการเว็บไซต์ทั้งหมด” ระบบจะแสดงหน้าจัดการเว็บไซต์ทั้งหมด ดังรูปที่ 4.10

- ส่วนหัวของหน้า (Header) แสดงชื่อหัวข้อของหน้า บุํม Refresh สำหรับอัปเดตข้อมูลเว็บไซต์ล่าสุด
- แท็บสรุปข้อมูลเว็บไซต์ (Summary Bar) จำนวนเว็บไซต์ทั้งหมด จำนวนเว็บไซต์ที่กำลังออนไลน์ จำนวนเว็บไซต์ที่กำลังออฟไลน์
- ช่องค้นหาเว็บไซต์ (Search Bar) สามารถค้นหาเว็บไซต์ตามชื่อผู้ใช้งานหรือชื่อเว็บไซต์

- รายการเว็บไซต์ทั้งหมด (Website List) ลำดับเว็บไซต์ ชื่อผู้ใช้งานของเว็บไซต์นั้น ชื่อเว็บไซต์ พร้อมลิงก์ (สามารถกดโคลนรูปโลกเพื่อเปิดเว็บไซต์ได้โดยตรง) พื้นที่จัดเก็บที่เว็บไซต์ใช้งาน วันที่สร้างเว็บไซต์
- การใช้ทรัพยากรของเว็บไซต์ (Resource Usage เช่น CPU, RAM, Disk)
- ปุ่มเครื่องมือ (Action Buttons) แก้ไขเว็บไซต์ แจ้งเตือนผู้ใช้งาน ปุ่มเตือน/ปิดใช้งานเว็บไซต์ หน้าการติดตั้งส่วนเสริมและแก้ไขไฟล์เป็นหน้าที่สำคัญของการแก้และทำให้เว็บไซต์สมบูรณ์ได้ เช่น การแก้ไข เพิ่ม ติดตั้งส่วนเสริม และคำสั่งให้เว็บไซต์ทำงาน การออกแบบให้ออกแบบเป็นมิตรกับผู้ใช้ ใช้งานง่ายเครื่องมือครบ

4.2 ผลการทดสอบระบบ

การทดสอบระบบเป็นขั้นตอนสำคัญในการพัฒนาเว็บไซต์ ADDMS จะนำเสนอผลการทดสอบ พังก์ชันการทำงานในแต่ละส่วน เพื่อให้มั่นใจว่าระบบทำงานได้อย่างถูกต้อง มีประสิทธิภาพ และตอบสนองความต้องการของผู้ใช้

ผลการทดสอบพังก์ชันการทำงานในแต่ละส่วน

ผลการทดสอบพังก์ชันต่าง ๆ ได้ดำเนินการตามแผนที่วางไว้ในบทที่ 3 โดยครอบคลุมที่พังก์ชันหลักของระบบ ผลการทดสอบแสดงให้เห็นถึงความสามารถในการทำงานของระบบตามที่ออกแบบไว้ ดังตารางที่ 4.1

ตารางที่ 4.1 ตารางการทดสอบการทำงานของพังก์ชัน

Group	TC	รายการทดสอบ	ผลลัพธ์
Authen	01	ลงทะเบียนผู้ใช้	ผ่าน
	02	ลงทะเบียนผู้ใช้	ผ่าน
	03	ลืมรหัสผ่าน	ผ่าน
	04	ได้รับคีย์ลับ	ผ่าน
Dashboard	05	แสดงข้อมูลผู้ใช้งาน	ผ่าน
	06	แสดงข้อมูลของเครื่องแม่ข่าย	ผ่าน
	07	แสดงจำนวนเว็บไซต์ เว็บไซต์ออนไลน์ พื้นที่ที่ใช้งานไปแล้ว	ผ่าน
	08	สามารถไปหน้า deploy ได้	ผ่าน
	09	สามารถสั่งทำงานเว็บไซต์ได้	ผ่าน

ตารางที่ 4.1 ตารางการทดสอบการทำงานของฟังก์ชัน (ต่อ)

Group	TC	รายการทดสอบ	ผลลัพธ์
	10	แสดงการแจ้งเตือนเว็บไซต์	ผ่าน
	11	สามารถสั่งหยุดเว็บไซต์ได้	ผ่าน
	12	สามารถสั่งลบเว็บไซต์ได้	ผ่าน
	13	สามารถไปหน้าแก้ไขได้	ผ่าน
edit	14	สามารถกดเส้นทางที่ต้องการได้	ผ่าน
	15	สามารถไปหน้า Dashboard ได้	ผ่าน
	16	สามารถเลือกที่จะใช้หน้า Server Configuration หรือ Configuration	ผ่าน
	17	สามารถพิมคำสั่งให้เว็บไซต์ทำงาน	ผ่าน
	18	สามารถติดตั้งส่วนเซริมได้	ผ่าน
	19	สามารถเช็คดูว่าคำสั่งเป็นยังไงได้	ผ่าน
	20	สามารถแก้ไขไฟล์ได้	ผ่าน
	21	สามารถแก้ไขชื่อไฟล์ได้	ผ่าน
	22	สามารถสั่งทำงานเว็บไซต์ได้	ผ่าน
	23	สามารถสั่งหยุดการทำงานได้	ผ่าน
	24	สามารถอัปโหลดไฟล์ได้หลายไฟล์	ผ่าน
	25	สามารถลบไฟล์เดอร์หรือไฟล์ได้	ผ่าน
	26	สามารถดูประวัตการติดตั้งส่วนเซริมได้	ผ่าน
	27	สามารถสร้างไฟล์เดอร์หรือไฟล์ได้	ผ่าน
Deploypage	28	ตั้งชื่อเว็บไซต์ที่ต้องการติดตั้ง	ผ่าน
	29	เลือกโดยเมนูส่วนตัวหรือโดยเมนูของเว็บไซต์	ผ่าน
	30	เลือกภาษาโปรแกรมในการติดตั้ง	ผ่าน
	31	เลือกเครื่องมือสำหรับติดตั้ง	ผ่าน
	32	เลือกการทำ Let's Encrypt	ผ่าน
Admin	33	สามารถดูรายการของชื่อทุกชื่อในเครื่องแม่ข่ายได้	ผ่าน
	34	สามารถดูรายการเว็บไซต์ทั้งหมดของเครื่องแม่ข่ายได้	ผ่าน
	35	สร้างคันหนาชื่อผู้ใช้งานได้	ผ่าน

ตารางที่ 4.1 ตารางการทดสอบการทำงานของฟังก์ชัน (ต่อ)

Group	TC	รายการทดสอบ	ผลลัพธ์
	36	สร้างคันหน้าชื่อเว็บไซต์ได้	ผ่าน
	37	แสดงสถานะของเว็บไซต์ ผู้ใช้ได้	ผ่าน
	38	สามารถเปลี่ยนลิทช์ของผู้ใช้งานได้	ผ่าน
	39	สามารถกำหนดพื้นที่ให้ผู้ใช้ได้	ผ่าน
	40	สามารถแก้ไขไฟล์ได้	ผ่าน
	41	แสดงการใช้ทรัพยากรของเว็บไซต์นั้น ๆ ได้	ผ่าน

จากผลการพัฒนาและการทดสอบระบบติดตั้งแอปพลิเคชันและเชื่อมโยงโอด เมนูอัตโนมัติ พบว่า ระบบสามารถทำงานได้ตามที่ออกแบบไว้ ผู้ใช้สามารถใช้งานได้อย่างสะดวก ตั้งแต่การลงทะเบียนเข้าสู่ระบบ จัดการเว็บไซต์ ไปจนถึงการติดตั้งเว็บไซต์ใหม่และส่วนเซริฟ ระบบมีความพร้อมสำหรับการใช้งานจริงในระดับหนึ่ง ทั้งในด้านความถูกต้อง ประสิทธิภาพ และความปลอดภัย

อย่างไรก็ตาม ยังมีข้อจำกัดบางประการ เช่น ประสิทธิภาพเมื่อใช้งานพร้อมกันหลายผู้ใช้ การจัดการทรัพยากรของเซิร์ฟเวอร์ และการปรับปรุงระบบรักษาความปลอดภัยที่ยังสามารถพัฒนาได้เพิ่มเติม งานวิจัยนี้จึงเสนอแนวทางการพัฒนาต่อยอดในอนาคต เช่น การเพิ่มระบบสำรองข้อมูลอัตโนมัติ การปรับปรุงประสิทธิภาพการประมวลผล และการรองรับเฟรมเวิร์กเพิ่มเติม เพื่อให้ระบบมีความสมบูรณ์และเหมาะสมต่อการใช้งานในระดับที่กว้างขึ้น



Sign Up

Log In

🚀 Deploy your websites in seconds

Automated Web Application Deployment with Domain Mapping

เพียงแค่อัปโหลดไฟล์โปรแกรมเบนก์ของคุณแล้วระบบจะดำเนินการให้โดยอัตโนมัติ ไม่ต้องกังวลเรื่องการตั้งค่าหรือการกำหนดค่า Deploy ที่ยุ่งยาก

Get Started

Auto Deploy

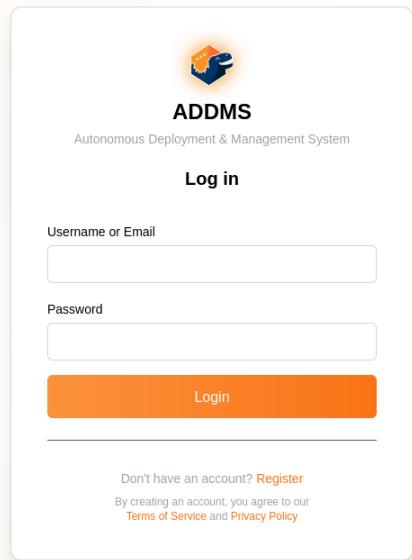
- Upload Files Drag & drop your project
- Auto Configure Smart detection & setup
- Deploy Live Your site is ready!

Custom Domain

ຖຸກອຍ່າງທີ່ຄຸນຕ້ອງການ ໃນທີ່ເດືອນ

ເພີ້ງແຕ່ອັບໂຫດໂປຣເຈິດຂອງຄຸນ ຮະບບະຈົດກາຣຖຸກອຍ່າງໃຫ້ໂດຍອັດໄນມັດ

ຮູບທີ່ 4.1 ມີ້າ LandingPage ທີ່ສະໝັກເພິ່ນ



รูปที่ 4.2 หน้าสำหรับการลงชื่อเข้าใช้งาน



ADDMS
Autonomous Deployment & Management System

Create your account

Username

Email

Password

Confirm Password

Create Account

Already have an account? [Login here](#)
By creating an account, you agree to our
[Terms of Service](#) and [Privacy Policy](#)

รูปที่ 4.3 หน้าสำหรับระบบการลงทะเบียนผู้ใช้ใหม่

The screenshot displays the ADDMS (Advanced Deployment Management System) dashboard. The top navigation bar includes the ADDMS logo, a user icon, and the text "Welcome : test1! 🙌". The sidebar on the left contains links for "Dashboard", "Settings", and "Logout". The main content area is divided into several sections:

- System Overview**: Displays "Website Overview" with 1 Active Websites and 0 Running Servers, and "Resource Usage" with 30% CPU Usage, 1.8 GB Memory Usage, and 12 GB Disk Usage.
- Your Websites & System Alerts**:
 - Your Websites**: Shows a single website entry for "awdh.endfroze...", which is currently OFFLINE. It provides details like Created: 8/28/2025, 4:43:18 PM, SSL Enabled: Yes, Verified: Yes, Storage Usage: 5 MB, Port: 27807, Framework: express, and Language: nodejs. It includes "Start Server" and "Delete" buttons.
 - System Alerts**: Lists recent events:
 - Stop Server: Stop server Successful (0 seconds ago)
 - Action: running server successful in port : 27807 (26 seconds ago)
 - New Deploy: New Deploy New Website : awdh.endfrozen.site (31 seconds ago)
 - Delete website: Delete Website done (33 minutes ago)
 - Stop Server: Stop Server (33 minutes ago)

ຮູບທີ 4.4 ມີຈຳ Dashboard

The screenshot shows the ADDMS web deployment interface. On the left, a sidebar includes a logo, 'ADDMS' text, 'Dashboard', 'Settings', and a 'Logout' button. The main area has three main sections: 'Basic Configuration' (with fields for website name and framework), 'Deployment Source' (set to Git from a repository URL), and 'Resources & Security' (with an SSL certificate toggle switch turned on). A large 'Deploy' button is at the bottom.

Basic Configuration

Enter Your website: .endfrozen.site

Use your domain:

Framework: Node.js

Framework: Express

Deployment Source

Git
From repository

Repository URL: `https://github.com/username/project.git`

Resources & Security

SSL Certificate
Enable HTTPS encryption

Deploy

รูปที่ 4.5 หน้าการติดตั้งเว็บไซต์ใหม่

The screenshot shows a web-based interface for managing a server configuration. At the top, the URL is `test1 / awdh.endfrozen.site` and there is a [Back to Dashboard](#) button. The main title is `awdh.endfrozen.site`. The interface is divided into two main sections: `Server Configuration` and `Plugin Configuration`. The `Server Configuration` section contains fields for `Path Directory` (set to `awdh.endfrozen.site`) and `Run Command` (set to `node server.js`). Below these are `Actions` buttons: a green `>Run server` button and a grey `Advanced Settings` button. A `Command Preview` box shows the command `awdh.endfrozen.site$ node server.js` with the note: `This command will be executed when you start the server`. A status indicator shows `Server offline`. The `Plugin Configuration` section is currently empty. Below the configuration sections is a `Files` section listing the contents of the server directory:

File	Size	Last Modified
<code>node_modules</code>	4.0 KB	8/28/2025, 4:43:18 PM
<code>views</code>	4.0 KB	8/28/2025, 4:43:17 PM
<code>.bashrc</code>	3.7 KB	8/28/2025, 4:43:23 PM
<code>.gitignore</code>	0.0 KB	8/28/2025, 4:43:17 PM
<code>.inputrc</code>	0.0 KB	8/28/2025, 4:43:23 PM
<code>config.json</code>	0.0 KB	8/28/2025, 4:43:18 PM
<code>package-lock.json</code>	28.3 KB	8/28/2025, 4:43:18 PM
<code>package.json</code>	0.3 KB	8/28/2025, 4:43:18 PM
<code>server.js</code>	0.5 KB	8/28/2025, 4:43:17 PM

รูปที่ 4.6 หน้าการติดตั้งส่วนเลริมและแก้ไขไฟล์



Forgot Password

Enter the secret code you saved during registration to reset your password.

Secret Code

New Password

Confirm Password

Reset Password

[Back to login](#)

ຮູບທີ 4.7 ມີ Forgot Password

ADDMS

Dashboard

Admin

Monitoring resource

Settings

Logout

Admin control panel

Manage and control your entire system from one place.

Manage Users

View and manage all user accounts

Manage Websites

View, manage, and control all websites in the system

รูปที่ 4.8 หน้าแดงค์ควบคุมผู้ดูแลระบบ



ADDMS

- [Dashboard](#)
- [Admin](#)
- [Monitoring resource](#)
- [Settings](#)

[Logout](#)

User Management

Manage and control all users in the system

Total Users **22**
Administrators **1**
Regular Users **21**
Recent Signups **16**

All Roles
Export

ID	USER	ROLE	STORAGE LIMIT	CREATED	LAST UPDATE	ACTIONS
1	jojo jojo@gmail.com	User	0 MB	Jun 23, 2025, 02:23 PM	Aug 29, 2025, 03:59 PM	⋮
2	yiyi yiyi@gmail.com	User	0 MB	Jun 25, 2025, 01:32 PM	Aug 29, 2025, 03:59 PM	⋮
3	homepro pornhub@gmail.com	User	0 MB	Jun 30, 2025, 01:59 PM	Aug 29, 2025, 03:59 PM	⋮
4	hello hello@gmail.com	User	0 MB	Jul 18, 2025, 12:06 AM	Aug 29, 2025, 03:59 PM	⋮
5	olo olo@gmail.com	User	0 MB	Jul 22, 2025, 03:47 PM	Aug 29, 2025, 03:59 PM	⋮
6	humdum humdum@gmail.com	User	0 MB	Aug 19, 2025, 10:25 PM	Aug 29, 2025, 03:59 PM	⋮
7	uidtest uidtest@gmail.com	User	0 MB	Aug 25, 2025, 01:16 PM	Aug 29, 2025, 03:59 PM	⋮

ຮູບທີ 4.9 ມານາກາຮຈັດກາຮຸ້ໆໃໝ່ງານ



ADDMS

- [Dashboard](#)
- [Admin](#)
- [Monitoring resource](#)
- [Settings](#)
- [Logout](#)

Website Management

Monitor and manage all websites in your system

TOTAL WEBSITES
22
All registered sites
ONLINE
0
Active & running
OFFLINE
22
Need attention

All Status

#	OWNER	DOMAIN	STATUS	STORAGE	CREATED	RESOURCES	ACTIONS
#1	jojo	test https://test	✖ Offline	0 MB	Jun 29, 2025, 11:57 AM	0.0% 0 MB	
#2	homepro	jaowudayuwvdugawwd https://jaowudayuwvdugawwd	✖ Offline	0 MB	Jul 17, 2025, 05:13 PM	0.0% 0 MB	
#3	homepro	dasdawdasfeawwd https://dasdawdasfeawwd	✖ Offline	0 MB	Jul 18, 2025, 12:01 AM	0.0% 0 MB	
#4	yiyy	12312 https://12312	✖ Offline	0 MB	Jul 22, 2025, 03:12 PM	0.0% 0 MB	
#5	yiyy	qwrdsf https://qwrdsf	✖ Offline	0 MB	Jul 22, 2025, 03:17 PM	0.0% 0 MB	
	...	mkkmmkmmkmkk	✖ Offline	—	—	—	

รูปที่ 4.10 หน้าการจัดการเว็บไซต์

ບທທີ 5

ບທສຽບ

ບທນີ້ນຳເສນອກກາຮຽນສຽບພິບພາບພື້ນນາແລກສອບຮບຕິດຕັ້ງເວັບແອປພລິເຄັ້ນແລກເຊື່ອມໂຍງໂດເມນອັດໂນມັຕີ (ADDMS) ຕາມທີ່ກລ່າວໄວ້ໃນບທທີ 4 ພຣັອມທັ້ງອົບປາຍພລັບພົບທີ່ໄດ້ ເປົ້າຍບໍ່ເຫັນກັບວັດຖຸປະສົງຕີ່ຕັ້ງໄວ້ ຮວມຄື່ງເສນອແນວທາງກາຮຽນພື້ນນາຕ່ອຍອດໃນອາຄາຕ ເພື່ອໃຫ້ຮບມີປະລິທິກາພແລກເໜາມສມຕ່ອກກາຮຽນໃນຮະດັບທີ່ກວ່າງຂຶ້ນ

5.1 ສຽບພິບພາບພື້ນນາ

ຈາກກາຮຽນພື້ນນາຮບ ADDMS ຕາມທີ່ອົກແບບໄວ້ໃນບທທີ 3 ແລກກາຮຽນພື້ນນາທີ່ກຳໜັດ ພລັບພົບທີ່ໄດ້ສາມາດສຽບໄດ້ຕັ້ງນີ້

1) ຄວາມຄຸກຕ້ອງຂອງກາຮຽນ

ຮບສາມາດກາຮຽນໄດ້ຕາມທີ່ອົກແບບໄວ້ ຄຣອບຄຣຸມຕັ້ງແຕກລົງທະເປີນ ກາຮຽນຈັດກາຮຽນໄວ້ ກາຮຽນຕິດຕັ້ງເວັບໄຊຕີ່ໃໝ່ ກາຮຽນປະຕິບັດ (deploy) ແລກກາຮຽນຕິດຕັ້ງສ່ວນເສີມ ໂດຍພັ້ງກົດຕ່າງໆ ຜ່ານກາຮຽນຕິດຕັ້ງກາຮຽນທີ່ກຳໜັດ

2) ປະລິທິກາພກາຮຽນ

ຜູ້ໃຊ້ສາມາດໃຊ້ຈາກຮບໄດ້ຍ່າງສະດວກແລກຮວດເຮົາ ຕັ້ງແຕ່ຂັ້ນຕອນກາຮຽນເຂົ້າສູ່ຮບປິປັນຄື່ງກາຮຽນຈັດກາຮຽນໄວ້ ຮບມືອນເທອຣີເຟຟີທີ່ໃຊ້ຈາກນັ່ງໆ ສບາຍຕາ ແລກເປັນມີຕຽກກັບຜູ້ໃຊ້

3) ຄວາມປິດປັບຂອງຮບ

ຮບມີກາຮຽນອົກແບບໃຫ້ຮອງຮັບກາຮຽນເຂົ້າຮ້າສ SSL ແລກກາຮຽນຈັດກາຮຽນໄວ້ ເພື່ອເພີ່ມຄວາມປິດປັບຂອງຮບໃນກາຮຽນຈັດກາຮຽນໄວ້ ກາຮຽນຈັດກາຮຽນໄວ້ ກາຮຽນຈັດກາຮຽນໄວ້ ກາຮຽນຈັດກາຮຽນໄວ້ ເພື່ອເພີ່ມເພີ່ມເຕີມໄດ້ເຊື່ອມຕົ້ນຕົ້ນ ຮະບັບສໍາຮອງຂໍ້ອມູນແລກກາຮຽນຈັດກາຮຽນໄວ້

4) ຄວາມພຣັອມໃນກາຮຽນ

ຮບມີຄວາມພຣັອມໃນຮະດັບໜຶ່ງທີ່ຈະຄູກນຳໄປໃຊ້ຈາກຈົງຈາກ ສາມາດຮອງຮັບກາຮຽນຈັດກາຮຽນໄວ້ ສ່ວນບຸຄຄລ໌ທີ່ຈະຄູກນຳໄປໃຊ້ຈາກຈົງຈາກ ເພື່ອເພີ່ມເພີ່ມເຕີມໄດ້

5.2 อกิจกรรมผล

เมื่อเปรียบเทียบผลลัพธ์กับวัตถุประสงค์ของงานวิจัย พบว่า ระบบ ADDMS สามารถตอบโจทย์ได้ครบถ้วน ดังนี้

- ตอบสนองความต้องการผู้ใช้ ระบบช่วยให้ผู้ใช้สามารถติดตั้งและจัดการเว็บไซต์ได้โดยไม่ต้องมีความรู้เชิงลึกทางเทคนิคมากนัก
- ลดความซับซ้อนของการจัดการเว็บไซต์ เว็บไซต์และส่วนเสริมต่าง ๆ ถูกทำให้ง่ายและเป็นขั้นตอนที่ชัดเจน
- ระบบยังมีข้อจำกัดบางอย่าง ได้แก่
 - 1) ประสิทธิภาพอาจลดลงเมื่อมีผู้ใช้งานจำนวนมากพร้อมกัน
 - 2) การจัดการทรัพยากรเซิร์ฟเวอร์ยังไม่มีคุณภาพเพียงพอ
 - 3) ระบบปรับใช้ความปลอดภัยควรพัฒนาให้รัดกุมขึ้น เช่น การตรวจสอบการโจรกรรมและระบบสำรองข้อมูลอัตโนมัติ

แม้ว่าระบบจะสามารถทำงานได้ตรงตามที่ออกแบบและมีคุณภาพในการใช้งานจริง แต่ยังต้องการการพัฒนาต่อยอดเพื่อให้รองรับการใช้งานในระดับที่ซับซ้อนและหลากหลายมากยิ่งขึ้น

5.3 ข้อเสนอแนะ

เพื่อพัฒนาระบบให้มีประสิทธิภาพและครอบคลุมการใช้งานมากขึ้น งานวิจัยนี้เสนอแนวทางดังต่อไปนี้

- 1) การเพิ่มระบบสำรองและกู้คืนข้อมูลอัตโนมัติ เพื่อความปลอดภัยของข้อมูลและลดความเสี่ยงจากการผิดพลาดหรือการโจมตีทางไซเบอร์
- 2) การปรับปรุงประสิทธิภาพการประมวลผล ควรปรับแต่งการจัดการทรัพยากรของเซิร์ฟเวอร์ เพื่อรองรับการใช้งานพร้อมกันหลายผู้ใช้ได้อย่างมีเสถียรภาพ
- 3) การเพิ่มเฟรมเวิร์กและภาษาโปรแกรมที่รองรับ นอกจาก Node.js, Go, PHP และ HTML ควรเพิ่มเฟรมเวิร์กอื่น ๆ เช่น Python (Django/Flask), Java (Spring) เพื่อให้ระบบมีคุณภาพและรองรับผู้ใช้ที่หลากหลาย
- 4) การเสริมระบบปรับใช้ความปลอดภัย เช่น การตรวจสอบสิทธิ์แบบหลายชั้น (Multi-Factor Authentication) และระบบแจ้งเตือนความผิดปกติแบบเรียลไทม์
- 5) การปรับปรุงประสบการณ์ผู้ใช้ (UX/UI) ควรออกแบบหน้าจอให้รองรับการใช้งานบนอุปกรณ์หลากหลาย (Responsive Design) และเพิ่มระบบช่วยเหลือผู้ใช้ (Help & Tutorial)

5.4 สรุป

งานวิจัยนี้สามารถพัฒนาระบบติดตั้งแอปพลิเคชันและเชื่อมโยงโดเมนอัตโนมัติ (ADDMS) ที่ตอบโจทย์การจัดการเว็บไซต์ได้อย่างสะดวก มีประสิทธิภาพ และพร้อมต่อการใช้งานจริงในระดับหนึ่ง แม้จะยังมีข้อจำกัดบางประการ แต่แนวทางการพัฒนาต่อขยุดที่นำเสนอจะช่วยให้ระบบมีความสมบูรณ์และเหมาะสมยิ่งขึ้นสำหรับการใช้งานในอนาคต

เอกสารอ้างอิง

- [1] D. Thailand, "Web Server คืออะไร? อธิบายง่ายๆ เข้าใจได้ภายใน 5 นาที," Devhub Thailand, [Online]. Available: <https://devhub.in.th/blog/what-is-web-server>. [Accessed 13 มกราคม 2568].
- [2] P. Ontime, "Web Server คืออะไร?," PP Ontime, [Online]. Available: https://www.ppontime.co.th/รายละเอียด/Web_Und_Server_Und_Und_คืออะไร. [Accessed 13 มกราคม 2568].
- [3] E. P. Tutor, "5 อันดับ Web Server ที่ได้รับความนิยมในปี 2024," Expert Programming Tutor, [Online]. Available: https://expert-programming-tutor.com/tutorial/article/KE000935_5_Top_Web_Server_that_is_popular_in_2024.php. [Accessed 13 มกราคม 2568].
- [4] H. Pacific, "Open Source Web Server," Host Pacific, [Online]. Available: <https://www.hostpacific.com/open-source-webserver/>. [Accessed 13 มกราคม 2568].
- [5] N. Robin, Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5, 5th ed., Sebastopol: O'Reilly Media, 2018.
- [6] L. T. Luke Welling, PHP and MySQL Web Development, 5th ed., Boston: Addison-Wesley, 2016.
- [7] P. Documentation, "PHP Manual," PHP.net, [Online]. Available: <https://www.php.net/docs.php>. [Accessed 13 มกราคม 2025].
- [8] K. S. B. B. Bryan Basham, Head First Servlets and JSP, 2nd ed., Sebastopol: O'Reilly Media, 2008.

- [9] Oracle, "Java EE 7 API Documentation: javax.servlet.Servlet," Oracle.com, 2013. [Online]. Available: <https://docs.oracle.com/javaee/7/api/javax/servlet/Servlet.html>. [Accessed 13 มกราคม 2025].
- [10] L. M. Mario Casciaro, Node.js Design Patterns, 3rd ed., Birmingham: Packt Publishing, 2020.
- [11] O. Foundation, "Node.js v21.0.0 Documentation: All Modules," Nodejs.org, 2024. [Online]. Available: <https://nodejs.org/api/all.html>. [Accessed 13 มกราคม 2025].
- [12] B. W. K. Alan A. A. Donovan, The Go Programming Language, Upper Saddle River: Addison-Wesley, 2015.
- [13] T. G. Authors, "Go Documentation," Go.dev, 2024. [Online]. Available: <https://go.dev/doc/>. [Accessed 13 มกราคม 2025].
- [14] A. Freeman, Pro ASP.NET Core MVC, 6th ed., New York: Apress, 2020.
- [15] S. Corona, Nginx: A Practical Guide to High Performance, Birmingham: Packt Publishing, 2015.
- [16] D. Aivaliotis, Mastering Nginx, 2nd ed., Birmingham: Packt Publishing, 2017.
- [17] A. D. L. Smith, The Complete Nginx Cookbook, Sebastopol: O'Reilly Media, 2018.
- [18] NGINX, "NGINX Documentation," NGINX.org, [Online]. Available: <https://nginx.org/en/docs/>. [Accessed 13 มกราคม 2025].
- [19] J. Turnbull, The Docker Book, Lulu.com, 2014.
- [20] Docker, "Docker Documentation," Docker, [Online]. Available: <https://docs.docker.com/>. [Accessed 13 มกราคม 2025].

- [21] B. B. J. B. Kelsey Hightower, *Kubernetes: Up and Running*, 3rd ed., Sebastopol: O'Reilly Media, 2022.
- [22] M. Luksa, *Kubernetes in Action*, 2nd ed., Shelter Island: Manning Publications, 2022.
- [23] Kubernetes, "Kubernetes Documentation," Kubernetes, [Online]. Available: <https://kubernetes.io/docs/home/>. [Accessed 13 มกราคม 2025].
- [24] J. S. S. P. Paul Baka, *SSL/TLS Under Lock and Key*, 2nd ed., San Francisco: ITSec Publications, 2021.
- [25] OpenSSL, "SSL/TLS Documentation," OpenSSL, [Online]. Available: <https://docs.openssl.org/1.1.1/man7/ssl/>. [Accessed 13 มกราคม 2025].
- [26] L. cPanel, "cPanel – Web Hosting Control Panel," cPanel, LLC, [Online]. Available: <https://www.cpanel.net>. [Accessed 13 มกราคม 2025].
- [27] Plesk, "Web Hosting Control Panel & Management Software," Plesk, [Online]. Available: <https://www.plesk.com>. [Accessed 13 มกราคม 2025].
- [28] DirectAdmin, "DirectAdmin – Web Hosting Control Panel," DirectAdmin, [Online]. Available: <https://www.directadmin.com>. [Accessed 13 มกราคม 2025].
- [29] Webmin, "Webmin – Powerful and Flexible Web Administration," Webmin, [Online]. Available: <https://webmin.com/>. [Accessed 13 มกราคม 2025].
- [30] ISPConfig, "ISPConfig – Professional Hosting Control System," ISPConfig, [Online]. Available: <https://www.ispconfig.org/>. [Accessed 13 มกราคม 2025].

រាជរដ្ឋាភិបាល

ภาคผนวก ก

คู่มือการติดตั้งอย่างละเอียด

ขั้นตอนที่ 1: การติดตั้งโปรแกรมที่จำเป็น

ก่อนเริ่มใช้งานเว็บแอปพลิเคชัน ต้องติดตั้งโปรแกรมและเครื่องมือดังต่อไปนี้

Node.js ดาวน์โหลดได้ที่ [Node.js — Run JavaScript Everywhere](<https://nodejs.org/>)

Git ดาวน์โหลดได้ที่: [Git – Downloads](<https://git-scm.com/downloads>)

Docker ดาวน์โหลดได้ที่: [Docker: Accelerated Container Application Development](<https://www.docker.com/get-started>)

MariaDB หรือ MySQL

MariaDB: [MariaDB Downloads](<https://mariadb.org/download/>)

MySQL: [MySQL :: Download MySQL Installer](<https://dev.mysql.com/downloads/installer/>)

Go (Golang) ดาวน์โหลดได้ที่: [Go Downloads](<https://go.dev/dl/>)

PHP ดาวน์โหลดได้ที่: [PHP Downloads](<https://www.php.net/downloads>)

NGINX ดาวน์โหลดได้ที่: [NGINX Downloads](<https://nginx.org/en/download.html>)

Composer ดาวน์โหลดได้ที่: [Composer](<https://getcomposer.org/download/>)

Visual Studio Code หรือ IDE อื่น ๆ ดาวน์โหลดได้ที่: [Visual Studio Code](<https://code.visualstudio.com/>)

Postman ดาวน์โหลดได้ที่: [Postman](<https://www.postman.com/downloads/>)

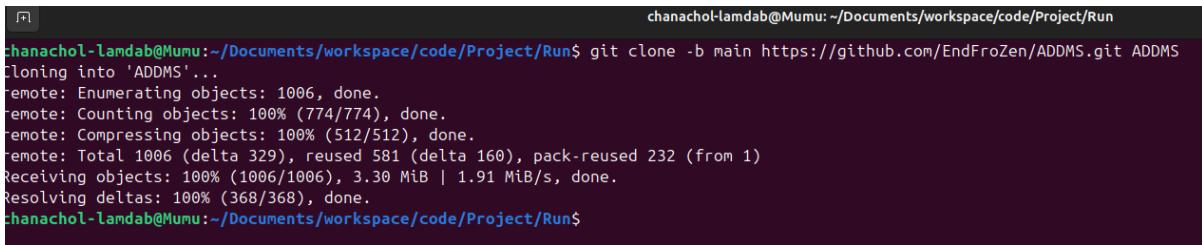
ขั้นตอนที่ 2 เตรียมพื้นที่สำหรับแอปพลิเคชัน



รูปที่ 1 สร้างโฟลเดอร์เปล่า

ตอนนี้ต้อง Run มาแล้วต่อมาเราจะทำการ clone github จากออนไลน์มาลงในโฟลเดอร์ ดังกล่าวโดยจะมีชื่อว่า ADDMS โดยใช้คำสั่ง

```
git clone -b main https://github.com/EndFroZen/ADDMS.git ADDMS
```



```
chanachol-lamdab@Mumu:~/Documents/workspace/code/Project/Run$ git clone -b main https://github.com/EndFroZen/ADDMS.git ADDMS
Cloning into 'ADDMS'...
remote: Enumerating objects: 1006, done.
remote: Counting objects: 100% (774/774), done.
remote: Compressing objects: 100% (512/512), done.
remote: Total 1006 (delta 329), reused 581 (delta 160), pack-reused 232 (from 1)
Receiving objects: 100% (1006/1006), 3.30 MiB | 1.91 MiB/s, done.
Resolving deltas: 100% (368/368), done.
chanachol-lamdab@Mumu:~/Documents/workspace/code/Project/Run$
```

รูปที่ 2 ตัวอย่างการใช้คำสั่ง

ขั้นตอนที่ 3 จะเกี่ยวกับการสร้างไฟล์ .env

ซึ่งใช้เก็บ ตัวแปรแวดล้อม (Environment Variables) ของโปรเจกต์ เช่น พอร์ตของเซิร์ฟเวอร์ ข้อมูลฐานข้อมูล และคีย์ลับต่าง ๆ โดยใช้คำสั่ง

```
cd ADDMS/backend/server/ && echo "DB_USER=your_database_user
DB_PASS=your_database_password
DB_HOST=your_database_host
DB_PORT=your_database_port
DB_NAME=your_database_name
JWT_SEC=your_jwt_secret
USER_Path="/path/to/your/corefolder"
Private_Base_Path="/path/to/your/home"
SERVER_HOST="your_server_domain"
SERVER_HOST_IP="your_server_ip"">>>.env
```

```
chanachol-lamdb@Mumu:~/Documents/workspace/code/Project/Run$ cd ADDMS/backend/server/ && echo "DB_USER=root
DB_PASS=4483
DB_HOST=localhost
DB_PORT=3306
DB_NAME=addms
JWT_SEC=CHINA
USER_Path="/home/chanachol-lamdb/Documents/workspace/code/Project/ADDMS/ADDMS_backend/corefolder"
Private_Base_Path = "/home/chanachol-lamdb"
SERVER_HOST = "endfrozen.site"
SERVER_HOST_IP = "127.0.0.1">.env
chanachol-lamdb@Mumu:~/Documents/workspace/code/Project/Run/ADDMS/backend/server$
```

รูปที่ 3 ตัวอย่างการใช้คำสั่ง

ขั้นตอนที่ 4 ทำการติดตั้งตัวข้อมูลจำเป็น

โดยใช้คำสั่ง npm install ในไฟลเดอร์ frontend และใช้คำสั่ง go mod tidy ในไฟลเดอร์ backend

```
chanachol-lamdb@Mumu:~/Documents/workspace/code/Project/Run$ cd ADDMS/frontend/
chanachol-lamdb@Mumu:~/Documents/workspace/code/Project/Run/ADDMS/frontend$ npm install

added 742 packages, and audited 743 packages in 14s

213 packages are looking for funding
  run `npm fund` for details

5 vulnerabilities (1 low, 4 moderate)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```

รูปที่ 4 ติดตั้งตัวข้อมูลจำเป็นในไฟลเดอร์ frontend

```
chanachol-lamdb@Mumu:~/Documents/workspace/code/Project/Run$ cd ADDMS/backend/server/
chanachol-lamdb@Mumu:~/Documents/workspace/code/Project/Run/ADDMS/backend/server$ go mod tidy
chanachol-lamdb@Mumu:~/Documents/workspace/code/Project/Run/ADDMS/backend/server$
```

รูปที่ 5 ตัวอย่างการใช้คำสั่ง

ขั้นตอนที่ 5 ทำการติดตั้ง database

ในตัวของ Mariadb ซึ่งที่ต้องกับ .env ที่สร้างโดยทางเราสร้าง database ที่ชื่อว่า addms

ขั้นตอนที่ 6 ต้องเข้าไปทำ sudoers

โดยการเพิ่มตัวของ [username] ALL=(ALL) NOPASSWD: /bin/systemctl reload nginx, /bin/systemctl restart nginx, /bin/systemctl stop nginx, /bin/systemctl start nginx โดยใช้คำสั่งเบิดไฟล์ sudo visudo ขั้นตอนนี้สำคัญมาก

```
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
chanachol-lamda ALL=(ALL) NOPASSWD: /usr/sbin/nginx -t, /usr/sbin/nginx -s reload
# See sudoers(5) for more information on "@include" directives:
@include /etc/sudoers.d
```

รูปที่ 6 ตัวอย่างการใช้คำสั่ง

ขั้นตอนที่ 7 สั่งให้ nginx มองไฟล์ภายนอก root

โดยใช้คำสั่ง sudo nano /etc/nginx/nginx.conf และไปเพิ่ม include /home/user/mynginx/conf.d/*.conf; โดยจะให้เปลี่ยนเป็น โฟลเดอร์หลักของคุณ

```
# gzip_types text/plain text/css application/json application/javascript tex
##
## Virtual Host Configs
##

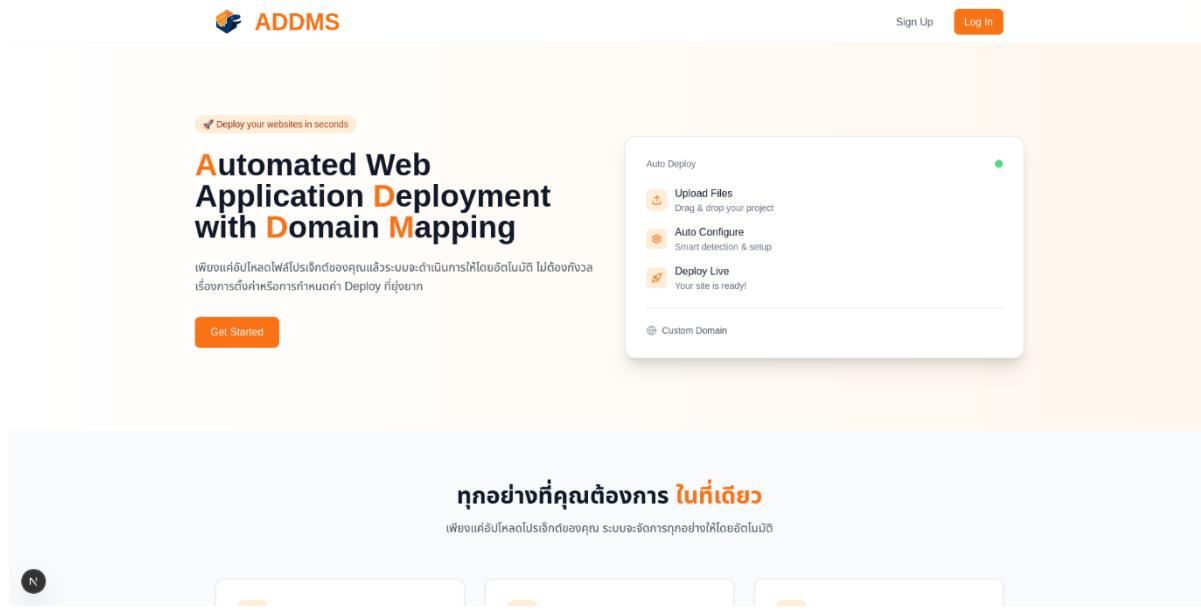
include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
include /home/chanachol-lamda/nginx-users/*.conf;
}

#mail {
#    # See sample authentication script at:
#    # http://wiki.nginx.org/ImapAuthenticateWithApachePhpScript
#
#    # auth_http localhost/auth.php;
#    # pop3_capabilities "TOP" "USER";
```

รูปที่ 7 ตัวอย่างการใช้คำสั่ง

ขั้นตอนที่ 8 สั่งทำงาน ADDMS

การพัฒนาเว็บแอปพลิเคชันตั้งกล่าวได้ออกแบบให้สามารถทำงานได้ทั้งในส่วนของ Frontend และ Backend โดยในส่วนของ Frontend ใช้คำสั่ง npm run build และ npm run start สำหรับการสร้างและเปิดใช้งานระบบ ขณะที่ส่วนของ Backend ใช้คำสั่ง go run . เพื่อสั่งให้ระบบเริ่มทำงาน ทั้งนี้ การออกแบบลักษณะตั้งกล่าวมีวัตถุประสงค์เพื่ออำนวยความสะดวกในการ Deploy บนระบบปฏิบัติการ Linux ให้มีความง่าย รวดเร็ว และมีประสิทธิภาพมากยิ่งขึ้น นอกจากนี้ หากผู้ใช้งานมีความประสงค์ที่จะติดตั้ง ADDMS ลงบนเซิร์ฟเวอร์ ระบบที่พัฒนาขึ้นสามารถรองรับการดำเนินการตั้งกล่าวได้อย่างราบรื่นและปลอดภัย โดยมีการจัดเตรียมเครื่องมือและกระบวนการที่ช่วยลดความซับซ้อนของขั้นตอนการทำงาน เพื่อให้การใช้งานระบบเว็บแอปพลิเคชันมีความสมบูรณ์และสามารถเข้าถึงได้จากภายนอก ผู้ใช้งานจำเป็นต้องดำเนินการจดทะเบียนชื่อโดเมน (Domain Name) และเชื่อมโยงกับเว็บแอปพลิเคชันผ่าน Public IP Address ขั้นตอนนี้จะช่วยให้เว็บไซต์สามารถเผยแพร่และให้บริการแก่ผู้ใช้งานได้อย่างมีประสิทธิภาพและสอดคล้องกับมาตรฐานการใช้งานเว็บแอปพลิเคชันในปัจจุบัน สามารถดูขั้นตอนที่ละเอียดได้ที่ <https://github.com/EndFroZen/ADDMS?tab=readme-ov-file#readme>



รูปที่ 8 หน้าโฮมเพจ

ประวัติผู้เขียน

ชื่อ-นามสกุล นายพงศ์ภัค ยศศันนิห์

วัน/เดือน/ปีเกิด วันที่ 19 ธันวาคม 2546 จังหวัดเชียงราย

ที่อยู่ 32/1 หมู่ 2 ตำบลแม่ฟ้าหลวง อำเภอแม่ฟ้าหลวง จังหวัดเชียงราย

ประวัติการศึกษา

- จบการศึกษาระดับประถมศึกษาตอนต้นและตอนปลาย จากโรงเรียนบ้านห้วยไร่สามัคคี ตำบลแม่ฟ้าหลวง อำเภอแม่ฟ้าหลวง จังหวัดเชียงราย
- จบการศึกษาระดับมัธยมศึกษาตอนต้น จากโรงเรียนบ้านห้วยไร่สามัคคี ตำบลแม่ฟ้าหลวง อำเภอแม่ฟ้าหลวง จังหวัดเชียงราย
- จบการศึกษาระดับมัธยมศึกษาตอนปลาย จากโรงเรียนแม่จันวิทยาคม ตำบลป่าซาง อำเภอแม่จัน จังหวัดเชียงราย
- ระดับปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยพะเยา ตำบลแม่กำเ umo อำเภอเมือง จังหวัดพะเยา
- Email Address : pongsapak.ykt@gmail.com

ชื่อ-นามสกุล นายชนะชล ลำดับ

วัน/เดือน/ปีเกิด 16 พฤศจิกายน 2546 จังหวัดสุโขทัย

ที่อยู่ 250 หมู่ 4 ตำบลแม่สิน อำเภอครีสชนาลัย จังหวัดสุโขทัย

ประวัติการศึกษา

- จบการศึกษาระดับประถมศึกษาตอนต้นและตอนปลาย จากโรงเรียนบ้านห้วยบี๊ ตำบลแม่สิน อำเภอครีสชนาลัย จังหวัดสุโขทัย
- จบการศึกษาระดับมัธยมศึกษาตอนต้น จากโรงเรียนเมืองเชลียง ตำบลหาดเลี้ยง อำเภอครีสชนาลัย จังหวัดสุโขทัย
- จบการศึกษาระดับมัธยมศึกษาตอนปลาย จากโรงเรียนเมืองเชลียง ตำบลหาดเลี้ยง อำเภอครีสชนาลัย จังหวัดสุโขทัย
- ระดับปริญญาตรี สาขาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยพะเยา ตำบลแม่กำเumo อำเภอเมือง จังหวัดพะเยา
- Email Address : Ef.chanachol@gmail.com