



รายงานสหกิจศึกษาฉบับสมบูรณ์

เว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์
Web Application for Page and Post Migration

นางสาวสุวัจนี จันทระโชติ

ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิชาวิศวกรรมสารสนเทศ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560



รายงานสหกิจศึกษาฉบับสมบูรณ์

เว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์
Web Application for Page and Post Migration

นางสาวสุวัจนี จันทร์โชติ

ภาควิชาวิศวกรรมคอมพิวเตอร์ สาขาวิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการสหกิจศึกษา เว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์
ชื่อ-สกุล นักศึกษา นางสาวสุวัจน์ จันทร์โชติ
คณะ วิศวกรรมศาสตร์ ภาควิชา วิศวกรรมสารสนเทศ
ชื่อ-สกุล อาจารย์นิเทศ ผศ. มยุรี เลิศเวชกุล
ชื่อ-สกุล ผู้นิเทศงาน นางสาวอภิญญา สิริบุญชร
ชื่อสถานประกอบการ บริษัทพรอนโต้ กรุ๊ป จำกัด

บทคัดย่อ

โครงการนี้เป็นการพัฒนาโปรแกรมเครื่องมือที่ช่วยสำหรับการย้ายเนื้อหาเว็บไซต์ไปยังเว็บไซต์ใหม่ที่จัดการด้วยโปรแกรมสำเร็จรูปเวิร์ดเพรสของบริษัทพรอนโต้ กรุ๊ป จำกัด เพื่อให้ช่วยให้การบริการลูกค้าในการย้ายเว็บไซต์สามารถดำเนินการได้สะดวกรวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น โดยผู้ใช้เลือกรายการหน้าเว็บเพจที่ต้องการคัดลอกใส่ในไฟล์สกุลซีเอสวีพร้อมกับระบุส่วนของเนื้อหาที่ต้องการใช้แสดงผลในเว็บไซต์ใหม่ โปรแกรมเครื่องมือที่สร้างขึ้นก็จะทำการอ่านเนื้อหาในเว็บไซต์ทั้งหมดทำการแปลงรูปแบบของไฟล์ให้อยู่ในรูปแบบที่พร้อมใช้งานบนโปรแกรมเวิร์ดเพรส นอกจากนี้ยังมีคุณสมบัติการทำงานอื่นๆ อาทิเช่น การดาวน์โหลดรูปภาพจากเว็บไซต์เก่ามาอัปโหลดขึ้นเว็บไซต์ใหม่ เปลี่ยนพารามิเตอร์ให้ชี้ไปยังตำแหน่งที่ถูกต้อง โดยเครื่องมือนี้จะช่วยลดขั้นตอนการทำงาน ลดโอกาสการเกิดความผิดพลาด และยังช่วยประหยัดเวลาในการทำงานอีกด้วย

คำสำคัญ : เว็บแอปพลิเคชัน โปรแกรมเครื่องมือ การย้ายเว็บไซต์ เวิร์ดเพรส

Co-operative Title: Web Application for Page and Post Migration
Student intern name: Miss Suwajanee Janchote
Faculty: Engineering Department: Information Engineering
Advisor name: Asst.Prof. Mayuree Lertwatechakul
Mentor name: Miss Apinya Sirikunchorn
Company: Pronto Group LTD.

ABSTRACT

This project is to develop a program tool for web-content migration from the origin website to the new one managed by WordPress. The main objective of this project to make the procedure of website migration one of the Pronto Group Corporation services to be more quicker and efficiently. This tools can handle all the processes of website migration, a user just choose pages and add it into .CSV file and then select the content from HTML files after that the tool will read the content and convert it into WordPress ready-to-use format. Moreover, there are other necessary features such as downloading images from the old site and upload them to the new one, change the images' path to be the correct relative path. As a result, the tool could improve the website migration procedure by reducing work process, reducing error and saving time as well.

Keywords: Web Application, Program tool, Website migration, Wordpress

กิตติกรรมประกาศ

โครงการเว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์เป็นโครงการของบริษัทพรอนโต้กรุ๊ป จำกัด โดยโครงการนี้สำเร็จลงได้ด้วยดีด้วยการสนับสนุนจากผู้มีพระคุณหลายท่าน ขอขอบคุณคุณอภิญา สิริบุญจร และพี่ๆ พนักงานบริษัทผู้ให้คำแนะนำ ให้ความช่วยเหลือเกี่ยวกับระบบงาน และให้ประการณ์ที่ดีแก่ผู้จัดทำ ขอขอบคุณ ผศ. มยุรี เลิศเวชกุล ผู้เป็นอาจารย์นิเทศโครงการสหกิจศึกษาที่ให้ความเมตตากรุณาคอยติดตามความคืบหน้าของโครงการและให้คำแนะนำแก่ผู้จัดทำมาโดยตลอดพร้อมทั้งอาจารย์ในคณะทุกท่านที่ได้ประสิทธิ์ประสาทความรู้แก่ผู้จัดทำและขอบคุณเพื่อนๆ ทุกคนที่เป็นกำลังใจและให้การช่วยเหลือทุกอย่างตลอดการทำโครงการนี้

สุวัจน์ จันทร์โชติ



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	VIII
สารบัญตาราง.....	XI
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.5 วิธีการดำเนินงาน.....	2
1.6 ระยะเวลาการดำเนินงาน.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 เว็บแอปพลิเคชัน (WEB APPLICATION).....	4
2.1.1 ข้อดีของเว็บแอปพลิเคชัน.....	4
2.1.2 ข้อเสียของเว็บแอปพลิเคชัน.....	4
2.2 เวิร์ดเพรส (WORDPRESS).....	5
2.2.1 องค์ประกอบพื้นฐานของเวิร์ดเพรส.....	5
2.2.2 ข้อดีของการสร้างเว็บด้วยเวิร์ดเพรส.....	5
2.2.3 หน้า (Page).....	6
2.2.4 บทความ (Post).....	6
2.3 ไพธอน (PYTHON).....	6
2.3.1 จุดเด่นของภาษาไพธอน.....	7

สารบัญ (ต่อ)

	หน้า
2.4 HTML (HYPER TEXT MARKUP LANGUAGE)	7
2.4.1 โครงสร้างของภาษา HTML.....	7
2.5 CSS (CASCADING STYLE SHEETS).....	8
2.5.1 ข้อดีของ CSS.....	8
2.6 DJANGO FRAMEWORK	9
2.6.1 คุณสมบัติของ Django Framework.....	9
2.6.2 ข้อดีของ Django Framework.....	10
2.6.3 ข้อเสียของ Django Framework	10
2.7 BOOTSTRAP	10
2.8 CSV (COMMA-SEPARATED VALUES)	10
2.8.1 โปรแกรมที่สนับสนุนไฟล์ CSV	11
2.9 API (APPLICATION PROGRAMMING INTERFACE).....	11
2.9.1 กระบวนการทำงานของ API	11
2.9.2 ประโยชน์ของ API.....	12
2.10 REST API	12
2.10.1 พื้นฐานของ REST.....	12
2.11 JSON.....	13
2.11.1 รูปแบบของ JSON	13
2.12 โฟลว์ชาร์ต (FLOWCHART).....	15
2.12.1 ประเภทของโฟลว์ชาร์ต.....	15
2.12.2 ประโยชน์ของโฟลว์ชาร์ต	15
2.12.3 สัญลักษณ์ของโฟลว์ชาร์ต.....	16
2.13 ยูสเคสไดอะแกรม (USE CASE DIAGRAM).....	17
2.13.1 ความสัมพันธ์ (Relationships).....	17
2.13.2 ประโยชน์ของยูสเคสไดอะแกรม.....	17
2.13.3 สัญลักษณ์ของยูสเคสไดอะแกรม.....	18
2.14 ซีควเอนซ์ไดอะแกรม (SEQUENCE DIAGRAM).....	19

สารบัญ (ต่อ)

	หน้า
2.14.1 สัญลักษณ์ของซีควেনซ์ไดอะแกรม	19
บทที่ 3 วิธีดำเนินการ.....	21
3.1 การวิเคราะห์ความต้องการจากผู้ใช้งาน	21
3.2 การออกแบบระบบ.....	22
3.2.1 ยูสเคสไดอะแกรม (Use Case Diagram)	22
3.2.2 โฟลว์ชาร์ต (Flowchart).....	23
3.2.2.1 โฟลว์ชาร์ตการใช้งานเว็บแอปพลิเคชัน.....	23
3.2.2.2 โฟลว์ชาร์ตฟังก์ชัน Export ของหน้า (Page).....	24
3.2.2.3 โฟลว์ชาร์ตฟังก์ชัน Import ของหน้าเว็บ (Page).....	25
3.2.2.4 โฟลว์ชาร์ตฟังก์ชัน Remove Element.....	26
3.2.2.5 โฟลว์ชาร์ตฟังก์ชัน Export ของบทความ (Post).....	27
3.2.2.6 โฟลว์ชาร์ตฟังก์ชัน Import ของบทความ (Post).....	28
3.2.3 ซีควেনซ์ไดอะแกรม (Sequence Diagram).....	29
3.2.3.1 ซีควেনซ์ไดอะแกรมของฟังก์ชัน Export.....	29
3.2.3.2 ซีควেনซ์ไดอะแกรมของฟังก์ชัน Import.....	29
3.2.3.3 ซีควেনซ์ไดอะแกรมของฟังก์ชัน Remove Element	30
3.3 การพัฒนาระบบ.....	31
3.3.1 ขั้นตอนการพัฒนาระบบ	31
3.3.2 ตัวช่วยในการพัฒนาระบบ	31
3.3.3 การพัฒนาฟังก์ชันต่างๆ.....	31
3.3.3.1 Page Export	32
3.3.3.2 Page Import.....	33
3.3.3.3 Remove Element.....	36
3.3.3.4 Post Export.....	37
3.3.3.5 Post Import	39
บทที่ 4 ผลการดำเนินการ	42

สารบัญ (ต่อ)

	หน้า
4.1 การตั้งค่าเว็บไซต์ก่อนการใช้งานเครื่องมือ	42
4.2 การเตรียมไฟล์ซีเอสวีสำหรับการใช้งาน	43
4.3 ตัวอย่างการใช้งานเว็บแอปพลิเคชัน	43
4.3.1 Page Export	44
4.3.2 Page Import	45
4.3.3 Remove Element	47
4.3.4 Post Export	49
4.3.5 Post Import	51
บทที่ 5 สรุปผลการดำเนินการ	54
5.1 สรุปผลการดำเนินการ	54
5.2 ปัญหาที่พบ	54
5.3 แนวทางการแก้ไขปัญหา	54
5.4 ข้อเสนอแนะ	55
เอกสารอ้างอิง	56
ภาคผนวก	59
ภาคผนวก ก. โปสเตอร์	60
ภาคผนวก ข. การติดตั้งภาษาไพธอน	62

สารบัญภาพ

	หน้า
ภาพที่ 2.1 สัญลักษณ์ของเวิร์ดเพรส.....	5
ภาพที่ 2.2 สัญลักษณ์ของภาษาไพธอน.....	6
ภาพที่ 2.3 โครงสร้างของภาษา HTML.....	8
ภาพที่ 2.4 โครงสร้างไฟล์เตอร์ของ Django Project.....	9
ภาพที่ 2.5 หน้าแรกของ API.....	11
ภาพที่ 2.6 JSON รูปแบบ Object.....	13
ภาพที่ 2.7 JSON รูปแบบ Array.....	13
ภาพที่ 2.8 JSON รูปแบบ Value.....	14
ภาพที่ 2.9 JSON รูปแบบ String.....	14
ภาพที่ 2.10 JSON รูปแบบ Number.....	15
ภาพที่ 2.11 ลักษณะของซีควเอนซ์ไดอะแกรม.....	19
ภาพที่ 3.1 แผนภาพวงจรการพัฒนาระบบ.....	21
ภาพที่ 3.2 ยูสเคสไดอะแกรม.....	22
ภาพที่ 3.3 โพล์ชาร์ตการใช้งานเว็บแอปพลิเคชัน.....	23
ภาพที่ 3.4 โพล์ชาร์ตฟังก์ชัน Export หน้าเว็บ (Page).....	24
ภาพที่ 3.5 โพล์ชาร์ตฟังก์ชัน Import ของหน้าเว็บ (Page).....	25
ภาพที่ 3.6 โพล์ชาร์ตฟังก์ชัน Remove Element.....	26
ภาพที่ 3.7 โพล์ชาร์ตฟังก์ชัน Export บทความ (Post).....	27
ภาพที่ 3.8 โพล์ชาร์ตฟังก์ชัน Import ของบทความ (Post).....	28
ภาพที่ 3.9 ซีควเอนซ์ไดอะแกรมของฟังก์ชัน Export.....	29
ภาพที่ 3.10 ซีควเอนซ์ไดอะแกรมของฟังก์ชัน Import.....	30
ภาพที่ 3.11 ซีควเอนซ์ไดอะแกรมของฟังก์ชัน Remove Element.....	30
ภาพที่ 3.12 โค้ดการทำงานของฟังก์ชัน Page Export.....	32
ภาพที่ 3.13 โค้ดการทำงานของฟังก์ชัน Page Import.....	33
ภาพที่ 3.14 โค้ดฟังก์ชันการจัดรูปลิงก์สำหรับดาวนโหลดรูปภาพ.....	34
ภาพที่ 3.15 โค้ดฟังก์ชันการดาวนโหลดและอัปโหลดรูปภาพสำหรับ Page.....	35
ภาพที่ 3.16 โค้ดฟังก์ชันการสร้าง Page.....	36

สารบัญภาพ (ต่อ)

	หน้า
ภาพที่ 3.17 โค้ดการทำงานของฟังก์ชัน Remove Element.....	36
ภาพที่ 3.18 โค้ดฟังก์ชันการจัดการค่าวันที่เผยแพร่ของบทความ	37
ภาพที่ 3.19 โค้ดการทำงานของฟังก์ชัน Post Export	38
ภาพที่ 3.20 โค้ดการทำงานของฟังก์ชัน Post Import.....	39
ภาพที่ 3.21 โค้ดฟังก์ชันการดาวน์โหลดและอัปโหลดรูปภาพสำหรับ Post	40
ภาพที่ 3.22 โค้ดฟังก์ชันการสร้าง Post.....	41
ภาพที่ 4.1 การตั้งค่าเว็บไซต์ก่อนการใช้งานเครื่องมือ.....	42
ภาพที่ 4.2 ตัวอย่างไฟล์ CSV สำหรับคัดลอกหน้าเว็บ (Page)	43
ภาพที่ 4.3 ตัวอย่างไฟล์ CSV สำหรับคัดลอก Post	43
ภาพที่ 4.4 เมนูการใช้งานของเว็บแอปพลิเคชัน.....	43
ภาพที่ 4.5 ตัวอย่างการใช้งาน Page Export	44
ภาพที่ 4.6 วิธีการดู tag HTML จากหน้าเว็บ	44
ภาพที่ 4.7 ข้อมูล JSON ที่ได้ฟังก์ชัน Page Export.....	45
ภาพที่ 4.8 ตัวอย่างการใช้งาน Page Import.....	45
ภาพที่ 4.9 หน้าการแสดงผลหลังจากสร้าง Page.....	46
ภาพที่ 4.10 Page ที่สร้างจากฟังก์ชัน Page Import.....	46
ภาพที่ 4.11 เปรียบเทียบ Page เว็บไซต์เก่าและเว็บไซต์ใหม่.....	47
ภาพที่ 4.12 ตัวอย่างการใช้งาน Remove Element.....	48
ภาพที่ 4.13 เปรียบเทียบไฟล์ JSON ก่อนและหลังใช้ฟังก์ชัน Remove Element.....	48
ภาพที่ 4.14 เปรียบเทียบหน้าเว็บที่สร้างก่อนและหลังใช้ฟังก์ชัน Remove Element.....	49
ภาพที่ 4.15 ตัวอย่างการใช้งาน Post Export.....	50
ภาพที่ 4.16 ข้อมูล JSON ที่ได้ฟังก์ชัน Post Export	50
ภาพที่ 4.17 ตัวอย่างการใช้งาน Post Import	51
ภาพที่ 4.18 หน้าการแสดงผลหลังจากสร้าง Post.....	51
ภาพที่ 4.19 Post ที่สร้างจากฟังก์ชัน Post Import	52
ภาพที่ 4.20 เปรียบเทียบ Post เว็บไซต์เก่าและเว็บไซต์ใหม่	53
ภาพที่ ก.1 โปสเตอร์.....	61

สารบัญญภาพ (ต่อ)

	หน้า
ภาพที่ ข.1 หน้าเว็บไซต์สำหรับดาวน์โหลดตัวติดตั้งภาษา	63
ภาพที่ ข.2 ไฟล์ที่ใช้ในการติดตั้งภาษา.....	63
ภาพที่ ข.3 หน้าแรกของตัวติดตั้งภาษา	64
ภาพที่ ข.4 หน้าเลือกที่อยู่ตัวติดตั้งภาษา.....	64
ภาพที่ ข.5 หน้าเลือกพีแอร์ที่ใช้.....	65
ภาพที่ ข.6 หน้ากำลังติดตั้ง.....	65
ภาพที่ ข.7 หน้าติดตั้งเสร็จสมบูรณ์.....	66
ภาพที่ ข.8 ตรวจสอบการติดตั้ง.....	66



สารบัญตาราง

	หน้า
ตารางที่ 1.1 ระยะเวลาการดำเนินงาน	3
ตารางที่ 2.1 สัญลักษณ์ในโพล์ชาร์ต	16
ตารางที่ 2.2 สัญลักษณ์ในยูสเคสไดอะแกรม	18
ตารางที่ 2.3 สัญลักษณ์ในซีควเอนซ์ไดอะแกรม	19



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

เนื่องจากบริษัทพรอนโต้กรุ๊ป จำกัด เป็นบริษัทรับทำเว็บไซต์ที่มีลูกค้าหลายลักษณะ ทั้งลูกค้าที่มาเริ่มต้นสร้างเว็บไซต์ใหม่และลูกค้าที่มีเว็บไซต์เป็นของตัวเองอยู่แล้วย้ายมาใช้บริการกับทางบริษัท โดยลูกค้าที่มีเว็บไซต์เดิมอยู่แล้วอาจจะมีความต้องการที่จะคัดลอกเนื้อหาจากหน้าและบทความของเว็บไซต์เก่ามายังเว็บไซต์ใหม่ด้วย ซึ่งการย้ายหน้าและบทความดังกล่าวโดยปกติจะเป็นหน้าที่ของทีม Production ในการสร้าง โดยที่การสร้างแต่ละหน้ามีหลายขั้นตอนตั้งแต่การคัดลอกหัวเรื่อง เนื้อหา การดาวน์โหลดรูปภาพจากเว็บไซต์เก่ามาอัปโหลดที่เว็บไซต์ใหม่และเปลี่ยนพารามิเตอร์ในเนื้อหาของหน้าและบทความนั้นๆ ให้เป็นพารามิเตอร์ที่ถูกต้อง สำหรับการสร้างบทความอาจจะมีกำหนดวันที่เผยแพร่บทความหรือกำหนดรูปภาพหน้าปกของบทความ (Feature image) เพิ่มเติมขึ้นอยู่กับความต้องการของผู้ดูแลจัดการเว็บไซต์นั้นๆ ซึ่งหากมีหน้าที่ต้องการคัดลอกจำนวนมากก็จะเป็นหน้าที่ของทีม Developer ในการคัดลอกเนื้อหาจากฐานข้อมูลหรือสร้างสคริปต์สำหรับคัดลอกเนื้อหาจากหน้าเว็บในกรณีที่ไม่สามารถเข้าถึงฐานข้อมูลเว็บไซต์เก่าได้

จึงได้คิดพัฒนาเครื่องมือสำหรับการย้ายหน้าและบทความเว็บไซต์ เป็นเว็บแอปพลิเคชันที่รวบรวมขั้นตอนที่กล่าวมาข้างต้นพร้อมทั้งฟีเจอร์เพิ่มเติมตามที่ผู้ดูแลโปรเจกต์ต้องการ มีการออกแบบให้ใช้งานง่ายเพื่อรองรับการใช้งานจากทีม Production และสามารถใช้ได้กับทุกงานที่มีลักษณะเดียวกัน

1.2 วัตถุประสงค์

1. รวมขั้นตอนพื้นฐานในการคัดลอกเนื้อหาเว็บไซต์ไว้ในเว็บแอปพลิเคชัน
2. สามารถคัดลอกเนื้อหาจากหน้าเว็บหลายหน้าในการทำงานครั้งเดียว
3. เพื่อสร้างเครื่องมือที่ช่วยให้พนักงานทำงานได้สะดวกและรวดเร็วมากขึ้น
4. ช่วยให้พนักงานทำงานได้อย่างมีประสิทธิภาพ ลดโอกาสการเกิดความผิดพลาด
5. เพื่อประยุกต์ใช้เทคโนโลยีกับระบบการทำงานได้
6. สามารถใช้กับงานอื่นๆ ที่มีลักษณะเดียวกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงการ

1. สามารถคัดลอกเนื้อหาจากหน้าเว็บไซต์เก่ามาสร้างหน้าหรือบทความในเว็บไซต์ใหม่ได้
2. มีฟีเจอร์เพิ่มเติมจากขั้นตอนพื้นฐานตาม Requirement ของผู้ดูแลโปรเจกต์และพนักงาน
3. เครื่องมือที่พัฒนาจะใช้ภาษาไพธอนเป็นภาษาในการพัฒนา โดยใช้ Django Framework
4. เว็บไซต์ใหม่ที่ต้องการสร้างหน้าและบทความเป็นเว็บไซต์ที่จัดการด้วยเวิร์ดเพรส
5. มีการทำงานร่วมกับปลั๊กอิน WordPress REST API ของเวิร์ดเพรสสำหรับการสร้างหน้า

และบทความ

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มทักษะในการสื่อสารพูดคุยในการรับ Requirement จากพนักงานคนอื่น
2. สามารถวางแผนการพัฒนาเครื่องมือได้อย่างเป็นขั้นตอน
3. สามารถพัฒนาเพื่อช่วยเหลืองานของพนักงานในบริษัท
4. ได้รับความรู้เกี่ยวกับเทคโนโลยีต่างๆ ที่ใช้ในการพัฒนาระบบ
5. สามารถเขียนโปรแกรมภาษาไพธอนในระดับพื้นฐานได้

1.5 วิธีการดำเนินงาน

1. ศึกษาขั้นตอนพื้นฐานในการคัดลอกเนื้อหา
2. พูดคุยกับผู้ใช้เพื่อรับ Requirement เพิ่มเติม
3. ศึกษาเทคโนโลยีที่ใช้ในการพัฒนา
4. วิเคราะห์และออกแบบระบบ
 - แบ่งการทำงานของระบบออกเป็นฟังก์ชัน
 - ออกแบบการทำงานของแต่ละฟังก์ชัน
 - ออกแบบหน้าตาของเว็บแอปพลิเคชันสำหรับผู้ใช้
5. พัฒนาเครื่องมือในฟังก์ชันต่างๆ ตามระบบที่ออกแบบไว้ในขั้นตอนที่ 3
6. ทดสอบการใช้งานและแก้ไขข้อผิดพลาด
7. สรุปผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 ระยะเวลาการดำเนินงาน

ตารางที่ 1.1 ระยะเวลาการดำเนินงาน

การดำเนินงาน	สิงหาคม.				กันยายน				ตุลาคม				พฤศจิกายน			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษาขั้นตอนการทำงานพื้นฐาน	■															
2. รับ Requirement จากผู้ใช้	■	■														
3. ศึกษาเทคโนโลยีที่ใช้	■	■	■	■												
4. ออกแบบฟังก์ชันการทำงาน				■	■											
5. พัฒนาเครื่องมือตามทีออกแบบ						■	■	■	■							
6. ทดสอบการใช้งานเครื่องมือ										■						
7. พัฒนาพีเจอร์เพิ่มเติมและแก้ไขข้อผิดพลาด											■					
8. จัดเตรียมแสดงตัวอย่างการใช้งานแก่ผู้ใช้งาน (Demo)													■			
9. จัดทำรูปเล่มสหกิจศึกษา															■	■

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 เว็บแอปพลิเคชัน (Web Application)

คือ แอปพลิเคชัน (Application) ที่ถูกเขียนขึ้นมาเพื่อเป็นเบราว์เซอร์ (Browser) สำหรับการใช้งานเว็บเพจ (Webpage) ต่างๆ ซึ่งถูกปรับแต่งให้แสดงผลแต่ส่วนที่จำเป็น เพื่อเป็นการลดทรัพยากรในการประมวลผลของตัวเครื่องสมาร์ทโฟน (Smartphone) หรือแท็บเล็ต (Tablet) ทำให้โหลดหน้าเว็บไซต์ได้เร็วขึ้น อีกทั้งผู้ใช้งานยังสามารถใช้งานผ่านอินเทอร์เน็ต (Internet) และอินทราเน็ต (Intranet) ในความเร็วต่ำได้

2.1.1 ข้อดีของเว็บแอปพลิเคชัน

1. เหมาะกับองค์กรขนาดเล็กเพราะมีค่าใช้จ่ายต่ำกว่าและคิดค่าใช้จ่ายตามจำนวนการใช้งานจริง
2. การใช้งานในองค์กรทำได้ง่ายเพียงแคมีเว็บเบราว์เซอร์ก็สามารถใช้งานได้
3. ข้อมูลจัดเก็บที่เดียว ง่ายต่อการจัดการ และไม่เกิดความซ้ำซ้อน
4. ไม่ต้องการเครื่องคอมพิวเตอร์ประสิทธิภาพสูงซึ่งมีราคาแพง
5. ทำงานได้ทุกที่เนื่องจากสามารถล็อกอินเข้าใช้งานได้เลยโดยไม่ต้องติดตั้งโปรแกรม
6. ไม่จำเป็นต้องมีบุคลากรด้านเทคนิค เนื่องจากผู้ให้บริการดูแลเซิร์ฟเวอร์และการบำรุงรักษาเองทั้งหมด
7. ส่วนมากใช้ได้หลากหลาย Platform ทั้ง Windows, Linux และ Mac ทำให้องค์กรสามารถเลือกใช้บางเครื่องเป็น Linux ได้เพื่อลดค่าใช้จ่ายด้านลิขสิทธิ์
8. เชื่อมต่อกับเว็บแอปพลิเคชันหรือบริการออนไลน์อื่นๆ ได้ง่าย

2.1.2 ข้อเสียของเว็บแอปพลิเคชัน

1. รูปร่างหน้าตาและการใช้งานมีได้จำกัดอาจไม่เหมาะกับงานที่ต้องการรูปแบบโปรแกรมที่แตกต่างจากโปรแกรมทั่วไป เช่น โปรแกรมตกแต่งรูป โปรแกรมตัดต่อวิดีโอ
2. เว็บแอปพลิเคชันหลายๆ ตัวต้องการอินเทอร์เน็ตในการใช้งานเสมอ

2.2 เว็รดเพรส (WordPress)

เป็น Open Source Web Software ที่สามารถติดตั้งบนเว็บเซิร์ฟเวอร์ของผู้ใช้งานเพื่อสร้างเว็บไซต์, Blog หรือ Community ตอนเริ่มแรกเว็บเพรสเป็นเครื่องมือไว้สำหรับสร้าง Blog แต่ได้รับการพัฒนามาเรื่อยๆ จนสามารถสร้างเป็นเว็บไซต์หรือเว็บ Community ได้ โดยมีระบบจัดการบทความหรือ Content Management System (CMS) ทำให้ง่ายต่อการใช้งาน ซึ่งเว็บเพรสสร้างขึ้นโดย Matt Mullenweg และ Mike Little ในปี 2003



ภาพที่ 2.1 สัญลักษณ์ของเว็บเพรส

2.2.1 องค์ประกอบพื้นฐานของเว็บเพรส

เว็บเพรสเขียนด้วยภาษา PHP และใช้ Apache, MySQL และ PhpMyAdmin ในการรันเป็นเซิร์ฟเวอร์ โดยเว็บเพรสจะรันอยู่บนฝั่งเซิร์ฟเวอร์ ผู้ใช้งานจึงต้องมีเครื่องมือจำลองเซิร์ฟเวอร์

2.2.2 ข้อดีของการสร้างเว็บด้วยเว็บเพรส

1. ตัวโปรแกรมเว็บเพรสสามารถโหลดและใช้งานฟรี
2. ติดตั้งง่าย
3. มีระบบ Plugin มากมายและเปิดปิดการใช้งานได้ง่าย
4. มีระบบ RSS Feeds ที่มีคุณภาพ
5. มีระบบตั้งค่ากำหนดให้แสดง URL เองได้
6. มีระบบ Post Categories สำหรับแยกหมวดหมู่ Post
7. มีระบบ Page ที่สามารถ ใช้สร้างเป็นหน้า Static Page สำหรับเว็บ
8. มีระบบ Media ที่สามารถเพิ่มรูปภาพ วิดีโอ เพลง รวมถึงไฟล์ต่างๆ ได้
9. มี Interface การใช้งานที่ง่ายเหมาะสำหรับผู้ใช้งานทั่วไป
10. มีระบบ Post และ Page ที่สามารถตั้งเวลา Post กำหนดผู้เขียนและสามารถป้องกันการดูด้วยรหัสผ่านได้
11. สามารถแก้ไขปรับเปลี่ยน Template, Layout รวมถึง Content ได้อย่างสะดวกและรวดเร็ว
12. มีรูปแบบ Theme ให้เลือกใช้มากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. สามารถเชื่อมต่อเข้ากับ Social Media ได้สะดวกและหลากหลาย

14. เป็นมิตรต่อการทำ SEO สามารถปรับแต่ง On-Page SEO ได้อย่างสะดวกรวดเร็ว

2.2.3 หน้า (Page)

คือการเผยแพร่บทความเฉพาะเรื่องหรือหัวข้อ จะไม่มีหมวดหมู่ (Categories) หรือแท็ก (Tags) และไม่มีการเรียงตาม วัน เดือน ปี ในการสร้าง หรือก็คือการสร้างหน้าเว็บไซต์เปล่าๆ โดยไม่มีการเชื่อมโยงบทความหรือหน้า Page อื่น ซึ่งหากต้องการเชื่อมโยงไปหน้า Page อื่นจะต้องสร้างลิงก์ขึ้นมาเพื่อทำการเชื่อมโยง ส่วนใหญ่การสร้าง Page จะเป็นหน้าที่ไม่ค่อยมีการเปลี่ยนแปลงมากนัก อาทิเช่น หน้า About, Contact

2.2.4 บทความ (Post)

เป็น Page ชนิดหนึ่ง เป็นการเขียนบทความที่มีการจัดหมวดหมู่ (Categories) มีแท็ก (Tags) ในการจัดกลุ่มของบทความ ซึ่ง Post นี้จะมีการเรียงลำดับการสร้างและทำการเผยแพร่ ปัจจุบันล่าสุดตามลำดับ เช่น บทความมีการอัปเดตข่าวประชาสัมพันธ์ใหม่ๆ บทความใหม่จะถูกเรียงลำดับมาไว้บนสุดเสมอทำให้ง่ายต่อการตรวจสอบบทความที่มีการอัปเดตล่าสุด สามารถจัดเรียงตามวันที่เผยแพร่ ผู้เขียน หมวดหมู่ แท็ก วันที่ เดือน ปี นอกจากนั้น Post เนื้อหาที่มีความเชื่อมโยงหา กันได้ทำให้ง่ายในการจัดกลุ่มบทความและค้นหา หรือก็คือ Post เป็นหน้าที่สร้างขึ้นมาเพื่อทำการอัปเดตข่าวสารอยู่เสมอ

2.3 ไพธอน (Python)

เป็นภาษาเขียนโปรแกรมระดับสูงที่ใช้กันอย่างกว้างขวางในการเขียนโปรแกรมสำหรับวัตถุประสงค์ทั่วไป ภาษาไพธอนนั้นสร้างโดย Guido van Rossum และถูกเผยแพร่ครั้งแรกในปี 1991 ไพธอนนั้นเป็นภาษาแบบ Interpreter ที่ถูกออกแบบให้โค้ดอ่านได้ง่ายขึ้นและโครงสร้างของภาษานั้นจะทำให้โปรแกรมเมอร์สามารถเข้าใจแนวความคิดการเขียนโค้ดโดยใช้บรรทัดที่น้อยลงกว่าภาษาอย่าง C++ และ Java ซึ่งภาษานั้นถูกกำหนดให้มีโครงสร้างที่ตั้งใจให้การเขียนโค้ดเข้าใจง่ายทั้งในโปรแกรมเล็กไปจนถึงโปรแกรมขนาดใหญ่



ภาพที่ 2.2 สัญลักษณ์ของภาษาไพธอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 จุดเด่นของภาษาไพธอน

1. เป็นภาษาแบบอินเทอร์พรีเตอร์ (Interpreter)
2. เป็นภาษาที่เรียนรู้ได้ง่ายและเขียนได้อย่างรวดเร็ว
3. ทำงานได้อย่างรวดเร็ว
4. รองรับการเขียนแบบ OOP
5. มีไลบรารี (Library) รองรับมากมาย
6. สามารถใช้พัฒนาเว็บและโปรแกรมได้
7. มีความปลอดภัยสูง
8. หลากๆ บริษัทและองค์กรวิจัยวิทยาศาสตร์ทั่วโลกได้ใช้ภาษานี้ในการเขียนโปรแกรม

วิเคราะห์ข้อมูล เช่น Google, NASA เป็นต้น

9. Open Source สามารถนำไปใช้งานได้ทั้งทางการค้าและภาคการศึกษา
10. มีการพัฒนาอย่างต่อเนื่อง

2.4 HTML (Hyper Text Markup Language)

คือภาษาคอมพิวเตอร์ที่ใช้ในการแสดงผลของเอกสารบนเว็บไซต์หรือที่เรียกว่าเว็บเพจ ถูกพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) และจากการพัฒนาทางด้าน Software ของ Microsoft ทำให้ภาษา HTML เป็นอีกภาษาหนึ่งที่ใช้เขียนโปรแกรมได้หรือที่เรียกว่า HTML Application

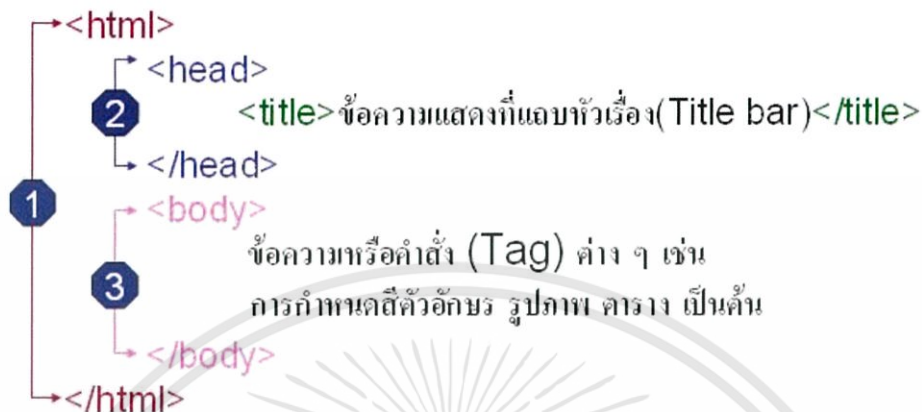
HTML เป็นภาษาประเภท Markup สำหรับการสร้างเว็บเพจโดยใช้ภาษา HTML สามารถทำโดยใช้โปรแกรม Text Editor ต่างๆ เช่น Notepad, Editplus หรือจะอาศัยโปรแกรมที่เป็นเครื่องมือช่วยสร้างเว็บเพจ เช่น Microsoft FrontPage, Dream Weaver ซึ่งอำนวยความสะดวกในการสร้างหน้า HTML ส่วนการเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะใช้โปรแกรม Web Browser เช่น Internet Explorer (IE), Mozilla Firefox, Safari, Opera, และ Netscape Navigator เป็นต้น

2.4.1 โครงสร้างของภาษา HTML

1. ส่วนที่กำหนดให้ Web Browser ทราบว่าเป็นเอกสารชนิด HTML ซึ่งจะมี Tag `<html>...</html>` กำกับอยู่ที่จุดเริ่มต้นและจุดสิ้นสุดของเอกสาร
2. ส่วนหัวเรื่อง (Head) เป็นส่วนที่กำหนดให้แสดงข้อความที่แถบหัวเรื่องของหน้าเว็บเพจนั้นๆ เช่น Tag `<title>...</title>` และเก็บ Tag พิเศษอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนเนื้อหา (Body) เป็นส่วนที่แสดงเนื้อหาของเว็บเพจทั้งหมดซึ่งประกอบด้วยข้อความและ Tag ต่างๆ เช่น Tag สำหรับจัดการกับรูปแบบของข้อความ ตาราง รูปภาพ กราฟิก ต่างๆ สีของตัวอักษร สีพื้น เป็นต้น



ภาพที่ 2.3 โครงสร้างของภาษา HTML

2.5 CSS (Cascading Style Sheets)

คือภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผลเอกสาร HTML โดยที่ CSS กำหนดกฎเกณฑ์ในการระบุรูปแบบของเนื้อหาในเอกสาร ได้แก่ สีของข้อความ สีพื้นหลัง ประเภทตัวอักษร และการจัดวางข้อความ ซึ่งการกำหนดรูปแบบ (Style) นี้ใช้หลักการของการแยกเนื้อหาเอกสาร HTML ออกจากคำสั่งที่ใช้ในการจัดรูปแบบการแสดงผลกำหนดให้รูปแบบของการแสดงผลเอกสารไม่ขึ้นอยู่กับเนื้อหาของเอกสารเพื่อให้ง่ายต่อการจัดรูปแบบการแสดงผลลัพท์ของเอกสาร HTML โดยเฉพาะในกรณีที่มีการเปลี่ยนแปลงเนื้อหาเอกสารบ่อยครั้งหรือต้องการควบคุมให้รูปแบบการแสดงผลเอกสาร HTML มีลักษณะของความสม่ำเสมอทั่วกันทุกหน้าเอกสารภายในเว็บไซต์เดียวกัน โดยกฎเกณฑ์ในการกำหนดรูปแบบของเอกสาร HTML ถูกเพิ่มเข้ามาครั้งแรกใน HTML 4.0 เมื่อปี พ.ศ. 2539 ในรูปแบบของ CSS level 1 Recommendations ที่กำหนดโดยองค์กร World Wide Web Consortium หรือ W3C

2.5.1 ข้อดีของ CSS

1. ช่วยลดการใช้ภาษา HTML ที่ใช้ในการจัดรูปแบบการแสดงผล
2. กำหนดรูปแบบการแสดงผลทั้งหน้าได้จากคำสั่งเพียงชุดเดียว
3. เมื่อต้องการแก้ไขสามารถแก้ไขได้ง่ายเพราะแก้ไขแค่ทีเดียว
4. ขนาดไฟล์มีขนาดเล็กส่งผลให้การประมวลผลทำได้เร็วมากขึ้น
5. สามารถควบคุมการแสดงผลบนหน้าจอให้ใกล้เคียงกันได้ในแต่ละเว็บเบราว์เซอร์
6. ช่วยให้ออกแบบเว็บเพจเพื่อแสดงผลกับโทรศัพท์มือถือได้ง่าย

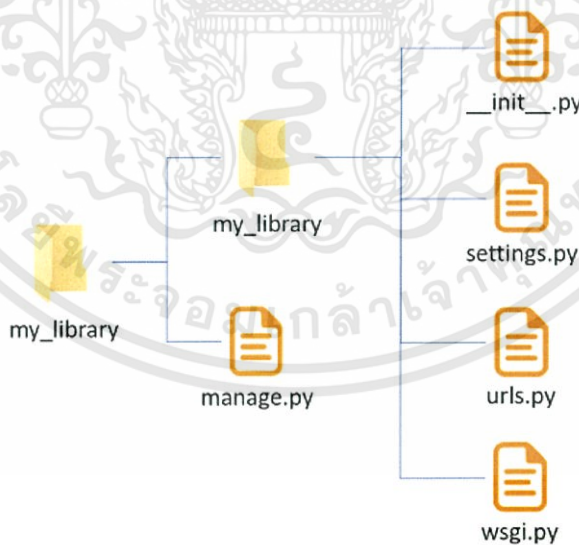
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Django Framework

เป็นชุดเครื่องมือ Framework สำหรับการพัฒนาเว็บไซต์ด้วยภาษาไพธอน โดยปัจจุบันภาษาไพธอนนั้นค่อนข้างได้รับความนิยมเพิ่มมากขึ้นเรื่อยๆ ทั้งในต่างประเทศและในเมืองไทย

2.6.1 คุณสมบัติของ Django Framework

1. Object-relational Mapper คือการกำหนด Data Model ในภาษา Python เพื่อการทำงานด้านข้อมูลและสนับสนุน Dynamic Database-access API
2. Automatic admin interface คือส่วนของการสร้าง Interface อัตโนมัติสำหรับการ add, edit, delete และ search ด้วย Django Framework
3. Elegant URL design คือการทำให้ URL มีความสวยงาม สั้น กระชับ และสื่อความหมายของหน้านั้นๆ ได้อย่างชัดเจนและเหมาะสมกับการทำ SEO ในปัจจุบัน
4. Template System คือ Django นั้นมีการออกแบบ Template Language เพื่อการเขียนแยกส่วนระหว่าง Design และ Business Logic
5. Cache System คือ ส่วนของการบันทึกหรือจัดการข้อมูลที่มีการดาวน์โหลดไปแล้ว เพื่อเพิ่มประสิทธิภาพการทำงานของเว็บไซต์ด้านความเร็วและด้านอื่นๆ
6. Internationalization คือ Django สนับสนุนแอปพลิเคชันที่มีความหลากหลายด้านภาษาในการแสดงผล



ภาพที่ 2.4 โครงสร้างโฟลเดอร์ของ Django Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 ข้อดีของ Django Framework

1. ใช้เวลาน้อยในการพัฒนาเว็บไซต์
2. เขียนโค้ดน้อยไม่ยุ่งยาก
3. ง่ายในการพัฒนา
4. โครงสร้างทำงานได้เร็ว
5. มีความปลอดภัยสูง
6. สามารถ Scale ได้ในกรณีที่เป็นเว็บไซต์ขนาดใหญ่
7. สามารถเปลี่ยนเป็นฐานข้อมูลแบบไหนก็ได้ เช่น mysql, sqlite, PostgreSQL, Oracle หรือแม้กระทั่ง nosql อย่าง MongoDB
8. มี Document ที่อ่านง่าย

2.6.3 ข้อเสียของ Django Framework

1. เนื่องจากเป็น WSGI App การ Deploy ลงเซิร์ฟเวอร์จึงไม่ตรงไปตรงมาเหมือน PHP โดย Django ต้องมีการติดตั้ง Package และ Setup Proxy Server ให้ชี้มายัง Django App
2. Host ในไทย ไม่ค่อยรองรับเพราะส่วนมากรองรับ PHP, .Net จึงต้องใช้ VPS หรือ เซิร์ฟเวอร์รูปแบบอื่น

2.7 Bootstrap

คือ ชุดโค้ดที่ช่วยในการพัฒนาส่วนหน้าของเว็บแอปพลิเคชัน (Front-end Framework) ที่นำเอา Javascript, HTML และ CSS มารวมเข้าด้วยกันเพื่อช่วยให้สามารถพัฒนาหน้าจอการแสดงผล (User Interface) ที่ปฏิสัมพันธ์กับผู้ใช้งานได้อย่างรวดเร็ว สวยงาม และใช้งานง่าย สามารถรองรับได้ทุก Smart Device ซึ่งการใช้ Bootstrap จะช่วยให้ผู้ใช้งานสามารถกรอกข้อมูลได้สะดวกมากขึ้นด้วยฟังก์ชันต่างๆ ที่ใช้ช่วยลดภาระการจัดวางทำให้ประหยัดเวลาในการพัฒนาเว็บไซต์และสามารถนำไปพัฒนาต่อได้ง่าย ในส่วนของไฟล์เตอร์ที่เก็บชุดคำสั่งหรือโค้ดต่างๆ ก็มีความเป็นระเบียบแบ่งส่วนชัดเจน 3 ไฟล์เตอร์ คือ js, css และ fonts

2.8 CSV (Comma-Separated Values)

เป็นรูปแบบการเก็บข้อมูลตัวเลขและตัวอักษรในรูปแบบอักขรธรรมดา ซึ่งรูปแบบอักขรธรรมดามีหมายถึงไฟล์ที่เก็บเฉพาะตัวอักษรเท่านั้น เป็นรูปแบบที่คนอ่านได้ด้วยโปรแกรมอ่านมาตรฐาน แต่ละบรรทัดในไฟล์เรียกว่า Record และแต่ละ Record จะประกอบไปด้วย Field ซึ่งคั่นด้วยเครื่องหมายลูกน้ำ (,) การใช้เครื่องหมายลูกน้ำ (Comma) นี้เป็นที่มาของชื่อรูปแบบไฟล์นั่นเอง

ไฟล์ CSV นี้มีการใช้งานอย่างแพร่หลายและมีรูปแบบที่ใกล้เคียงกับ Delimiter-separated formats รูปแบบที่มีการคั่นด้วยตัวคั่น เช่น การเก็บข้อมูลที่มี Tab คั่นจะเรียก Tab-separated values หรือการเก็บข้อมูลที่มี Space คั่นจะเรียก Space-separated values ซึ่งไฟล์ CSV มี Comma คั่นจึงเรียกรูปแบบไฟล์ CSV ว่า Comma-separated values ไฟล์ CSV จะมีนามสกุลไฟล์ *.CSV ไม่ว่าจะมิตัวคั่นเป็น Tab, Space หรือ Comma จะมีนามสกุลไฟล์ตัวเดียวกันคือ *.csv

2.8.1 โปรแกรมที่สนับสนุนไฟล์ CSV

ไฟล์ CSV เป็นรูปแบบไฟล์ที่ประกอบด้วยตัวเลขและตัวอักษรจึงมีโปรแกรมที่สนับสนุนในการอ่านและแก้ไขมากมาย เช่น Microsoft Excel, Notepad

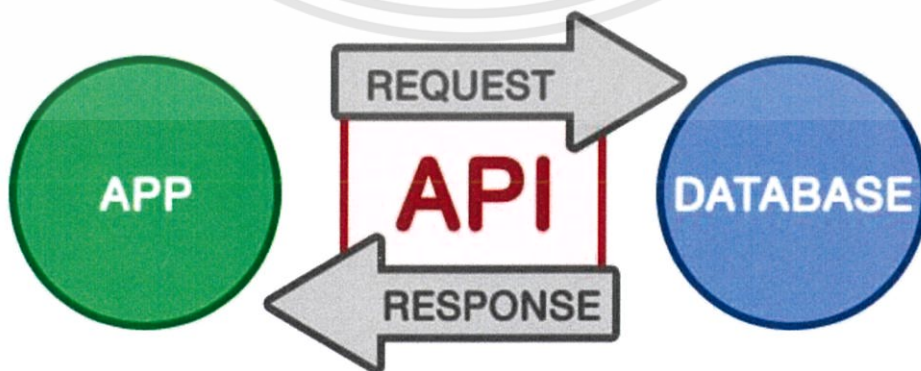
2.9 API (Application Programming Interface)

คือช่องทางการเชื่อมต่อระหว่างเว็บไซต์หนึ่งไปยังอีกเว็บไซต์หนึ่งหรือเป็นการเชื่อมต่อระหว่างผู้ใช้งานกับเซิร์ฟเวอร์หรือจากเซิร์ฟเวอร์เชื่อมต่อไปหาเซิร์ฟเวอร์ซึ่ง API นี้เปรียบได้เป็นภาษาคอมพิวเตอร์ที่ทำให้คอมพิวเตอร์สามารถสื่อสารและแลกเปลี่ยนข้อมูลกันได้อย่างอิสระ

2.9.1 กระบวนการทำงานของ API

API จะมีกระบวนการทำงานหลักๆ ในการรับส่งข้อมูลกัน โดยแบ่งกระบวนการทำงานได้ดังนี้

1. POST Method ใช้สำหรับการสร้างข้อมูลใหม่ในฐานข้อมูล
2. GET Method ใช้สำหรับร้องขอข้อมูลในรูปแบบต่างๆ
3. PUT Method ใช้สำหรับแทนที่ข้อมูลเดิม
4. PATCH Method ใช้สำหรับปรับปรุงแก้ไขข้อมูลเดิม
5. DELETE Method ใช้สำหรับการลบข้อมูล



ภาพที่ 2.5 หน้าที่ของ API

2.9.2 ประโยชน์ของ API

1. ช่วยในการพัฒนาเว็บไซต์หรือแอปพลิเคชันได้ง่ายและรวดเร็วซึ่ง API จะเป็นตัวช่วยที่นักพัฒนาไม่ต้องเข้าไปแก้ไขโค้ดคำสั่งเลยทำให้สะดวกสบายในการใช้งาน
2. ช่วยให้นักพัฒนาเว็บไซต์หรือเจ้าของเว็บไซต์สามารถฐานผู้ชมเว็บไซต์ให้มากขึ้น
3. ทำให้ผู้ใช้งานเว็บไซต์ต่างๆ ที่มีการติดตั้ง API ของอีกเว็บไซต์หนึ่ง ไม่ต้องเข้าหน้าเว็บไซต์ที่เป็นเจ้าของ API เพียงแต่เข้ามายังเว็บไซต์ที่มีการติดตั้ง API เท่านั้นทำให้การรับรู้ข่าวสารต่างๆ ทัวถึงกันและสะดวกในการใช้งานของผู้ใช้งานเว็บไซต์
4. API สามารถรับส่งข้อมูลข้ามเซิร์ฟเวอร์ได้

2.10 REST API

Representational State Transfer (REST) ถูกพูดถึงครั้งแรกในปี 2000 โดย Roy Thomas Fielding ซึ่ง Representational State Transfer (REST) นั้นเป็น Architecture (สถาปัตยกรรม การสื่อสารข้อมูล) รูปแบบหนึ่งที่ใช้แพร่กระจายสื่อ เป็นแนวทางใหม่ในการสร้าง Web Service แบบเรียบง่าย โดยเรียกใช้ผ่านทาง HTTP Method GET / POST / PUT / DELETE และส่งข้อมูลออกมา ในรูปของ XML ทำให้ปริมาณข้อมูลที่รับส่ง น้อยกว่าการใช้ Protocol SOAP อยู่มาก ซึ่งข้อดีข้อนี้ของ REST ทำให้ Developer หลายคนหันมาสนใจการเขียนโปรแกรมแบบใช้ RESTful Web Service กันมากขึ้น เพราะมีผลกับเรื่อง Performance ของการใช้งานโปรแกรมเป็นอย่างมาก แต่เนื่องจากเรื่อง REST นี้เพิ่งเกิดขึ้นมาเมื่อปี 2000 ทำให้ยังไม่มีมาตรฐานที่กำหนดให้บังคับใช้งานเหมือน Protocol SOAP เดิม ซึ่งจะสังเกตได้ว่าจะไม่มี REST Specification อยู่บน W3C และไม่มี REST Developer Toolkit ถ้าหากอยากใช้ RESTful ก็ต้องกำหนดเงื่อนไขจากภาษาที่เขียนเอาเอง โดยภาษาในยุคปัจจุบันนี้มีการรองรับ RESTful Web Service หรืออีกชื่อหนึ่งคือ RESTful Web API กันเป็นส่วนใหญ่แล้ว

2.10.1 พื้นฐานของ REST

การใช้งาน Resource ต่างๆ

1. ผ่าน HTTP protocol
2. ผ่าน URI ที่สื่อสารได้ชัดเจน
3. ผ่าน HTTP method ที่เหมาะสม เช่น
 - HTTP GET สำหรับการดึงข้อมูล
 - HTTP POST สำหรับการสร้างข้อมูล
 - HTTP PUT สำหรับการแก้ไขข้อมูล
 - HTTP DELETE สำหรับลบข้อมูล

รูปแบบของข้อมูลที่ใช้สื่อสารกันอยู่ในรูปแบบ XML, JSON, Atom และอื่นๆ อีกมากมาย แต่หนึ่งในรูปแบบที่ได้รับความนิยมคือ JSON

2.11 JSON

JavaScript Object Notation คือรูปแบบของข้อมูลที่ใช้สำหรับแลกเปลี่ยนข้อมูลที่มีขนาดเล็ก ซึ่งสามารถทำความเข้าใจได้ง่าย และสามารถถูกสร้างและอ่านได้ง่าย มันถูกกำหนดภายใต้ภาษา JavaScript (JavaScript Programming Language, Standard ECMA-262 3rd Edition – December 1999.)

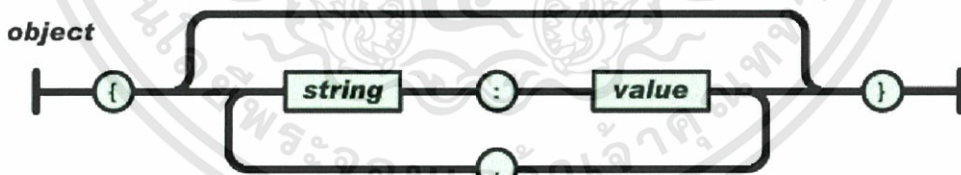
JSON เป็นรูปแบบข้อมูลตัวอักษรที่มีความเป็นอิสระอย่างสมบูรณ์ แต่จะมีหลักการการเขียนที่คุ้นเคยกับนักเขียนโปรแกรมภาษาต่างๆ ได้ไม่ว่าจะเป็น ภาษา C, C++, C#, Java, Javascript, Perl, Python และอื่นๆ คุณสมบัติเหล่านี้ทำให้ JSON เป็นภาษาแลกเปลี่ยนข้อมูลที่มีสมบูรณ์แบบ

ปัจจุบัน JSON นิยมใช้ในเว็บแอปพลิเคชัน โดยเฉพาะ AJAX โดย JSON เป็นฟอร์แมตทางเลือกในการส่งข้อมูลนอกเหนือไปจาก XML ซึ่งนิยมใช้กันอยู่แต่เดิม สาเหตุที่ JSON เริ่มได้รับความนิยมเป็นเพราะกระชับและเข้าใจง่ายกว่า XML

2.11.1 รูปแบบของ JSON

1. Object

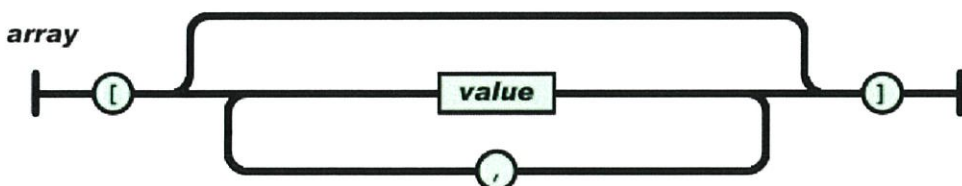
เป็นชุดของข้อมูลที่มีชื่อข้อมูลและค่าของข้อมูลนั้นคู่กัน ซึ่งจะถูกเริ่มต้นด้วยเครื่องหมาย { และจะปิดท้ายข้อมูลด้วยเครื่องหมาย } ข้อมูลแต่ละค่าจะมีเครื่องหมาย : กำกับระหว่างชื่อข้อมูลกับค่าของข้อมูลและแต่ละข้อมูลจะมีเครื่องหมาย , คั่น



ภาพที่ 2.6 JSON รูปแบบ Object

2. Array

เป็นลำดับของข้อมูลซึ่งจะถูกเริ่มต้นด้วยเครื่องหมาย [และจะจบด้วยเครื่องหมาย] แต่ละค่าของข้อมูลจะถูกคั่นด้วยเครื่องหมาย ,

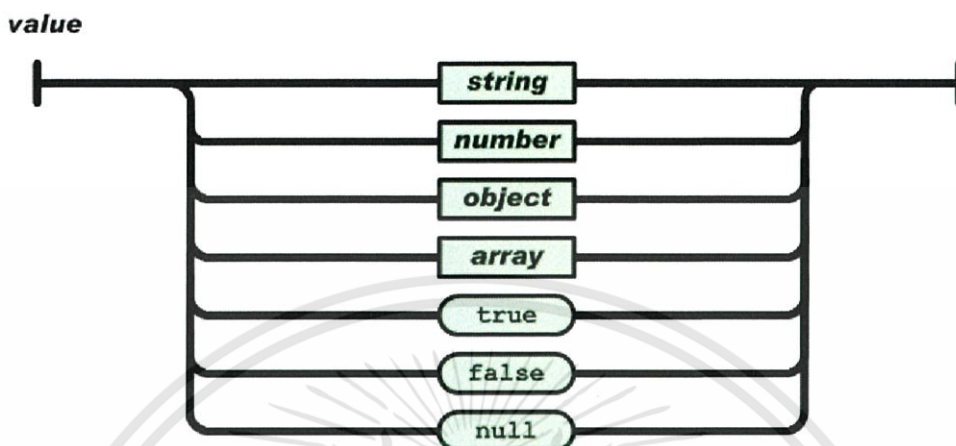


ภาพที่ 2.7 JSON รูปแบบ Array

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ¹³ วิชาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Value

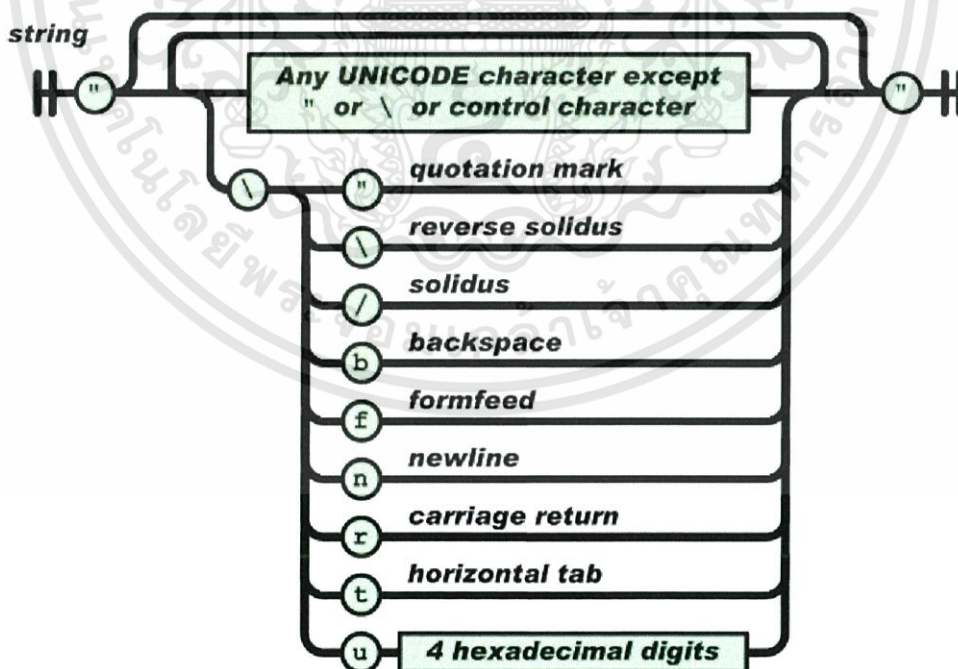
เป็น String ที่อยู่ในเครื่องหมาย “ ” หรือตัวเลขหรือค่าทางตรรกศาสตร์ True, False หรือค่า Null หรือ Object หรือ Array ซึ่งโครงสร้างสามารถวางซ้อนกันได้



ภาพที่ 2.8 JSON รูปแบบ Value

4. String

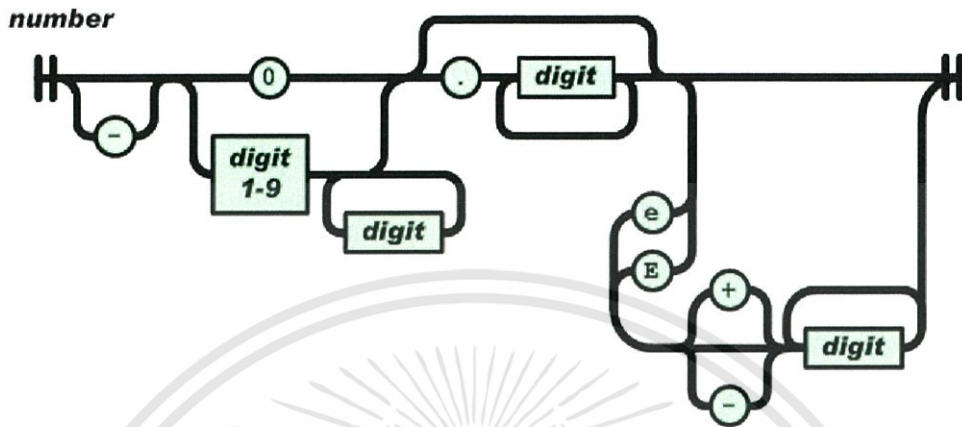
เป็นลำดับของตัวอักษรตั้งแต่ 0 ตัวอักษรหรือมากกว่านั้น ซึ่งอยู่ภายใต้เครื่องหมาย “ ” และจะใช้เครื่องหมาย \ ในการใส่เครื่องหมายกำกับต่างๆ ซึ่งจะมีลักษณะคล้ายกับ String ในภาษา C หรือภาษา Java



ภาพที่ 2.9 JSON รูปแบบ String

5. Number

มีความคล้ายคลึงกับ Number ในภาษา C หรือภาษา Java อย่างมากแต่จะไม่สามารถใช้เลขฐาน 8 กับเลขฐาน 16 ได้



ภาพที่ 2.10 JSON รูปแบบ Number

2.12 โฟลว์ชาร์ต (Flowchart)

คือ แผนภาพที่ใช้แสดงลำดับของขั้นตอนการทำงาน โดยใช้สัญลักษณ์หรือรูปภาพต่างๆ ในการเขียนผังงาน เพื่อใช้เป็นเครื่องมือในการลำดับความคิดในการเขียนโปรแกรม เป็นวิธีที่นิยมใช้เพราะมองเห็นภาพรวมและลำดับขั้นตอนที่ชัดเจน

2.12.1 ประเภทของโฟลว์ชาร์ต

1. ผังงานระบบ (System Flowchart) คือผังงานที่แสดงขั้นตอนการทำงานในระบบอย่างกว้างๆ แต่ไม่เจาะลงในระบบงานย่อย
2. ผังงานโปรแกรม (Program Flowchart) คือผังงานที่แสดงถึงขั้นตอนในการทำงานของโปรแกรม ตั้งแต่รับข้อมูล คำนวณ จนถึงแสดงผลลัพธ์







2.12.2 ประโยชน์ของโฟลว์ชาร์ต

1. สามารถศึกษาการทำงานของโปรแกรมได้ง่าย เพราะผังงานเป็นที่นิยมใช้ในการลำดับขั้นตอนการทำงานภาพรวมของระบบและไม่ขึ้นกับภาษาใดภาษาหนึ่ง
2. ใช้ผังงานตรวจสอบความผิดพลาดของลำดับขั้นตอนการทำงานได้รวดเร็ว
3. สามารถลำดับการทำงานได้ง่ายขึ้นและแยกแยะปัญหาได้ชัดเจน
4. การปรับปรุง เปลี่ยนแปลง แก้ไข ทำได้อย่างสะดวกและรวดเร็ว
5. สามารถนำผังงานไปใช้ได้กับการเขียนโปรแกรมทุกภาษาเพราะผังงานเป็นเครื่องมือที่ใช้ในการอธิบายหลักการทำงานเท่านั้น

6. การบำรุงรักษาโปรแกรมหรือการเปลี่ยนแปลงแก้ไขโปรแกรมให้มีประสิทธิภาพ ถ้าดูจากผังงานระบบจะช่วยให้สามารถทบทวนงานในโปรแกรมก่อนปรับปรุงได้ง่ายขึ้น

2.12.3 สัญลักษณ์ของโฟลว์ชาร์ต

ตารางที่ 2.1 สัญลักษณ์ในโฟลว์ชาร์ต

สัญลักษณ์	ชื่อ	ความหมาย
	Action or Process	การกำหนดค่าหรือการประมวลผลทั่วไป
	Decision	การตัดสินใจ การเปรียบเทียบ จะมีผลใน 2 ทิศทาง คือ กรณีผลตรวจสอบเงื่อนไขเป็นเท็จและเป็นจริง
	Input/Output	รับหรือแสดงข้อมูลโดยไม่ระบุชนิดอุปกรณ์
	Document Output	การแสดงผลเอกสารหรือการแสดงผลออกทางเครื่องพิมพ์
	Terminator	การเริ่มต้นหรือการสิ้นสุด
	Manual Input	การรับข้อมูลเข้าทางแป้นพิมพ์
	Connector	จุดเชื่อมต่อในหน้าเดียวกัน
	Display	จอภาพแสดงผล

2.13 ยูสเคสไดอะแกรม (Use Case Diagram)

คือ แผนภาพที่แสดงหน้าที่ที่ระบบจะต้องกระทำ (Functionality) ทั้งหมด โดยระบบดังกล่าว จะต้องเป็นระบบที่สนใจ หรือบางครั้งเรียกว่าเป็น ระบบย่อย (Subsystem) และแสดงปฏิสัมพันธ์ ระหว่างระบบงานและสิ่งที่อยู่นอกระบบงาน

ในการพัฒนาระบบงานใดๆ นั้น การเก็บรวบรวมความต้องการของผู้ใช้มีความสำคัญมาก และ จะทำในระยะแรกๆ ของการพัฒนาระบบงานเสมอ Use Case Diagram เป็น Diagram ที่ทำหน้าที่ Capture Requirement

2.13.1 ความสัมพันธ์ (Relationships)

ความสัมพันธ์ระหว่างยูสเคส หมายถึง ความสัมพันธ์ที่แต่ละยูสเคส ภายในระบบเองมี ความสัมพันธ์กัน โดยความสัมพันธ์ของยูสเคสนั้นสามารถแบ่งออกได้ 2 แบบ คือ Include และ Extend

1. ความสัมพันธ์แบบ Include คือความสัมพันธ์แบบเรียกใช้ จำเป็นต้องเกิดขึ้นทุกครั้ง หมายถึงการที่ยูสเคสหนึ่งเรียกใช้งานยูสเคสอีกอันหนึ่งคล้ายๆ กับการเรียกใช้งานโปรแกรมย่อยโดย โปรแกรมหลัก








2. ความสัมพันธ์แบบ Extend คือความสัมพันธ์แบบส่วนขยายหรือส่วนเพิ่ม จะเกิดขึ้น หรือไม่เกิดก็ได้แล้วแต่เหตุการณ์ หมายถึงการที่ยูสเคสหนึ่งไปมีผลต่อการทำงานตามปกติของอีกยูส เคสหนึ่ง ยูสเคสที่มา Extend นั้นจะมีผลทำให้การทำงานของยูสเคสที่ถูก Extend ถูกรบกวนหรือมี กิจกรรมที่เปลี่ยนแปลง

2.13.2 ประโยชน์ของยูสเคสไดอะแกรม

1. ทราบความสามารถของระบบ
2. ทราบผู้ใช้งานในแต่ละส่วนของระบบ
3. ง่ายต่อการสื่อสารระหว่างลูกค้าและผู้พัฒนาระบบ
4. ใช้ทดสอบระบบว่าตรงตามความต้องการของระบบหรือไม่
5. ช่วยให้ผู้พัฒนาระบบสามารถแยกแยะกิจกรรมที่อาจจะเกิดขึ้นในระบบ
6. เป็น Diagram พื้นฐาน ที่สามารถอธิบายสิ่งต่างๆ ได้โดยใช้รูปภาพที่ไม่ซับซ้อน

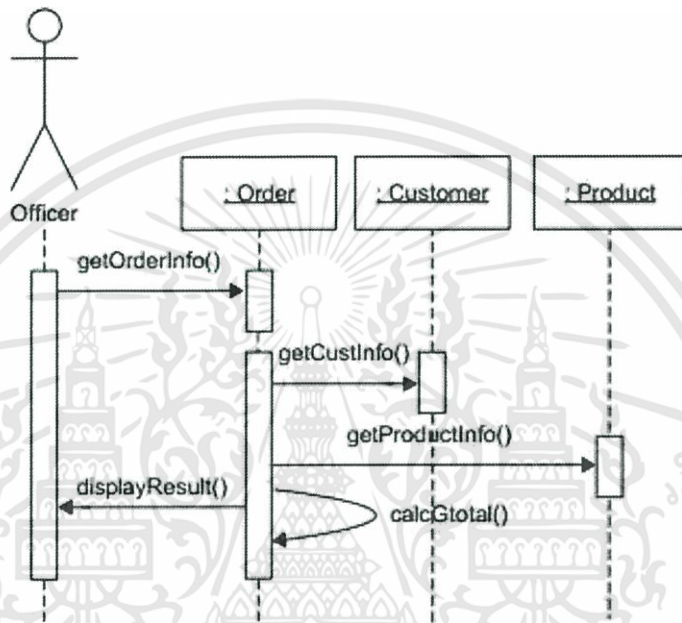
2.13.3 สัญลักษณ์ของยูสเคสไดอะแกรม

ตารางที่ 2.2 สัญลักษณ์ในยูสเคสไดอะแกรม

สัญลักษณ์	ชื่อ	ความหมาย
	Use Case	หน้าที่หรืองานต่างๆ ที่ระบบต้องทำ
	Actor	ผู้ที่มีส่วนเกี่ยวข้องกับระบบ
	System Boundary	ระบุขอบเขตของระบบที่กำลังพัฒนา
	Connection	เส้นเชื่อมแบบสื่อสารทางเดียว
	Connection	เส้นเชื่อมแบบสื่อสารสองทาง
	Include Relationship	ความสัมพันธ์แบบรวม
	Extend Relationship	ความสัมพันธ์แบบขยาย

2.14 ซีควেনซ์ไดอะแกรม (Sequence Diagram)

เป็นแผนภาพที่แสดงให้เห็นถึงการปฏิสัมพันธ์ (Interaction) ระหว่าง Object ของคลาส โดยเฉพาะการส่ง Message ระหว่าง Object ตามลำดับของเวลา (Sequence) ที่เกิดเหตุการณ์ขึ้น จากน้อยไปมาก โดยจะมีสัญลักษณ์แสดงให้เห็นลำดับของการส่ง Message ตามเวลาส่งอย่างชัดเจน แสดงลักษณะของซีควেনซ์ไดอะแกรม ดังภาพที่ 2.11









ภาพที่ 2.11 ลักษณะของซีควেনซ์ไดอะแกรม

จากภาพที่ 2.11 จะเห็นได้อย่างชัดเจนว่าซีควেনซ์ไดอะแกรมเป็นแผนภาพที่สามารถแสดงให้เห็นถึงการปฏิสัมพันธ์ระหว่าง Object ของคลาสตามลำดับของเวลาเป็นสำคัญมาให้ทราบว่า Message ไตเกิดก่อน Message ไตเกิดหลัง

2.14.1 สัญลักษณ์ของซีควেনซ์ไดอะแกรม

ตารางที่ 2.3 สัญลักษณ์ในซีควেনซ์ไดอะแกรม

สัญลักษณ์	ชื่อ	ความหมาย
	Actor	ผู้ที่มีส่วนเกี่ยวข้องกับระบบ

	Object	Object ที่ต้องทำหน้าที่ตอบสนองต่อ Actor
	Lifeline	เส้นแสดงชีวิตของ Object หรือ Class
	Focus of Control / Activation	จุดเริ่มต้นและจุดสิ้นสุดของแต่ละกิจกรรมในระหว่างที่มีชีวิตอยู่
	Message	คำสั่งหรือฟังก์ชันที่ Object หนึ่ง ส่งให้อีก Object หนึ่ง ซึ่งสามารถส่งกลับได้ด้วย
	Callback / Self Delegation	การประมวลผลและคืนค่าที่ได้ภายใน Object เดียวกัน

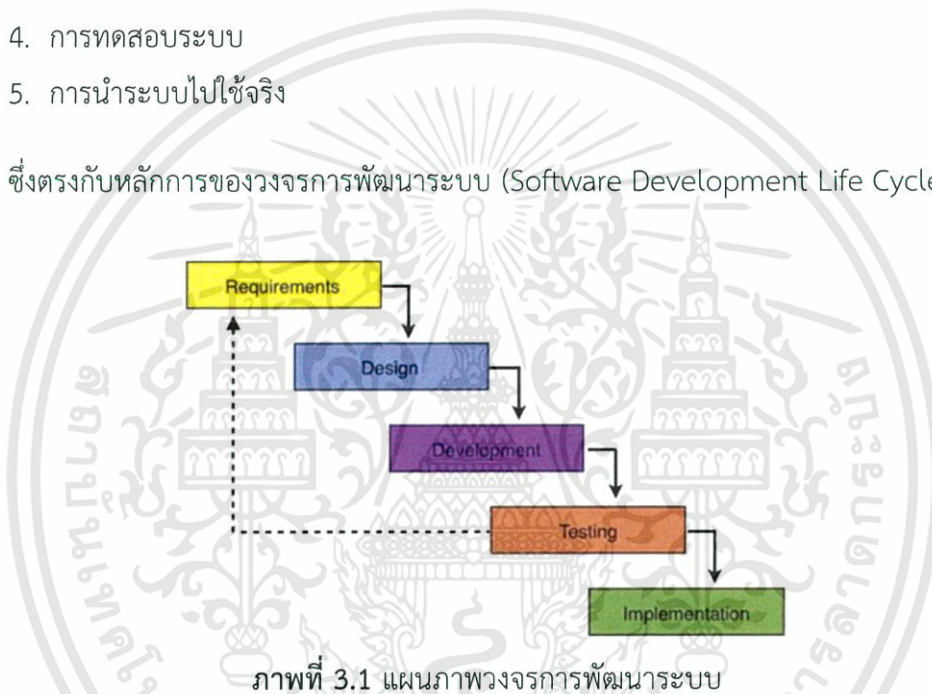
บทที่ 3

วิธีดำเนินการ

ในการทำเว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์ สามารถแบ่งการดำเนินการได้เป็น 5 ขั้นตอน ดังนี้

1. การวิเคราะห์ความต้องการจากผู้ใช้งาน
2. การออกแบบระบบ
3. การพัฒนาระบบ
4. การทดสอบระบบ
5. การนำระบบไปใช้จริง

ซึ่งตรงกับหลักการของวงจรการพัฒนาาระบบ (Software Development Life Cycle)



ภาพที่ 3.1 แผนภาพวงจรการพัฒนาาระบบ

3.1 การวิเคราะห์ความต้องการจากผู้ใช้งาน

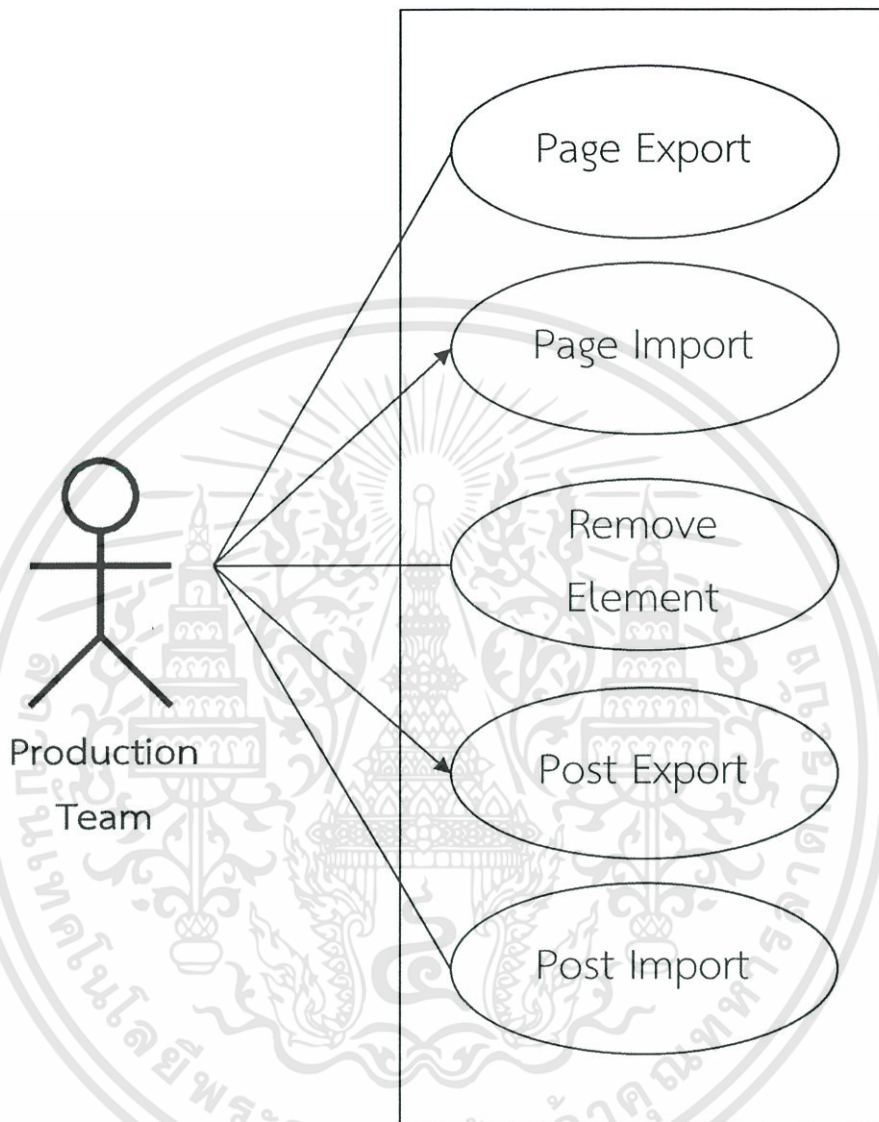
ก่อนเริ่มทำเว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์ ผู้จัดทำต้องไปพบปะกับผู้ที่ต้องการใช้งานเครื่องมือ โดยศึกษาว่าเดิมมีขั้นตอนการทำงานอย่างไร เพื่อนำมาพัฒนาระบบต่อไป โดยความต้องการของผู้ใช้งาน มีดังนี้

1. เป็นเว็บแอปพลิเคชันที่มีการใช้งานง่าย ไม่ยุ่งยาก
2. รวบรวมขั้นตอนการทำงานทุกอย่างไว้ในระบบ
3. สามารถคัดลอกเนื้อหาและสร้างได้หลายหน้าในการทำงานครั้งเดียว
4. แปลงพารูปภาพให้เป็นรีเลทีฟพาสเพื่อรองรับการเปลี่ยนโดเมนในอนาคต
5. สามารถตรวจสอบเนื้อหาที่คัดลอกมาจากเว็บไซต์เก่าก่อนจะนำไปสร้าง
6. ใช้ภาพแรกของบทความ (Post) นั้นๆ เป็น Feature image (ภาพหน้าปกของบทความ)
7. หน้าเว็บ (Page) ใช้ Slug เดิมจากเว็บไซต์เก่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบระบบ

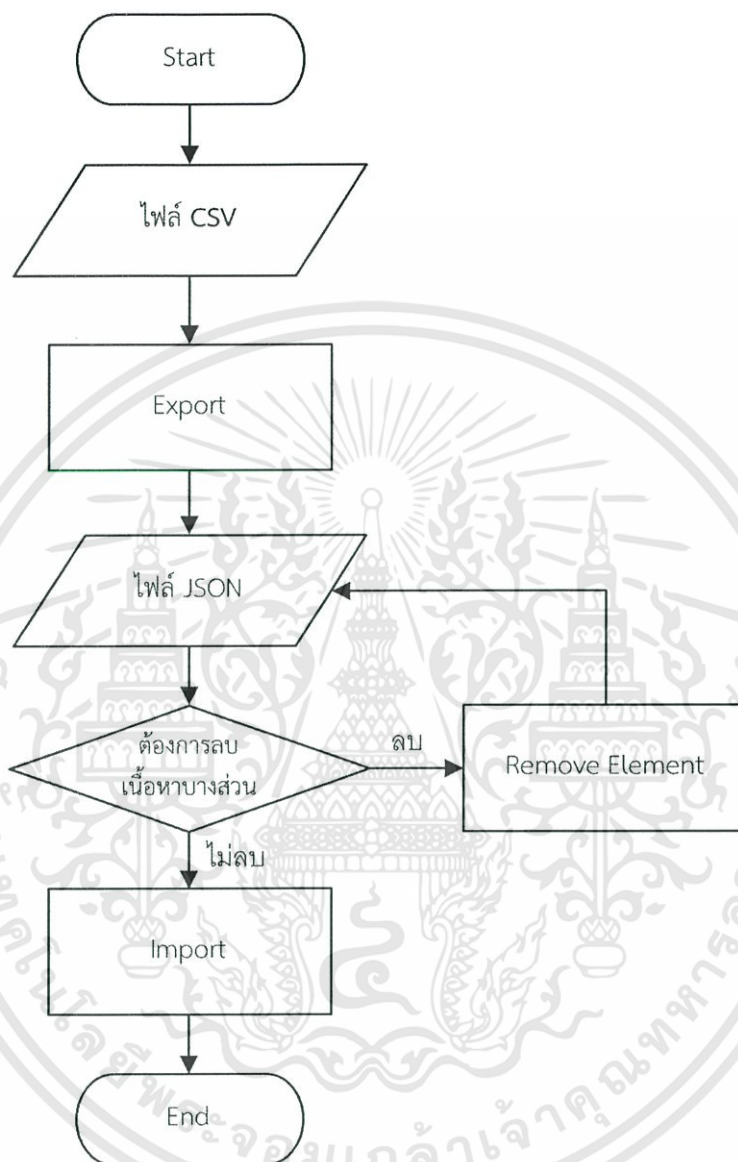
3.2.1 ยูสเคสไดอะแกรม (Use Case Diagram)



ภาพที่ 3.2 ยูสเคสไดอะแกรม

3.2.2 โฟลว์ชาร์ต (Flowchart)

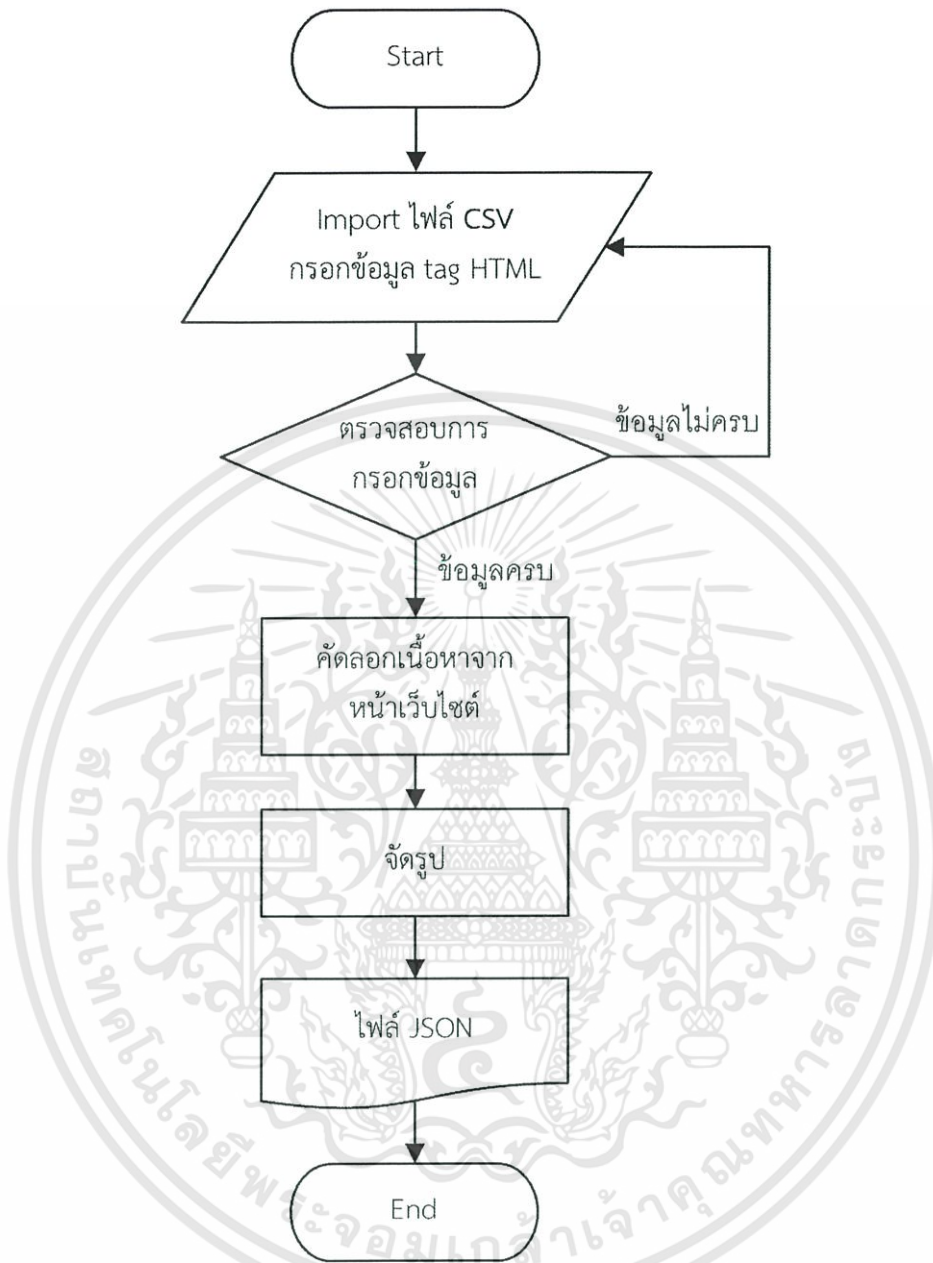
3.2.2.1 โฟลว์ชาร์ตการใช้งานเว็บแอปพลิเคชัน



ภาพที่ 3.3 โฟลว์ชาร์ตการใช้งานเว็บแอปพลิเคชัน

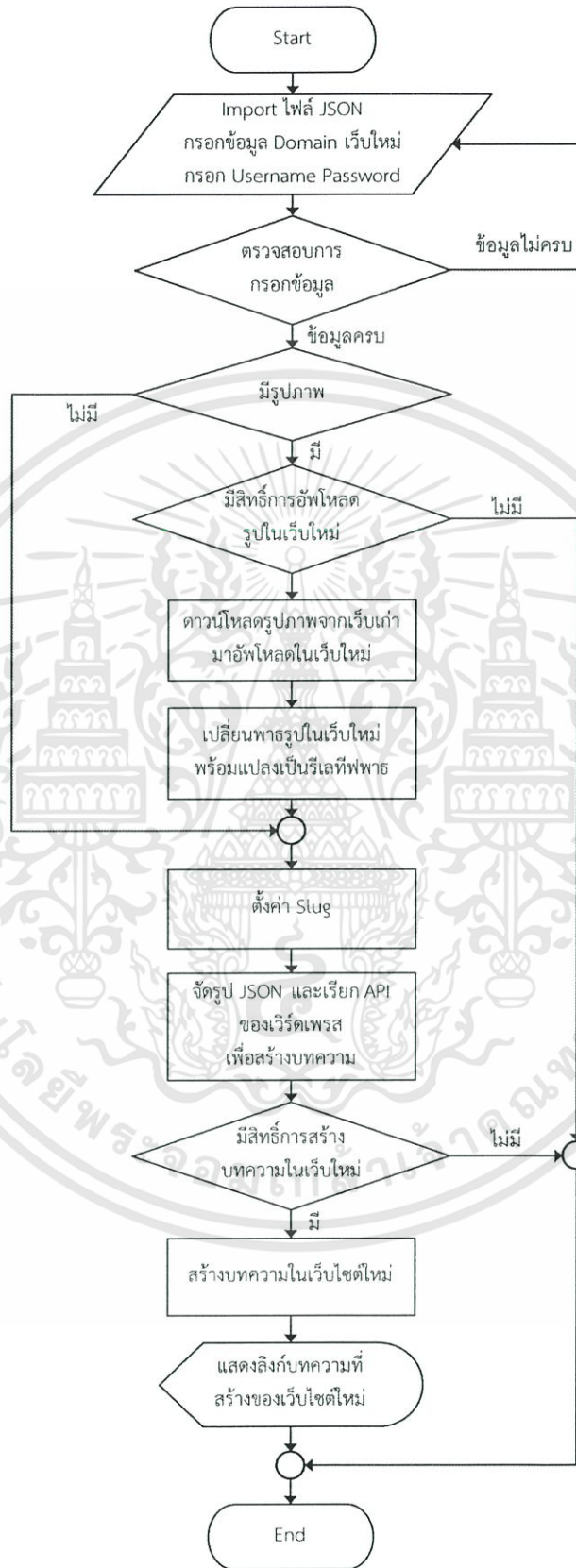
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.2 โฟลว์ชาร์ตฟังก์ชัน Export ของหน้า (Page)



ภาพที่ 3.4 โฟลว์ชาร์ตฟังก์ชัน Export หน้าเว็บ (Page)

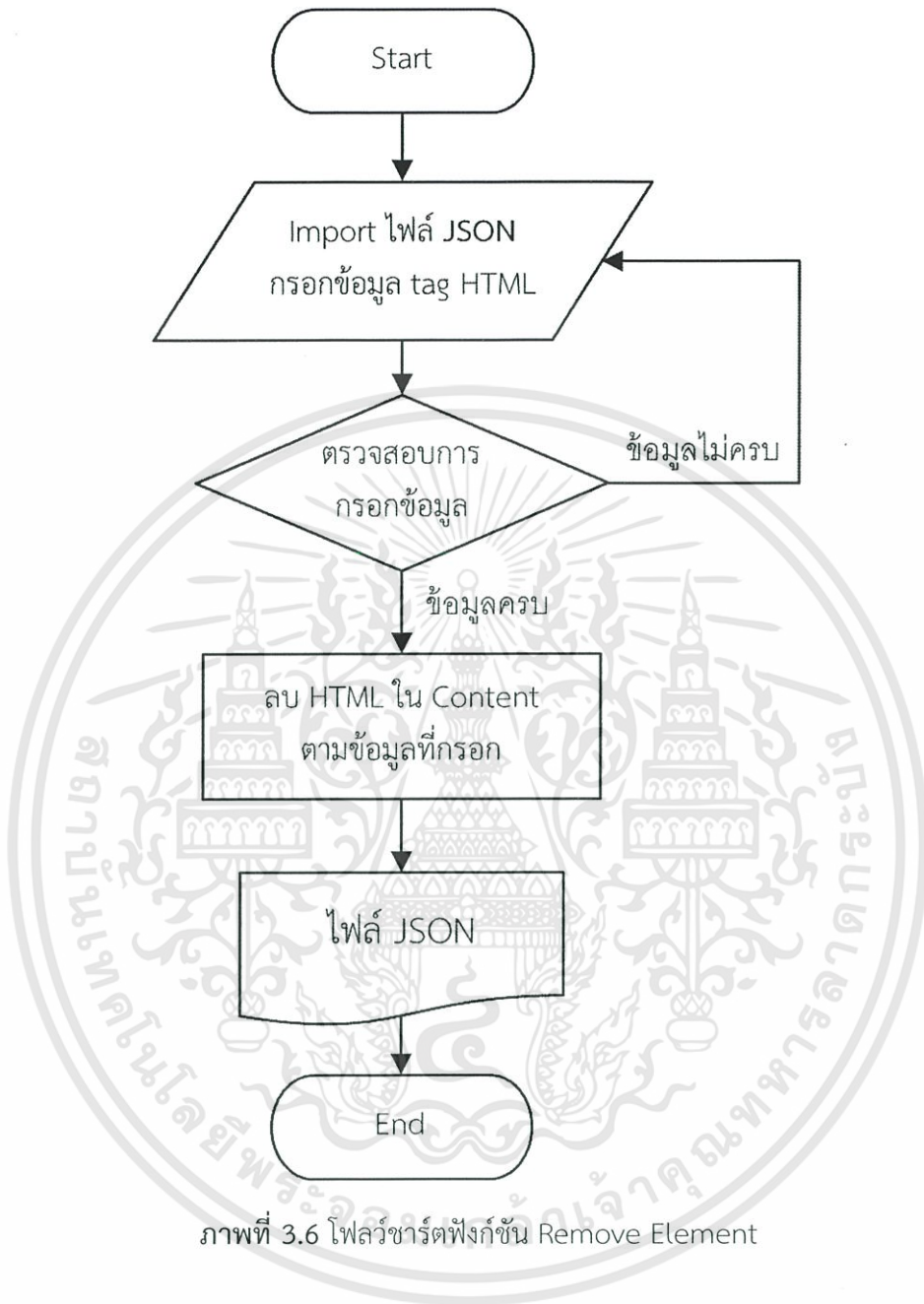
3.2.2.3 โฟลว์ชาร์ตฟังก์ชัน Import ของหน้าเว็บ (Page)



ภาพที่ 3.5 โฟลว์ชาร์ตฟังก์ชัน Import ของหน้าเว็บ (Page)

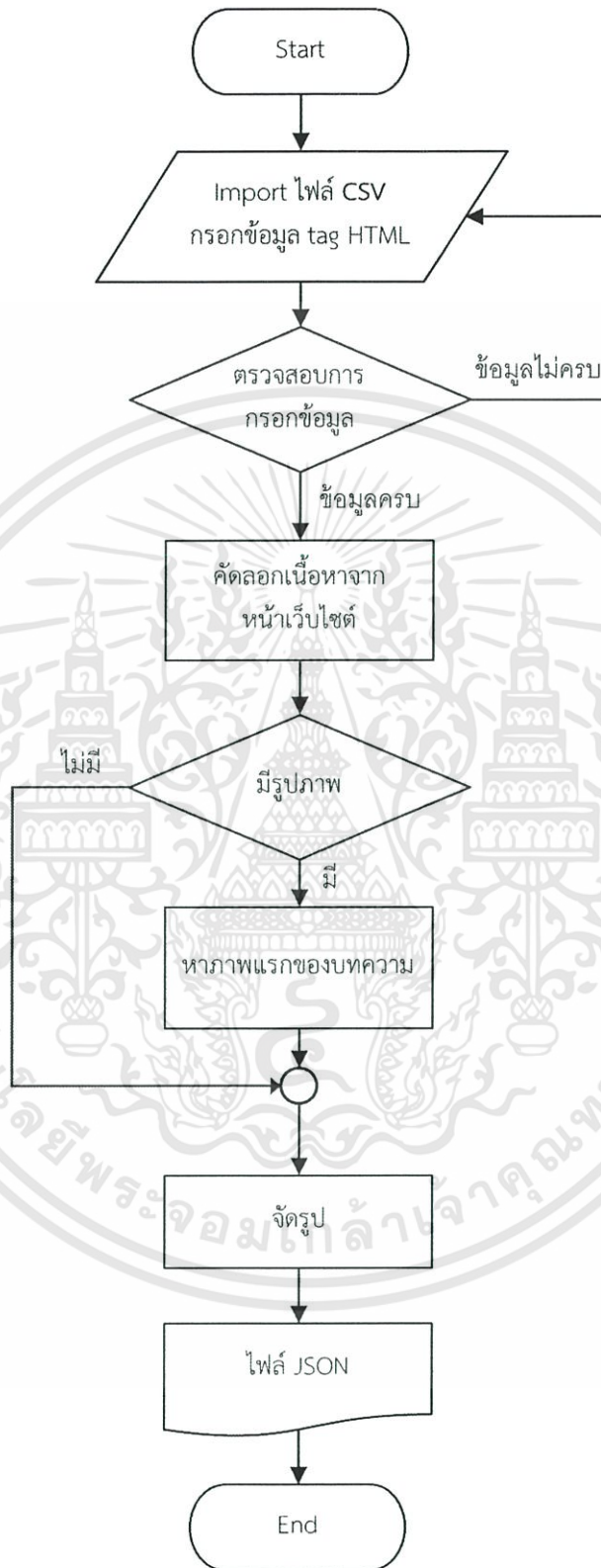
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.4 โพล์ชาร์ตฟังก์ชัน Remove Element



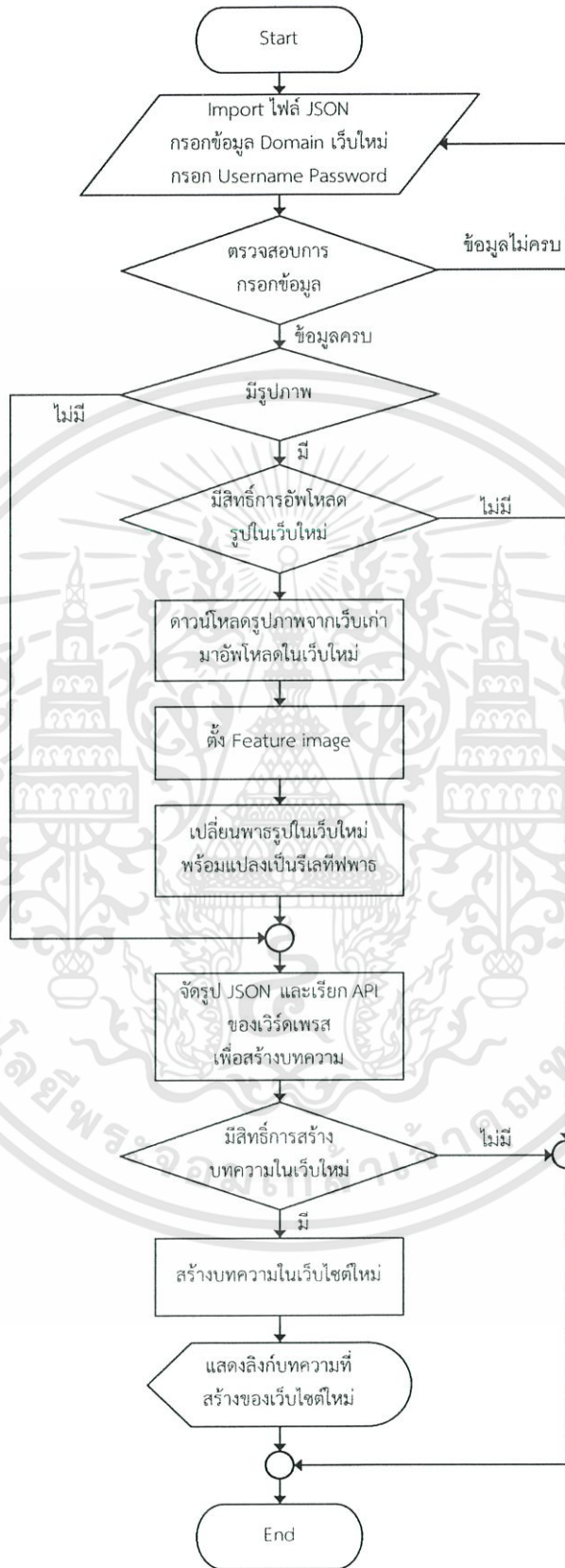
ภาพที่ 3.6 โพล์ชาร์ตฟังก์ชัน Remove Element

3.2.2.5 โฟลว์ชาร์ตฟังก์ชัน Export ของบทความ (Post)



ภาพที่ 3.7 โฟลว์ชาร์ตฟังก์ชัน Export บทความ (Post)

3.2.2.6 โฟลว์ชาร์ตฟังก์ชัน Import ของบทความ (Post)



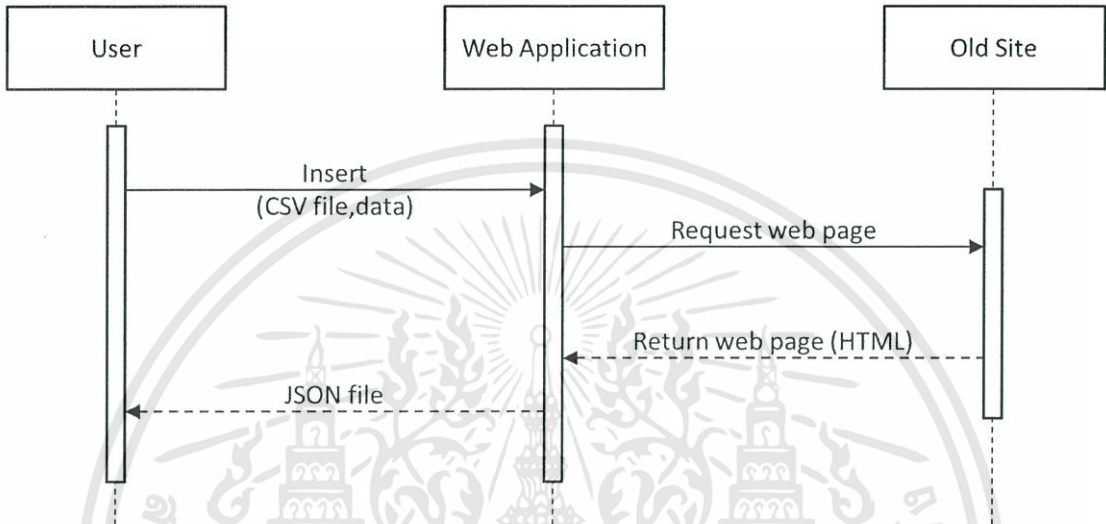
ภาพที่ 3.8 โฟลว์ชาร์ตฟังก์ชัน Import ของบทความ (Post)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ซีควนซ์ไดอะแกรม (Sequence Diagram)

3.2.3.1 ซีควนซ์ไดอะแกรมของฟังก์ชัน Export

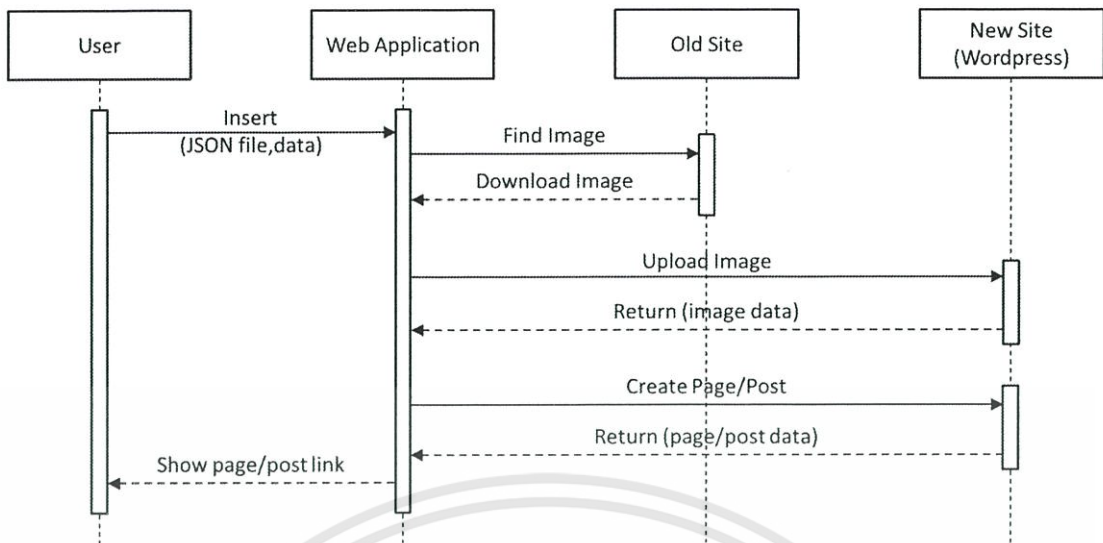
ผู้ใช้งานจะต้อง import ไฟล์ CSV ที่ลิสต์ลิงก์หน้าที่ต้องการคัดลอกและกรอกข้อมูล tag HTML ส่วนที่ต้องการคัดลอกเพื่อทางระบบจะไปดึงเนื้อหาจากหน้าเว็บไซต์เก่าโดยอิงจาก tag HTML ที่ผู้ใช้งานระบุ จากนั้นระบบจะนำเนื้อหามาจัดรูปแล้วส่งกลับให้ผู้ใช้งานเป็นไฟล์ JSON



ภาพที่ 3.9 ซีควนซ์ไดอะแกรมของฟังก์ชัน Export

3.2.3.2 ซีควนซ์ไดอะแกรมของฟังก์ชัน Import

ผู้ใช้งานจะต้อง Import ไฟล์ JSON ที่ได้จากการ Export และกรอกข้อมูลโดเมนเว็บไซต์ใหม่พร้อมกับ Username และ Password สำหรับตรวจสอบสิทธิ์ของผู้ใช้ในการสร้างหน้าและบทความของเว็บนั้นๆ โดยระบบจะรับข้อมูลจากไฟล์ JSON และข้อมูล que ผู้ใช้กรอกไปจัดการกับเนื้อหาตั้งแต่การดาวน์โหลดและอัปโหลดรูปภาพ เปลี่ยนพารามิเตอร์พร้อมทั้งแปลงเป็นรีเลทีฟพาทตั้ง Feature Image จัดรูปแบบให้เป็น JSON ที่พร้อมใช้งานจากนั้นเรียก API ของเวิร์ดเพรสเพื่อสร้างหน้าหรือบทความตามต้องการ



ภาพที่ 3.10 ซีควเอนซ์ไดอะแกรมของฟังก์ชัน Import

3.2.3.3 ซีควเอนซ์ไดอะแกรมของฟังก์ชัน Remove Element

ผู้ใช้งานจะต้อง Import ไฟล์ JSON ที่ได้จากการ Export และกรอกข้อมูล Tag HTML ส่วนที่ต้องการลบเพื่อทางระบบจะนำข้อมูลไปหาและลบส่วนที่ผู้ใช้งานต้องการลบใน JSON จากนั้นจะส่งกลับให้ผู้ใช้งานเป็นไฟล์ JSON เช่นเดิม



ภาพที่ 3.11 ซีควเอนซ์ไดอะแกรมของฟังก์ชัน Remove Element

3.3 การพัฒนาระบบ

3.3.1 ขั้นตอนการพัฒนาระบบ

1. ศึกษาเทคโนโลยีที่ต้องใช้ในการพัฒนา
2. พัฒนาเครื่องมือตามทีออกแบบไว้
3. ออกแบบการจัดวางฟังก์ชันต่างๆ (User Interface) โดยการใช้ Bootstrap ในการจัดตำแหน่ง
4. นำเครื่องมือที่ทำเสร็จสมบูรณ์ไปขึ้นบนอินเทอร์เน็ตของทางบริษัท

3.3.2 ตัวช่วยในการพัฒนาระบบ

1. ใช้ไลบรารี (Library) ชื่อว่า Beautifulsoup ของภาษาไพธอนสำหรับการดึงข้อมูลที่เป็นภาษา HTML จากหน้าเว็บและใช้ฟังก์ชันในการค้นหาส่วน Tag HTML ที่ผู้ใช้ต้องการโดยอ้างอิงจากชื่อคลาสและไอดีเพื่อทำการดึงข้อมูลจาก Tag นั้นๆ
2. ใช้ปลั๊กอิน WordPress REST API ของเวิร์ดเพรสในการส่งข้อมูล JSON สำหรับการสร้างหน้าหรือบทความเว็บไซต์ที่สร้างด้วยเวิร์ดเพรส
3. ใช้ปลั๊กอิน JSON Basic Authentication คู่กับ WordPress REST API สำหรับยืนยันตัวตนในการสร้างหน้าและบทความของเว็บไซต์นั้นๆ โดยการส่ง Username และ Password ของผู้ใช้ใน Header พร้อมการส่งข้อมูล JSON เพื่อสร้างหน้าและบทความ
4. ใช้ Bootstrap สำหรับการจัดวางตำแหน่งและจัดการรูปแบบหน้าเมนูต่างๆ

3.3.3 การพัฒนาฟังก์ชันต่างๆ

จากการออกแบบเว็บแอปพลิเคชันประกอบด้วย 5 ฟังก์ชัน ดังนี้

1. Page Export
2. Page Import
3. Remove Element
4. Post Export
5. Post Import

3.3.3.1 Page Export

การทำงานเริ่มจากรับข้อมูล Tag HTML และไฟล์ CSV จากผู้ใช้ เพื่อค้นหา Tag HTML ที่มี Element ตามที่ผู้ใช้กำหนดในเว็บไซต์ที่ถูกलिस्टมาในไฟล์ CSV คัดลอกตามส่วนที่กำหนด และนำมาจัดรูปให้อยู่ในรูปแบบ JSON จากนั้น Export ไฟล์ JSON ออกมาให้ผู้ใช้ใช้งานต่อไป

โดย JSON ที่สร้างขึ้นกำหนดให้มีคีย์ (Key) ดังนี้

1. Title สำหรับส่วนหัวเรื่องของหน้าเว็บ
2. Content สำหรับส่วนเนื้อหาของหน้าเว็บ
3. Page source สำหรับเก็บลิงก์หน้าเว็บเก่าเพื่อใช้งานในฟังก์ชันต่อไป

```
def post(self, request):
    csv_file = request.FILES['upload_file']

    title_tag = request.POST['title_tag']
    title_class_type = request.POST['title_class_type']
    title_class = request.POST['title_class']
    content_tag = request.POST['content_tag']
    content_class_type = request.POST['content_class_type']
    content_class = request.POST['content_class']

    page_json = []
    reader = csv.reader(csv_file, delimiter=',')
    for row in reader:
        website_url = row[0]

        html_page = urllib2.urlopen(website_url)
        soup = BeautifulSoup(html_page, "html.parser" )

        try:
            for titles in soup.find(title_tag, {title_class_type:title_class}):
                try:
                    title = titles.text
                except:
                    title = str(titles)
            except:
                title = ''

            content = ''
            for contents in soup.find(content_tag, {content_class_type:content_class}):
                content += str(contents)

            data_json = {
                'title': title,
                'content': content,
                'page_source' : website_url
            }

            page_json.append(data_json)

    data = json.dumps(page_json)
    response = HttpResponse(content=data)
    response['Content-Type'] = 'application/json'
    response['Content-Disposition'] = 'attachment; filename="page_content.json"'
    return response
```

ภาพที่ 3.12 โค้ดการทำงานของฟังก์ชัน Page Export

3.3.3.2 Page Import

การทำงานเริ่มจากรับข้อมูลโดเมน Username, Password และไฟล์ JSON ที่ผู้ใช้ได้จากฟังก์ชัน Page Export

```
def post(self, request):
    json_file = request.FILES['upload_file']
    host_url = request.POST['host_url']
    user_id = request.POST['id_field']
    user_password = request.POST['password_field']

    host_url = host_url.strip('/')

    auth = base64.b64encode(user_id + ':' + user_password)
    headers = {'Authorization': 'Basic ' + auth}

    json_file = json_file.read()
    json_data = json.loads(json_file)

    pages = []
    for data in json_data:
        image_list = self.gather_image(json_data)
        img_dict = self.upload_image(headers, host_url, image_list)
        for key,value in img_dict.iteritems():
            value = value.replace(host_url, '')
            json_file = json_file.replace(key,value)

        page_url = self.create_page(headers, host_url, data)
        pages.append(
            {
                'url': page_url,
                'title': data['title']
            }
        )
    return render(
        request, 'result.html', {'datas': pages, 'type': 'Page'})
```

ภาพที่ 3.13 โค้ดการทำงานของฟังก์ชัน Page Import

นำ Content จาก JSON มาจัดการหารูปภาพภายใน Content โดยหากมีรูปภาพจะทำการจัดรูปพาให้พร้อมสำหรับการดาวน์โหลด

```

def gather_image(self, json_data):
    image_list_outer = []

    for data in json_data:
        soup = BeautifulSoup(data['content'], 'html.parser')
        page_source = data['page_source']

        for img_srcs in soup.findAll('img'):
            img_src = img_srcs.get('src')

            image_list_inner = []

            if "http" in img_src:
                old_image = img_src
                image_list_inner.append(old_image)

                new_image = img_src
                image_list_inner.append(new_image)

            elif ".." in img_src:
                old_image = img_src
                image_list_inner.append(old_image)

                new_image = page_source + old_image.split('..')[1]
                image_list_inner.append(new_image)

            else:
                old_image = img_src
                image_list_inner.append(old_image)

                new_image = page_source + '/' + old_image
                image_list_inner.append(new_image)

            image_list_outer.append(image_list_inner)

    return image_list_outer

```

ภาพที่ 3.14 โค้ดฟังก์ชันการจัดรูปลิงก์สำหรับดาวน์โหลดรูปภาพ

จากภาพที่ 3.14 ฟังก์ชันจะมีการคืนค่าพารามิเตอร์อยู่ในรูป Array ที่มีทั้งพารามิเตอร์ใหม่สำหรับการดาวน์โหลดและพารามิเตอร์เก่าสำหรับการ Replace พารามิเตอร์ในภายหลัง

นำลิงก์ดาวน์โหลดรูปภาพมาจัดการดาวน์โหลดและอัปเดตด้วยการเรียก API ของเว็บไซต์ โดยใส่ Header จากภาพที่ 3.13 เพื่อยืนยันตัวตนว่ามีสิทธิ์ในการอัปเดตรูปภาพในเว็บไซต์นี้

```

def upload_image(self, headers, host_url, images):
    img_dict = {}
    url = host_url + '/wp-json/wp/v2/media'
    for img_url in images:
        key = img_url[0]
        download_url = img_url[1]
        if not img_dict.has_key(key):
            img_name = download_url.split('/')[-1]
            img_path = 'media/' + img_name
            urllib.urlretrieve(download_url, img_path)
            with open(img_path) as img:
                files = {'file': img}
                img_data = {
                    'title':img_name,
                    'status':'publish'
                }
            r = requests.post(url, data=img_data, headers=headers, files=files)
            img_response = json.loads(r.content)
            img_dict[key] = img_response['guid']['rendered']
    return img_dict

```

ภาพที่ 3.15 โค้ดฟังก์ชันการดาวน์โหลดและอัปโหลดรูปภาพสำหรับ Page

จากภาพที่ 3.15 ฟังก์ชันจะมีการคืนค่าในรูป Dictionary ซึ่งมีคีย์ (Key) เป็นพารามิเตอร์จากเว็บเดิมและมีค่า (Value) เป็นพารามิเตอร์ใหม่ที่อัปโหลดไปในเว็บใหม่สำหรับการ Replace พารามิเตอร์ใน Content ให้เป็นพารามิเตอร์ที่ต้องการ

โดยก่อนทำการ Replace มีการเปลี่ยนค่าของพารามิเตอร์ในเว็บไซต์ใหม่ให้เป็น Relative Path ตามความต้องการของผู้ใช้

เมื่อจัดการกับรูปภาพและ Content แล้วนำข้อมูลทั้งหมดมาจัดรูปให้อยู่ในรูปแบบ JSON ที่พร้อมสำหรับสร้างหน้าเว็บ โดยกำหนดคีย์ (Key) และค่า (Value) สำหรับการสร้างหน้าเว็บดังนี้

1. Status ระบุสถานะของหน้าเว็บที่สร้าง
2. Title ระบุส่วนหัวเรื่องของหน้าเว็บ
3. Content ระบุส่วนเนื้อหาของหน้าเว็บ
4. Slug ระบุหน้าที่แสดงใน URL ของหน้าเว็บ

เมื่อได้ JSON ตามต้องการแล้ว ทำการสร้างหน้าเว็บด้วยการเรียก API ของเวิร์ดเพรสโดยใส่ Header จากภาพที่ 3.13 เพื่อยืนยันตัวตนว่ามีสิทธิ์ในการสร้างหน้าเว็บในเว็บไซต์นี้

```

def create_page(self, headers, host_url, data):
    host_url = host_url.strip('/')

    url = host_url + '/wp-json/wp/v2/pages'
    last_url_source = data['page_source']

    last_url = last_url_source.strip('/')
    last_url = last_url.split('/')[-1]

    post_data = {
        'status': 'publish',
        'title': data['title'],
        'content': data['content'],
        'slug': last_url
    }

    r = requests.post(url, headers=headers, data=post_data)
    response = json.loads(r.content)
    page_url = response['guid']['rendered']
    return page_url

```

ภาพที่ 3.16 โค้ดฟังก์ชันการสร้าง Page

จากภาพที่ 3.16 ฟังก์ชันจะมีการคืนค่าลิงก์ของหน้าเว็บที่สร้างในเว็บไซต์ใหม่เพื่อนำไปแสดงแก่ผู้ใช้เมื่อทำการสร้างหน้าเว็บเสร็จ ตามโค้ดจากภาพ 3.13

3.3.3.3 Remove Element

การทำงานเริ่มจากรับข้อมูล Tag HTML และไฟล์ JSON ที่ผู้ใช้ได้จากฟังก์ชัน Page Export เพื่อค้นหา Tag HTML ที่มี Element ตามที่ผู้ใช้กำหนดใน Content จากไฟล์ JSON จากนั้น Export ไฟล์ JSON ออกมาให้ผู้ใช้ใช้งานต่อไป

```

def post(self, request):
    json_file = request.FILES['upload_file']
    tag_data = str(request.POST['decompose_tag'])
    class_data = str(request.POST['decompose_class'])
    decompose_type = str(request.POST['decompose_type'])

    json_data = json.loads(json_file.read())
    page_json = []

    decompose_order_list = []
    for data in json_data:
        soup = BeautifulSoup(data['content'], 'html.parser')

        try:
            soup.find(tag_data, {decompose_type: class_data}).decompose()
        except:
            pass

        data['content'] = str(soup)
        page_json.append(data)

    data = json.dumps(page_json)
    response = HttpResponse(content=data)
    response['Content-Type'] = 'application/json'
    response['Content-Disposition'] = 'attachment; filename="page_content.json"'
    return response

```

ภาพที่ 3.17 โค้ดการทำงานของฟังก์ชัน Remove Element

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.4 Post Export

การทำงานเริ่มจากรับข้อมูล Tag HTML และไฟล์ CSV จากผู้ใช้ เพื่อค้นหา Tag HTML ที่มี Element ตามที่ผู้ใช้กำหนดในเว็บไซต์ที่ถูกलिस्टมาในไฟล์ CSV คัดลอกตามส่วนที่กำหนด และนำมาจัดรูปให้อยู่ในรูปแบบ JSON จากนั้น Export ไฟล์ JSON ออกมาให้ผู้ใช้ใช้งานต่อไป

โดย JSON ที่สร้างขึ้นกำหนดให้มีคีย์ (Key) ดังนี้

1. Title สำหรับส่วนหัวเรื่องของบทความ
2. Date สำหรับส่วนวันที่เผยแพร่บทความ
3. Content สำหรับส่วนเนื้อหาของบทความ
4. Image สำหรับเก็บพารามิเตอร์ของรูปภาพแรกของบทความ
5. Page source สำหรับเก็บลิงก์หน้าเว็บเก่าเพื่อใช้งานในฟังก์ชันต่อไป

โดย Tag ที่ผู้ใช้ระบุมาสำหรับ Date (วันที่เผยแพร่) อาจมีข้อความอื่นที่ไม่เกี่ยวกับตัวเลขหรือวันที่ที่ต้องการ จึงมีฟังก์ชันสำหรับการลบข้อความส่วนนั้นออกเพื่อให้เก็บค่าวันที่ในรูปแบบที่ถูกต้อง

```
def clean_date_text(self, date_data, date_remove_text):  
    date = date_data.replace(date_remove_text, '')  
    date = date.strip('\t\r\n')  
    return date
```

ภาพที่ 3.18 โค้ดฟังก์ชันการจัดการค่าวันที่เผยแพร่ของบทความ

```

def post(self, request):
    csv_file = request.FILES['upload_file']

    title_tag = request.POST['title_tag']
    title_class_type = request.POST['title_class_type']
    title_class = request.POST['title_class']
    date_tag = request.POST['date_tag']
    date_class_type = request.POST['date_class_type']
    date_class = request.POST['date_class']
    date_remove_text = request.POST['date_remove_text']
    content_tag = request.POST['content_tag']
    content_class_type = request.POST['content_class_type']
    content_class = request.POST['content_class']

    post_json = []
    reader = csv.reader(csv_file, delimiter=',')
    for row in reader:
        website_url = row[0]

        html_page = urllib2.urlopen(website_url)
        soup = BeautifulSoup(html_page, "html.parser")

        title = ''
        for titles in soup.find(title_tag, {title_class_type:title_class}):
            try:
                title = titles.text
            except:
                title = str(titles)

        date = ''
        for dates in soup.find(date_tag, {date_class_type:date_class}):
            date_data = str(dates)
            date = self.clean_date_text(date_data, date_remove_text)

        content = ''
        for contents in soup.find(content_tag, {content_class_type:content_class}):
            content += str(contents)

        image = ''
        try:
            soup = BeautifulSoup(content, "html.parser")
            image = soup.find('img').get('src')
        except:
            pass

        data_json = {
            'title': title,
            'date': date,
            'content': content,
            'image': image,
            'page_source': website_url,
        }

        post_json.append(data_json)

    data = json.dumps(post_json)
    response = HttpResponse(content=data)
    response['Content-Type'] = 'application/json'
    response['Content-Disposition'] = 'attachment; filename="post_content.json"'
    return response

```

ภาพที่ 3.19 โค้ดการทำงานของฟังก์ชัน Post Export

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.5 Post Import

การทำงานเริ่มจากรับข้อมูลโดเมน Username, Password และไฟล์ JSON ที่ผู้ใช้ได้จากฟังก์ชัน Post Export

```
def post(self, request):
    json_file = request.FILES['upload_file']

    host_url = request.POST['host_url']
    user_id = request.POST['id_field']
    user_password = request.POST['password_field']

    host_url = host_url.strip('/')

    auth = base64.b64encode(user_id + ':' + user_password)
    headers = {'Authorization': 'Basic ' + auth}

    json_file = json_file.read()
    json_data = json.loads(json_file)

    posts = []
    for data in json_data:
        image_list = self.gather_image(data)
        img_dict, img_id = self.Upload_image(headers, host_url, image_list, data['image'])

        if data['image']:
            img_id = img_id
        else:
            img_id = 0

        for key,value in img_dict.iteritems():
            value = value.replace(host_url, '')
            data['content'] = data['content'].replace(key, value)

        post_url = self.create_post(headers, host_url, data, img_id)

        posts.append(
            {
                'url': post_url,
                'title': data['title']
            }
        )
    return render(
        request, 'result.html', {'datas': posts, 'type': 'Post'})
```

ภาพที่ 3.20 โค้ดการทำงานของฟังก์ชัน Post Import

โดยการทำงานของ Post Import มีลักษณะคล้าย Page Import ต่างกันที่ Post Import จะมีส่วนที่จัดการในเรื่องของ Feature Image เพิ่มขึ้นมา

```

def upload_image(self, headers, host_url, images, feature_img):
    img_dict = {}
    img_id = ''
    url = host_url + '/wp-json/wp/v2/media'
    for img_url in images:
        key = img_url[0]
        download_url = self.clean_img_url(img_url[1])
        if not img_dict.has_key(key):
            img_name = download_url.split('/')[-1]

            img_path = 'media/' + img_name
            urllib.urlretrieve(download_url, img_path)
            with open(img_path, 'rb') as img:
                files = {'file': img}

            img_data = {
                'title': img_name,
                'status': 'publish'
            }

            r = requests.post(url, data=img_data, headers=headers, files=files)
            img_response = json.loads(r.content)
            img_dict[key] = img_response['media_details']['sizes']['large']['source_url']

            if feature_img == key:
                img_id = img_response['id']

    return img_dict, img_id

```

ภาพที่ 3.21 โค้ดฟังก์ชันการดาวน์โหลดและอัปโหลดรูปภาพสำหรับ Post

จากภาพที่ 3.21 ในฟังก์ชันจะมีส่วนของการเช็ค Feature Image ของแต่ละ Post และคืนค่า ID ของรูปภาพ Feature Image ในเว็บไซต์ใหม่ เพิ่มมาจากฟังก์ชันในภาพที่ 3.15

เมื่อจัดการกับรูปภาพและ Content แล้ว นำข้อมูลทั้งหมดมาจัดรูปให้อยู่ในรูปแบบ JSON ที่พร้อมสำหรับสร้างหน้าเว็บ โดยกำหนดคีย์ (Key) และค่า (Value) สำหรับการสร้างบทความ ดังนี้

1. Status ระบุสถานะของบทความที่สร้าง
2. Title ระบุส่วนหัวเรื่องของบทความ
3. Content ระบุส่วนเนื้อหาของบทความ
4. Date ระบุวันที่เผยแพร่ของบทความ
5. Feature Media ระบุ ID ของรูปที่ต้องการตั้งเป็น Feature Image

เมื่อได้ JSON ตามต้องการแล้ว ทำการสร้างบทความด้วยการเรียก API ของเวิร์ดเพรสโดยใส่ Header จากภาพที่ 3.20 เพื่อยืนยันตัวตนว่ามีสิทธิ์ในการสร้างบทความในเว็บไซต์นี้

```

def create_post(self, headers, host_url, data, img_id):
    #check / behind url
    host_url = host_url.strip('/')

    url = host_url + '/wp-json/wp/v2/posts'
    post_date = datetime.strptime(data['date'], '%B %d, %Y')

    post_data = {
        'status': 'publish',
        'title': data['title'],
        'content': data['content'],
        'date': post_date,
        'featured_media': img_id
    }

    r = requests.post(url, headers=headers, data=post_data)
    response = json.loads(r.content)
    post_url = response['guid']['rendered']
    return post_url

```

ภาพที่ 3.22 โค้ดฟังก์ชันการสร้าง Post



บทที่ 4

ผลการดำเนินการ

จากการพัฒนาเครื่องมือสำหรับคัดลอกเนื้อหาหน้าเว็บแล้วเสร็จ จำเป็นต้องมีการทดสอบการทำงานของเครื่องมือว่าการทำงานให้ผลลัพธ์ตรงตามจุดประสงค์ของการพัฒนาโปรแกรม ตามความต้องการของพนักงาน และมีประสิทธิภาพ สามารถนำไปใช้งานได้จริง

1. การตั้งค่าเว็บไซต์ก่อนการใช้งานเครื่องมือ
2. การเตรียมไฟล์ CSV สำหรับการใช้งาน
3. ตัวอย่างการใช้งานเว็บแอปพลิเคชัน
 - 1) Page Export
 - 2) Page Import
 - 3) Remove Element
 - 4) Post Export
 - 5) Post Import

4.1 การตั้งค่าเว็บไซต์ก่อนการใช้งานเครื่องมือ

ทำการติดตั้งและ Activate ปลั๊กอิน WP REST API และ JSON Basic Authentication

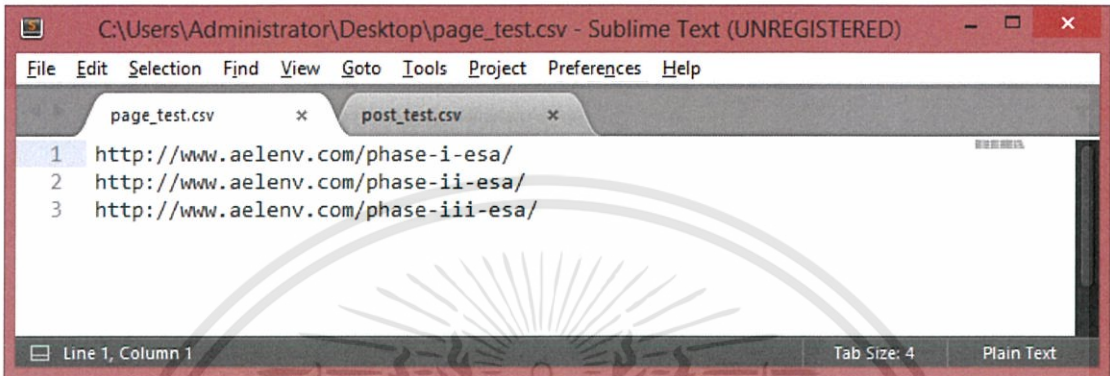


ภาพที่ 4.1 การตั้งค่าเว็บไซต์ก่อนการใช้งานเครื่องมือ

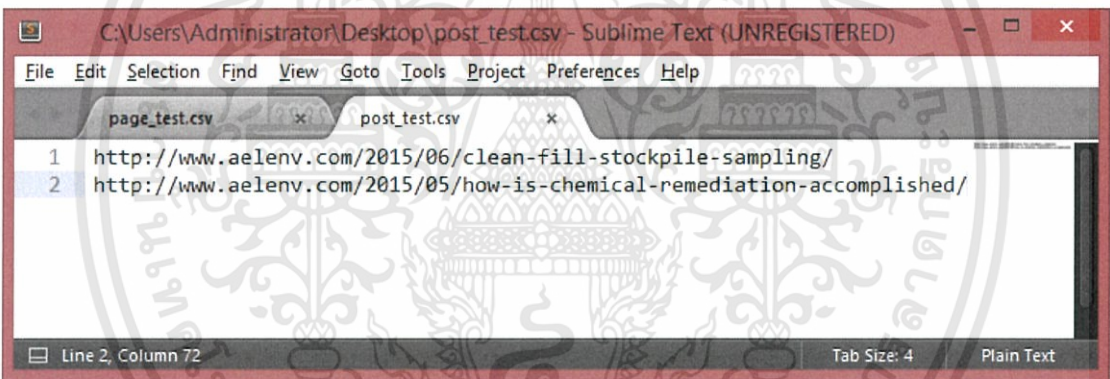
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การเตรียมไฟล์ซีเอสวีสำหรับการใช้งาน

สามารถลิสต์ลิงก์หน้าที่ต้องการคัดลอกใส่ใน Microsoft Excel แล้วเซฟเป็นสกุล CSV หรือลิสต์ใส่ใน Text editor และเซฟเป็น *.csv โดยลิงก์ที่ลิสต์มาในไฟล์เดียวกันจะต้องมีรูปแบบการจัดหน้าเหมือนกัน

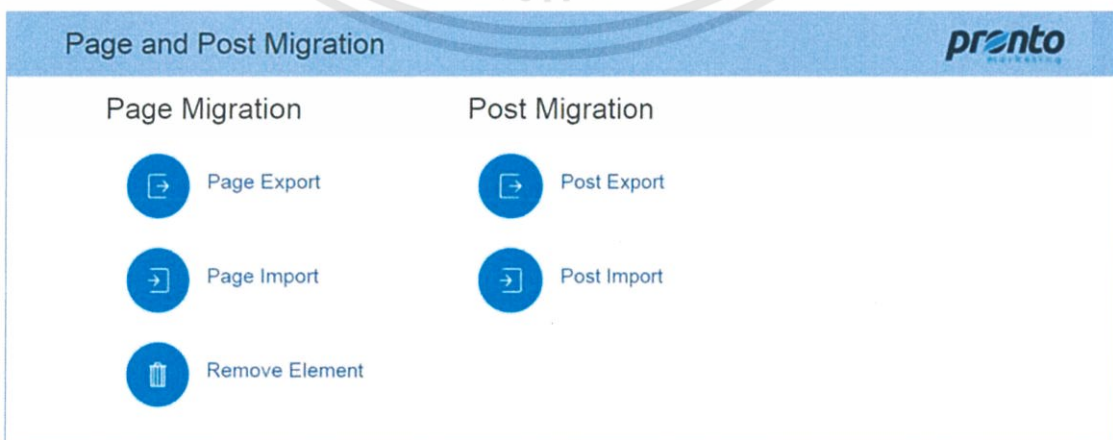


ภาพที่ 4.2 ตัวอย่างไฟล์ CSV สำหรับคัดลอกหน้าเว็บ (Page)



ภาพที่ 4.3 ตัวอย่างไฟล์ CSV สำหรับคัดลอก Post

4.3 ตัวอย่างการใช้งานเว็บแอปพลิเคชัน

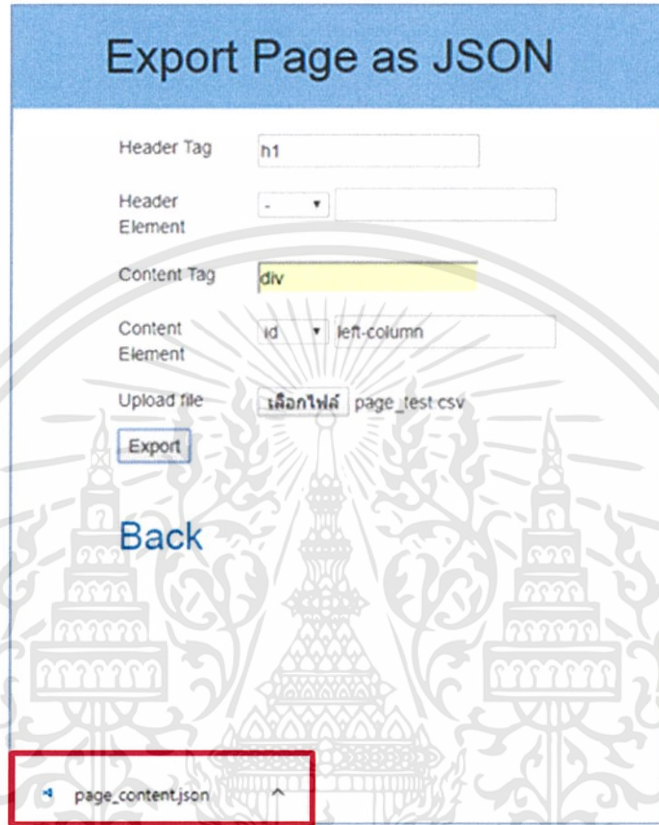


ภาพที่ 4.4 เมนูการใช้งานของเว็บแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

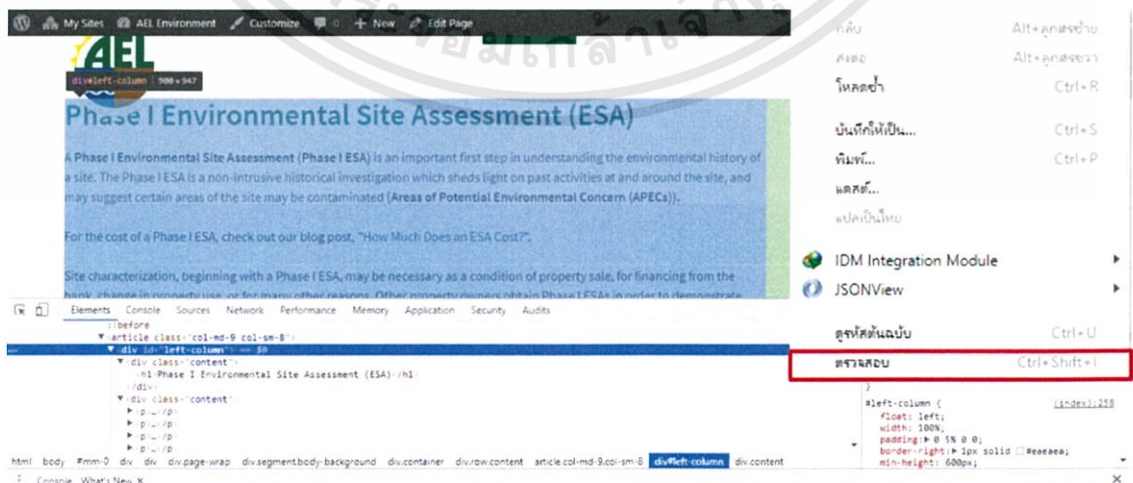
4.3.1 Page Export

สำหรับการคัดลอกหน้าเว็บ (Page) ทำโดยการนำไฟล์ CSV ที่ลิสต์ลิงก์เตรียมไว้มา Import กรอกข้อมูล Tag HTML ส่วนที่ต้องการคัดลอกในฟอร์มและกด Export จะได้ไฟล์ JSON ออกมาดังภาพ



ภาพที่ 4.5 ตัวอย่างการใช้งาน Page Export

รายละเอียด Tag HTML สามารถดูได้โดยการคลิกขวาและเลือกตรวจสอบ (Inspect) จากหน้าเว็บไซต์



ภาพที่ 4.6 วิธีการดู tag HTML จากหน้าเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
C:\Users\Administrator\Downloads\page_content.json - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
page_content.json x page_content(1).json x
1 [
2 {
3   "content": "\n<div class=\"content\">\n<h1>Phase I Environmental Site Assessment (ESA)</h1>\n</div>\n<div class=\"content\">\n<p>A\u00a0&str
4   "page_source": "http://www.aelenv.com/phase-i-esa/",
5   "title": "Phase I Environmental Site Assessment (ESA)"
6 },
7
8   "content": "\n<div class=\"content\">\n<h1>Phase II Environmental Site Assessment (ESA)</h1>\n</div>\n<div class=\"content\">\n<p>A&span Cla
9   "page_source": "http://www.aelenv.com/phase-ii-esa/",
10  "title": "Phase II Environmental Site Assessment (ESA)"
11 },
12
13   "content": "\n<div class=\"content\">\n<h1>Phase III ESA \u2013 Environmental Site Remediation Phase</h1>\n</div>\n<div class=\"content\">\n
14   "page_source": "http://www.aelenv.com/phase-iii-esa/",
15   "title": "Phase III ESA - Environmental Site Remediation Phase"
16 }
17 ]
```

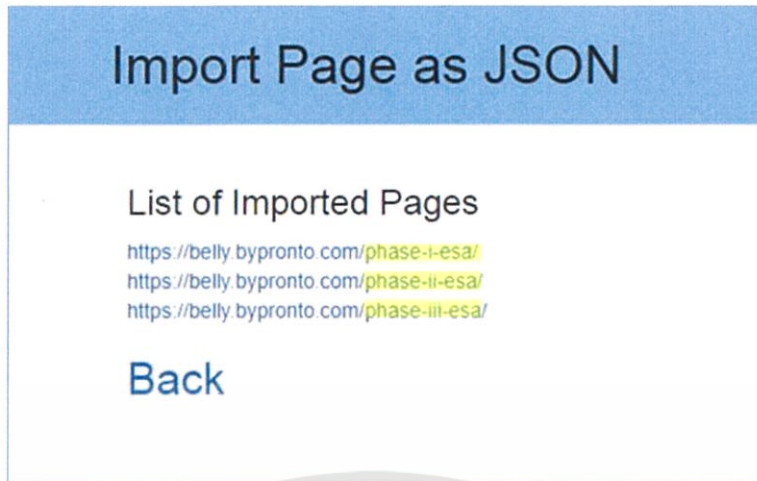
ภาพที่ 4.7 ข้อมูล JSON ที่ได้ฟังก์ชัน Page Export

4.3.2 Page Import

การสร้างหน้าเว็บ (Page) ทำโดยการนำไฟล์ JSON ที่ได้จากการ Export มา Import กรอกข้อมูลโดเมนของเว็บไซต์ใหม่ และ Username, Password ในฟอร์มเพื่อยืนยันตัวตนว่ามีสิทธิ์ในการสร้างหน้าของเว็บไซต์นี้และกด Import ดังภาพ

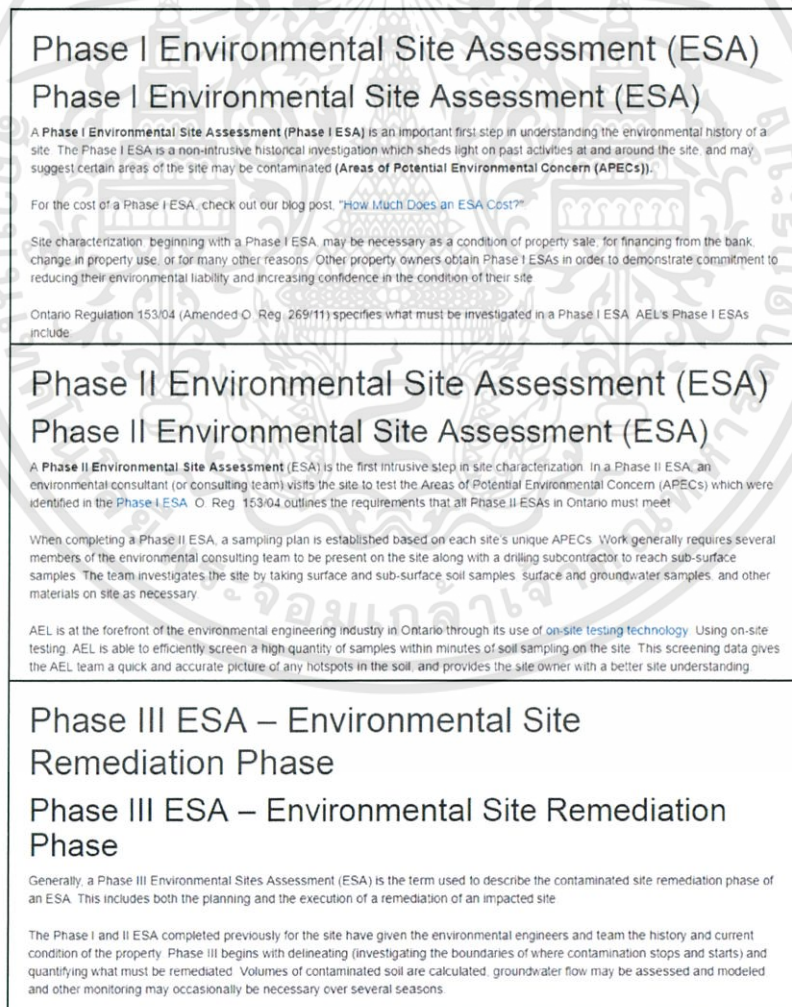
ภาพที่ 4.8 ตัวอย่างการใช้งาน Page Import

เมื่อระบบทำการสร้างหน้าเว็บเรียบร้อยแล้ว จะแสดงลิงก์ของหน้าเว็บที่สร้างใหม่แก่ผู้ใช้งาน ซึ่งจากลิงก์นี้ Slug จะเป็นไปตาม Slug ของหน้าเว็บไซต์เก่า ดังภาพ



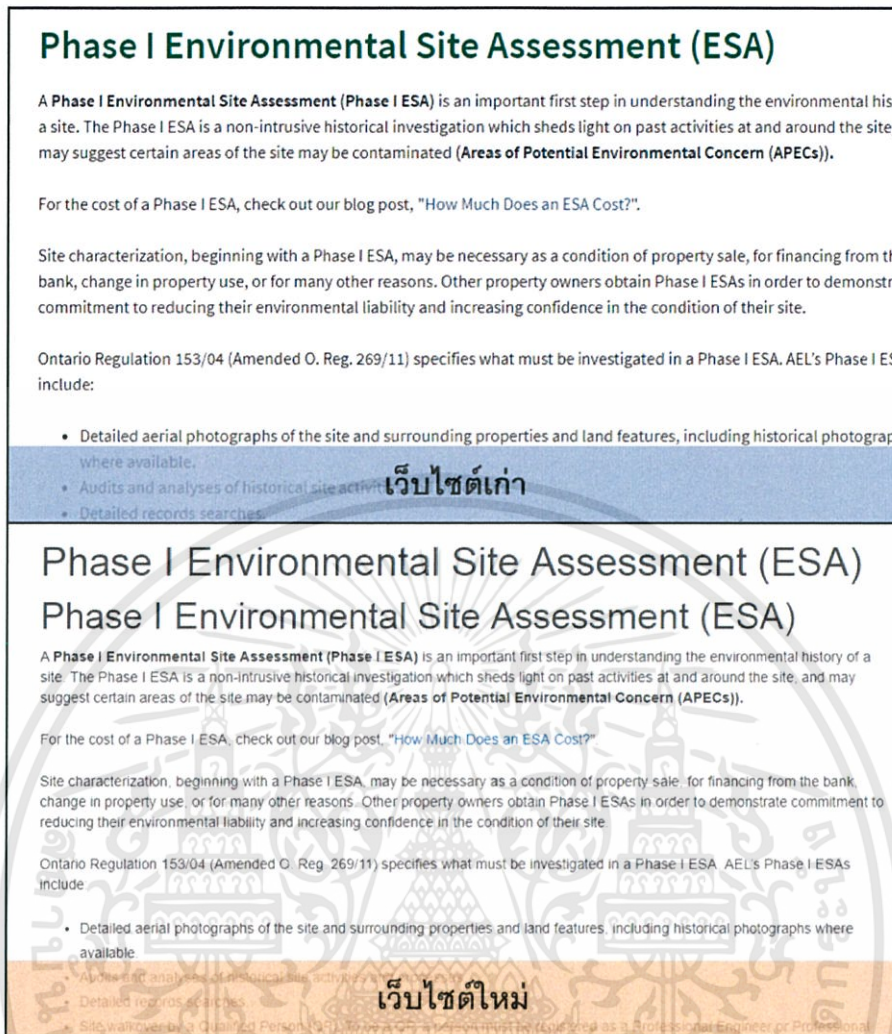
ภาพที่ 4.9 หน้าการแสดงผลหลังจากสร้าง Page

โดยหน้าที่สร้างขึ้นจากข้อมูล JSON (ภาพที่ 4.7) จะมีเนื้อหาตามที่คุณเลือกจากเว็บไซต์เก่า ดังภาพ



ภาพที่ 4.10 Page ที่สร้างจากฟังก์ชัน Page Import

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 46
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



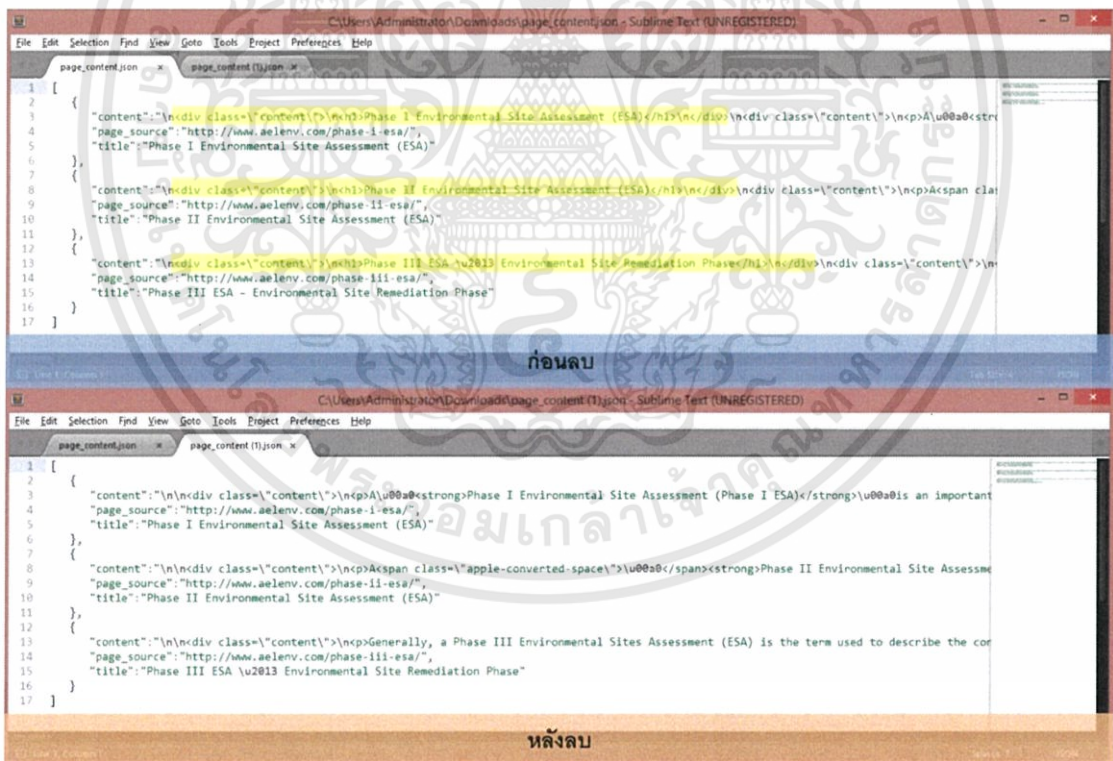
ภาพที่ 4.11 เปรียบเทียบ Page เว็บไซต์เก่าและเว็บไซต์ใหม่

4.3.3 Remove Element

จากไฟล์ JSON (ภาพที่ 4.7) และหน้าเว็บที่สร้างใหม่ (ภาพที่ 4.10) มีส่วนหัวเรื่องที่เกิดขึ้นมาใน Content ซึ่งสามารถลบส่วนที่ไม่ต้องการจาก Content ได้โดยใช้ฟังก์ชัน Remove Element การลบทำได้โดยการนำไฟล์ JSON ที่ได้จากการ Export มา Import กรอกข้อมูล Tag HTML ส่วนที่ต้องการลบจาก Content และกด Export จะได้ไฟล์ JSON ที่แก้ไขแล้วออกมา ดังภาพ

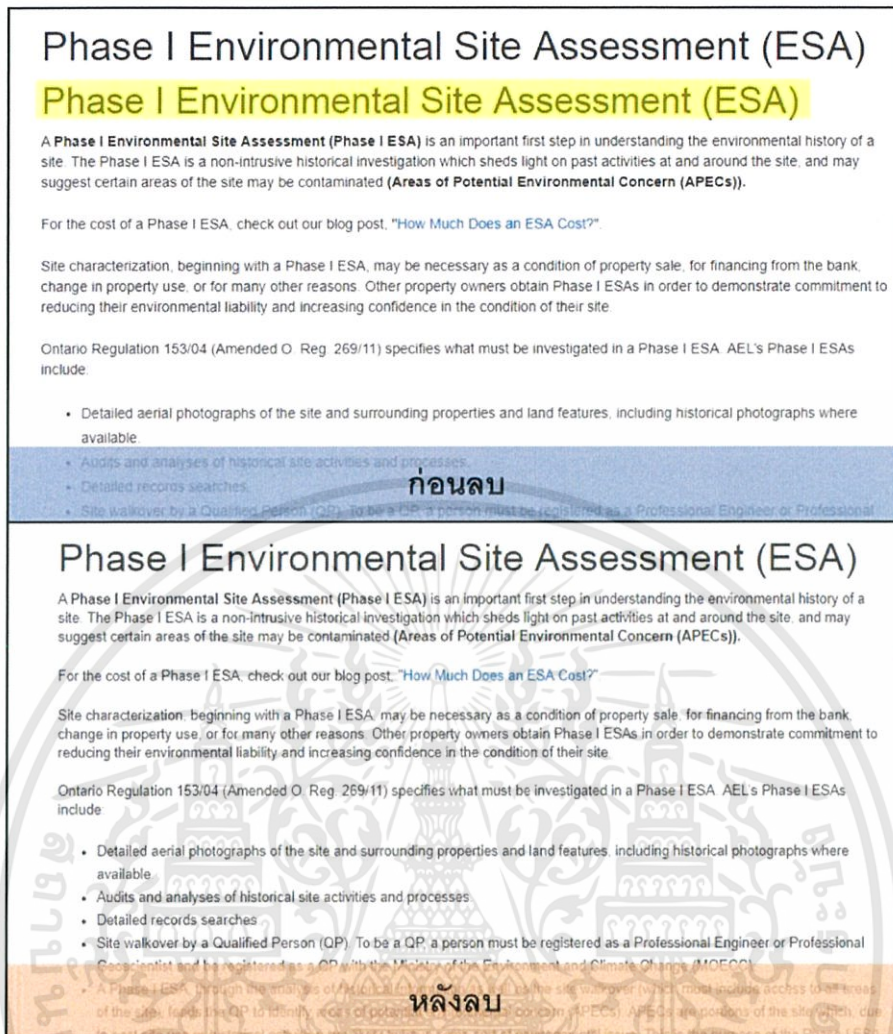


ภาพที่ 4.12 ตัวอย่างการใช้งาน Remove Element



ภาพที่ 4.13 เปรียบเทียบไฟล์ JSON ก่อนและหลังใช้ฟังก์ชัน Remove Element

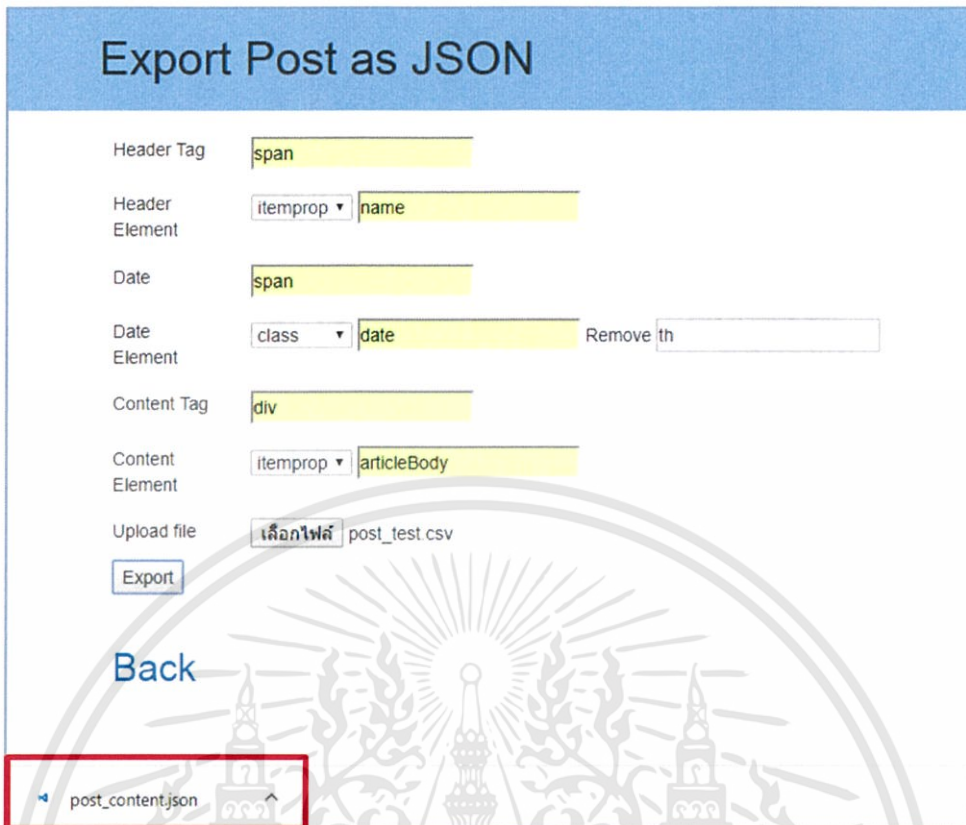
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ⁴⁸ วิชาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



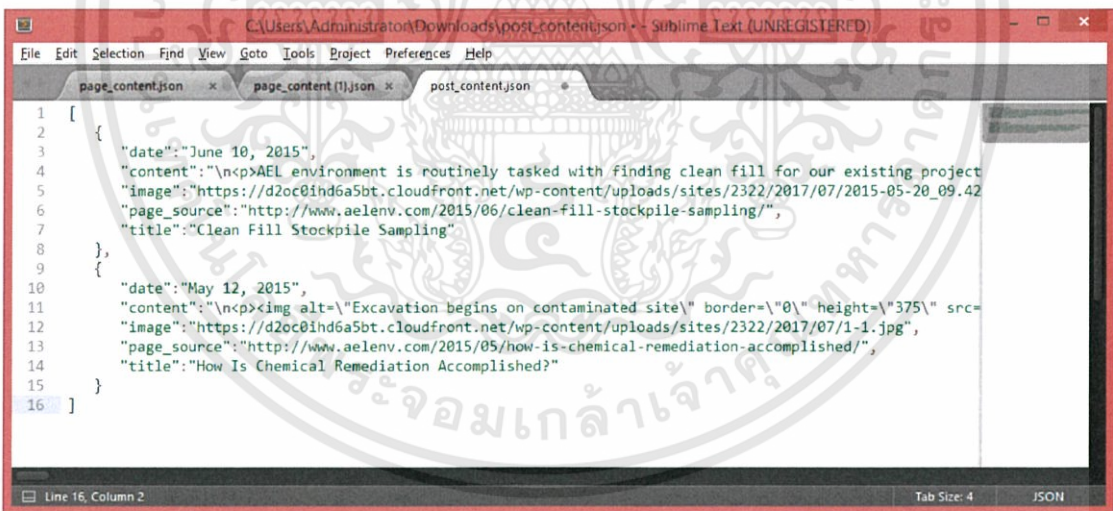
ภาพที่ 4.14 เปรียบเทียบหน้าเว็บที่สร้างก่อนและหลังใช้ฟังก์ชัน Remove Element

4.3.4 Post Export

สำหรับการคัดลอกบทความ (Post) ทำโดยการนำไฟล์ CSV ที่ลิสต์ลิงก์เตรียมไว้มา Import กรอกข้อมูล Tag HTML ส่วนที่ต้องการคัดลอกในฟอร์มและกด Export จะได้ไฟล์ JSON ออกมา ดังภาพ



ภาพที่ 4.15 ตัวอย่างการใช้งาน Post Export



ภาพที่ 4.16 ข้อมูล JSON ที่ได้ฟังก์ชัน Post Export

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ 50 ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 Post Import

การสร้างหน้าบทความ (Post) ทำโดยการนำไฟล์ JSON ที่ได้จากการ Export มา Import กรอกข้อมูลโดเมนของเว็บไซต์ใหม่ และ Username, Password ในฟอร์มเพื่อยืนยันตัวตนว่ามีสิทธิ์ในการสร้างหน้าของเว็บไซต์นี้และกด Import ดังภาพ

Import Post from JSON

Host URL

Upload File post_content.json

Import As: (wordpress id)

Username

Password

Cautions

First, you need to activate plugins as list below.

- 1) JSON Basic Authentication
- 2) WP REST API

[Back](#)

ภาพที่ 4.17 ตัวอย่างการใช้งาน Post Import

เมื่อระบบทำการสร้างบทความเรียบร้อยแล้ว จะแสดงลิงก์ของหน้าเว็บที่สร้างใหม่แก่ผู้ใช้งาน ดังภาพ

Import Post as JSON

List of Imported Posts

<https://belly.bypronto.com/2015/06/10/clean-fill-stockpile-sampling/>

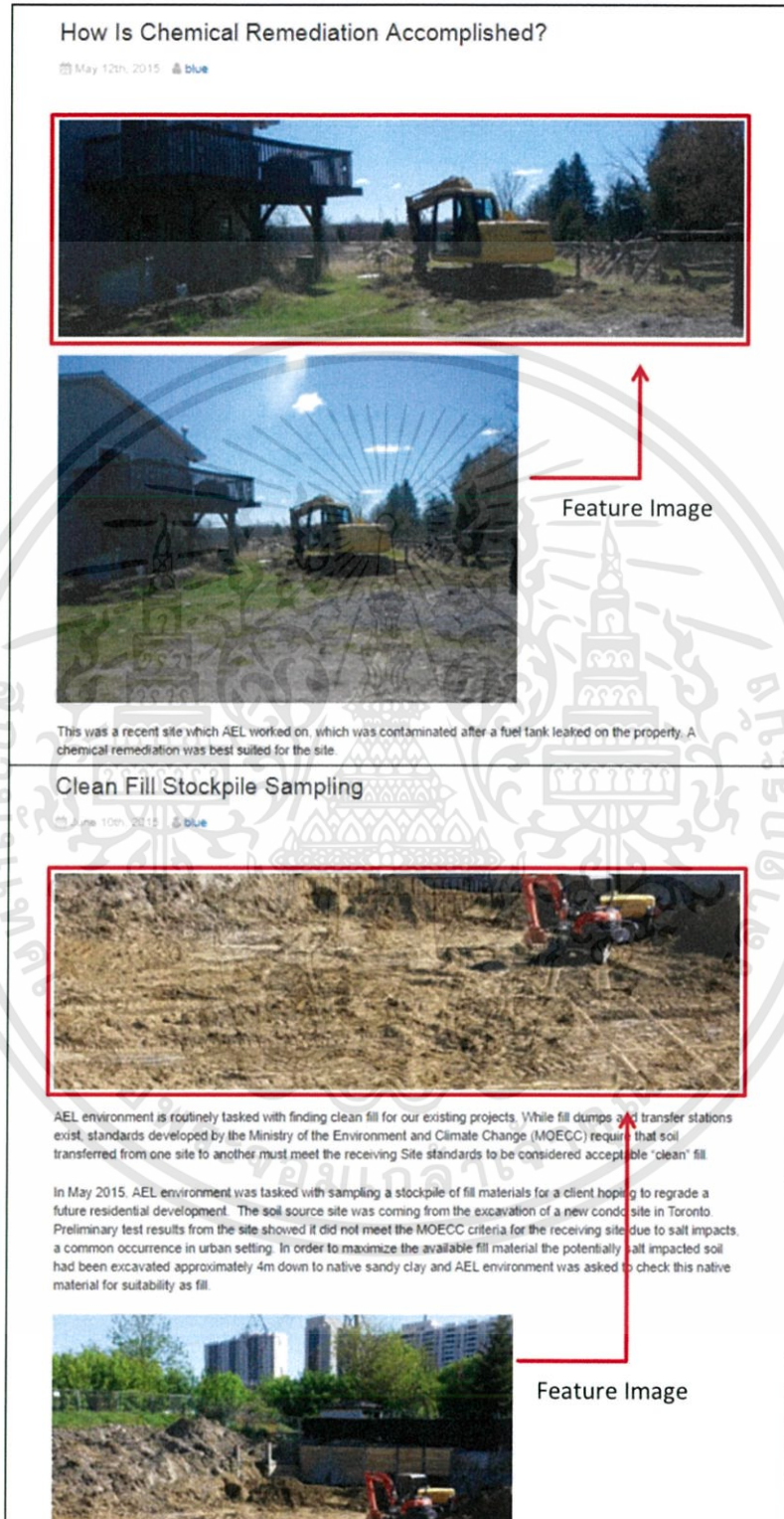
<https://belly.bypronto.com/2015/05/12/how-is-chemical-remediation-accomplished/>

[Back](#)

ภาพที่ 4.18 หน้าการแสดงผลหลังจากสร้าง Post

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยบทความที่สร้างขึ้นจากข้อมูล JSON (ภาพที่ 4.16) จะมีเนื้อหาตามที่ถูกเลือกจากเว็บไซต์เก่า รวมทั้งมีการใช้ภาพแรกของบทความเป็น Feature Image ดังภาพ



ภาพที่ 4.19 Post ที่สร้างจากฟังก์ชัน Post Import

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



This was a recent site which AEL worked on, which was contaminated after a fuel tank leaked on the property. A chemical remediation was best suited for the site.

This photo shows the excavator beginning to dig a hole for remediation trench. The hole would be approximately 5 m x 5 m down to groundwater, which was measured in the groundwater wells at about 1.7 m below surface.

Remediation is aimed at the groundwater, which contains a dissolved plume of hydrocarbons.

The ground was solid bedrock right below surface, but the layers were fairly thin. AEL's staff were fortunate and were able to get a lot of the bedrock out with the excavator teeth, but also used a rock breaker.



เว็บไซต์เก่า

This was a recent site which AEL worked on, which was contaminated after a fuel tank leaked on the property. A chemical remediation was best suited for the site.

This photo shows the excavator beginning to dig a hole for remediation trench. The hole would be approximately 5 m x 5 m down to groundwater, which was measured in the groundwater wells at about 1.7 m below surface.

Remediation is aimed at the groundwater, which contains a dissolved plume of hydrocarbons.

The ground was solid bedrock right below surface, but the layers were fairly thin. AEL's staff were fortunate and were able to get a lot of the bedrock out with the excavator teeth, but also used a rock breaker.



เว็บไซต์ใหม่

ภาพที่ 4.20 เปรียบเทียบ Post เว็บไซต์เก่าและเว็บไซต์ใหม่

บทที่ 5

สรุปผลการดำเนินการ

5.1 สรุปผลการดำเนินการ

เว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์เป็นเครื่องมือที่ให้พนักงานในบริษัทสามารถใช้ทำงานได้จริง ถูกพัฒนามาเพื่ออำนวยความสะดวกในการทำงานให้แก่พนักงานในบริษัทสำหรับงานย้ายเว็บไซต์ที่ต้องการย้ายหน้าเว็บและบทความจากเว็บไซต์เก่าไปเว็บไซต์ใหม่ซึ่งมีหน้าเว็บที่ต้องย้ายเป็นจำนวนมาก โดยเครื่องมือจะรวมขั้นตอนพื้นฐานในการคัดลอกเนื้อหาในหน้าเว็บไซต์ตั้งแต่การคัดลอกหัวเรื่อง เนื้อหา ตาวันโหลรูปภาพจากเว็บไซต์เก่ามาอัปโหลดในเว็บไซต์ใหม่อย่างครบถ้วน เปลี่ยนพารามิเตอร์ให้เป็นพารามิเตอร์ที่ถูกต้อง อีกทั้งยังมีฟีเจอร์อื่นๆ ตามที่ผู้ใช้งานต้องการ เช่น การแปลงพารามิเตอร์เป็นรีเลทีฟพารามิเตอร์ การตั้งค่า Feature Image แต่ละบทความ เป็นต้น จากที่กล่าวมาข้างต้น เครื่องมือนี้จะช่วยให้พนักงานสามารถทำงานได้อย่างสะดวกและรวดเร็ว ลดขั้นตอนในการทำงาน ไม่ต้องคัดลอกและแก้ไขทีละหน้าแบบเดิมที่เคยเป็น ลดความผิดพลาดในการทำงาน ทำให้ทำงานได้อย่างมีประสิทธิภาพและสามารถใช้กับทุกงานที่มีลักษณะเดียวกัน เป็นการลดภาระของ ทีม Production รวมถึงทีม Developer ด้วย

5.2 ปัญหาที่พบ

1. เนื่องจากไม่เคยพัฒนาเครื่องมือโดยใช้ Django framework มาก่อนจึงทำให้เสียเวลาในการศึกษาทำความเข้าใจโครงสร้างการทำงานเบื้องต้นพอสมควร
2. Response ที่ได้รับจากเซิร์ฟเวอร์ในการอัปโหลดรูปภาพหรือสร้างหน้าและบทความเป็น Object ที่มีความซับซ้อนจึงทำให้ค่อนข้างเสียเวลาในการเรียกใช้ข้อมูลที่ต้องการจาก Response ที่ได้รับ

5.3 แนวทางการแก้ไขปัญหา

1. พยายามหาตัวช่วยที่ทำให้ศึกษาการใช้งานได้เร็วโดยดูจากตามเว็บไซต์ต่างๆ ที่มีการสอนเรื่อง Django framework รวมถึงหาตัวอย่างในการใช้คำสั่งต่างๆ ที่จำเป็น
2. ในการเรียกใช้งาน Response ที่ได้จากเซิร์ฟเวอร์ควรทำการ Debug เพื่อให้เห็นข้อมูล Object ทั้งหมดและอาจมีการใช้เครื่องมือจากตามเว็บไซต์ต่างๆ ที่ช่วยให้อ่าน Object ง่ายขึ้นทำให้เรียกใช้งานได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ข้อเสนอแนะ

1. พัฒนาเครื่องมือโดยใช้เป็น Python 3 จากเดิมที่เป็น Python 2 เนื่องจาก Python 2 จะถูกยกเลิกการใช้งานในอนาคต
2. พัฒนาให้สามารถเลือกเนื้อหาส่วนที่ต้องการได้มากกว่า 1 จุดหรือสามารถเลือก Tag HTML ที่ต้องการได้ในกรณีที่ Tag HTML ที่ต้องการมีชื่อคลาสซ้ำกันมากกว่า 1 จุด
3. สามารถรวมฟังก์ชันบางฟังก์ชันเข้าด้วยกันได้ เช่น Export กับ Remove Element
4. ควรจัดรูปแบบการเขียนโค้ดให้เป็น Standard เพื่อง่ายต่อการกลับมาอ่านหรือพัฒนาต่อ

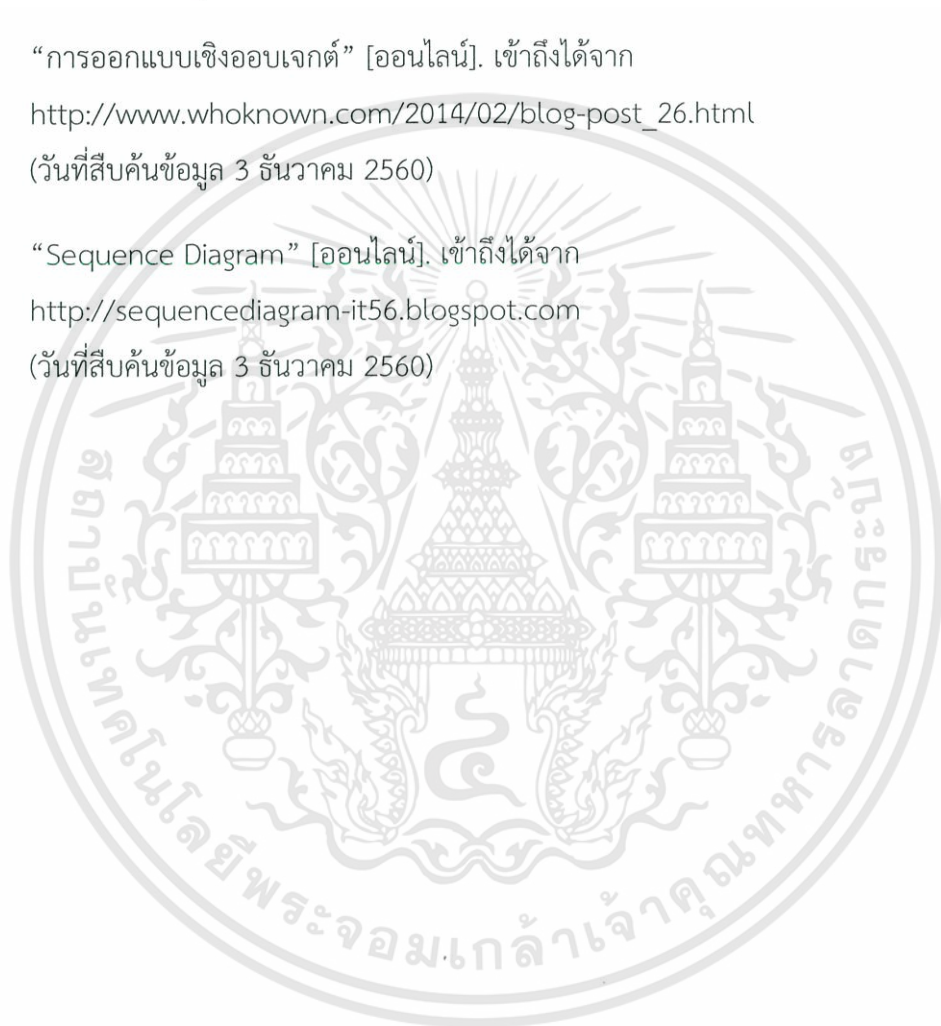


เอกสารอ้างอิง

- [1] “Web Application (เว็บแอปพลิเคชัน) คืออะไร” [ออนไลน์]. เข้าถึงได้จาก <https://www.mdsoft.co.th/ความรู้/359-web-application.html> (วันที่สืบค้นข้อมูล 18 พฤศจิกายน 2560)
- [2] “ข้อดี/ข้อเสีย ของ Web Application” [ออนไลน์]. เข้าถึงได้จาก <http://www.openlinksolutions.com/ข้อดีข้อเสีย-ของ-web-application> (วันที่สืบค้นข้อมูล 18 พฤศจิกายน 2560)
- [3] “WordPress คืออะไร” [ออนไลน์]. เข้าถึงได้จาก <http://www.wordpress.in.th/wordpress-article/wordpress-คืออะไร> (วันที่สืบค้นข้อมูล 18 พฤศจิกายน 2560)
- [4] “ข้อดีของการสร้างเว็บด้วย Wordpress” [ออนไลน์]. เข้าถึงได้จาก <http://www.jojho.com/2014/10/what-is-wordpress.html> (วันที่สืบค้นข้อมูล 18 พฤศจิกายน 2560)
- [5] “Page และ Post แตกต่างกันอย่างไรรใน WordPress” [ออนไลน์]. เข้าถึงได้จาก <https://blog.metrabyte.cloud/pageandpost-wordpress> (วันที่สืบค้นข้อมูล 18 พฤศจิกายน 2560)
- [6] “แนะนำภาษา Python” [ออนไลน์]. เข้าถึงได้จาก <http://marcuscode.com/lang/python/introduction> (วันที่สืบค้นข้อมูล 19 พฤศจิกายน 2560)
- [7] “ภาษาไพธอน (Python) คืออะไร” [ออนไลน์]. เข้าถึงได้จาก <http://python3.wannaphong.com/2015/07/python.html> (วันที่สืบค้นข้อมูล 19 พฤศจิกายน 2560)
- [8] “HTML คืออะไร” [ออนไลน์]. เข้าถึงได้จาก <http://www.mindphp.com/คู่มือ/73-คืออะไร/2026-html-คืออะไร.html> (วันที่สืบค้นข้อมูล 19 พฤศจิกายน 2560)

- [9] “CSS คืออะไร” [ออนไลน์]. เข้าถึงได้จาก
<http://www.mindphp.com/คู่มือ/73-คืออะไร/2193-css-คืออะไร.html>
(วันที่สืบค้นข้อมูล 19 พฤศจิกายน 2560)
- [10] “Django Framework” [ออนไลน์]. เข้าถึงได้จาก
<http://www.amplysoft.com/knowledge/what-is-django-framework-python.html>
(วันที่สืบค้นข้อมูล 19 พฤศจิกายน 2560)
- [11] “ข้อดีของ Django” [ออนไลน์]. เข้าถึงได้จาก
<http://ekkarach.com/2016/12/11/django-part-1>
(วันที่สืบค้นข้อมูล 19 พฤศจิกายน 2560)
- [12] “ข้อเสียของ Django” [ออนไลน์]. เข้าถึงได้จาก
<https://medium.com/@mitjy/คุณรู้จัก-django-หรือเปล่า-ดีจังอ่าว-72fcb4d11aa3>
(วันที่สืบค้นข้อมูล 19 พฤศจิกายน 2560)
- [13] “Bootstrap คืออะไร” [ออนไลน์]. เข้าถึงได้จาก
<http://www.siamhtml.com/bootstrap-คืออะไร-สอนวิธีใช้>
(วันที่สืบค้นข้อมูล 25 พฤศจิกายน 2560)
- [14] “Comma-separated values” [ออนไลน์]. เข้าถึงได้จาก
https://th.wikipedia.org/wiki/Comma-separated_values
(วันที่สืบค้นข้อมูล 25 พฤศจิกายน 2560)
- [15] “API คืออะไร” [ออนไลน์]. เข้าถึงได้จาก
<http://www.เกร็ดความรู้.net/api>
(วันที่สืบค้นข้อมูล 26 พฤศจิกายน 2560)
- [16] “Method กระบวนการทำงานหลักๆ ของ API” [ออนไลน์]. เข้าถึงได้จาก
<https://www.codebee.co.th/labs/api-คืออะไร-ทำความรู้จัก>
(วันที่สืบค้นข้อมูล 26 พฤศจิกายน 2560)
- [17] “REST (Representational State Transfer) คืออะไร” [ออนไลน์]. เข้าถึงได้จาก
<http://www.siamhttp.com/site/article/restful-web-service.html>
(วันที่สืบค้นข้อมูล 26 พฤศจิกายน 2560)

- [18] “Introducing JSON” [ออนไลน์]. เข้าถึงได้จาก
<http://www.json.org>
(วันที่สืบค้นข้อมูล 26 พฤศจิกายน 2560)
- [19] “Flowchart” [ออนไลน์]. เข้าถึงได้จาก
<http://meteeblog.blogspot.com/2015/01/flowchart.html>
(วันที่สืบค้นข้อมูล 3 ธันวาคม 2560)
- [20] “การออกแบบเชิงออบเจกต์” [ออนไลน์]. เข้าถึงได้จาก
http://www.whoknown.com/2014/02/blog-post_26.html
(วันที่สืบค้นข้อมูล 3 ธันวาคม 2560)
- [21] “Sequence Diagram” [ออนไลน์]. เข้าถึงได้จาก
<http://sequencediagram-it56.blogspot.com>
(วันที่สืบค้นข้อมูล 3 ธันวาคม 2560)





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ59ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

โปสเตอร์

สหกิจศึกษา ปีการศึกษา 2560
เว็บแอปพลิเคชันสำหรับย้ายหน้าและบทความเว็บไซต์ **pronto**
group Ltd.
Web Application for Page and Post Migration

ชื่อนักศึกษา : นางสาวสุวัจนี จันทร์โชค อาจารย์ที่ปรึกษา : ผศ. มยุรี เลิศเวชกุล
สาขาวิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ
เว็บแอปพลิเคชันนี้จัดทำขึ้นเพื่อย้ายหน้าเว็บและบทความจากเว็บไซต์เก่าจำนวนมากมายังเว็บไซต์ใหม่ที่สร้างด้วยโปรแกรมสำเร็จรูปเวิร์ดเพรสสำหรับบล็อกที่มีเว็บไซต์อยู่แล้วย้ายมาใช้บริการกับทางบริษัทพรอนโต้ กรุ๊ป จำกัด จึงพัฒนาเครื่องมือที่จะช่วยให้พนักงานสามารถทำงานได้อย่างสะดวกและรวดเร็ว

วัตถุประสงค์
1. เพื่อพัฒนาเครื่องมือที่ช่วยคัดลอกหน้าเว็บและบทความจากเว็บไซต์เก่ามายังเว็บไซต์ที่จัดการด้วยเวิร์ดเพรส
2. ช่วยให้พนักงานทำงานสะดวกและลดเวลาในการทำงาน
3. ช่วยลดความผิดพลาดในการทำงาน

เทคโนโลยีที่ใช้
▶ เทคโนโลยีที่ใช้ในการพัฒนา
▶ ภาษาที่ใช้ในการพัฒนา

การทำงานของระบบ
การทำงานของระบบประกอบด้วย 3 ส่วนหลัก คือ
1. **Export** เป็นส่วนที่คัดลอกข้อมูลในส่วนที่ผู้ใช้ระบุจากหน้าเว็บไซต์หรือบทความตามลิงก์ที่ลิศต์มาในไฟล์ CSV และนำมาแปลงเป็นไฟล์ JSON
2. **Import** เป็นส่วนที่จัดการข้อมูลที่ได้จากไฟล์ JSON เช่น การดาวน์โหลดรูปภาพจากเว็บเก่ามาเก็บในเว็บใหม่พร้อมกับการเปลี่ยนรูปภาพให้เป็นพิกเซลที่ถูกต้อง เป็นต้น จากนั้นก็เรียก API ของเวิร์ดเพรสเพื่อสร้างเว็บไซต์ใหม่
3. **Remove Element** ใช้สำหรับลบส่วนที่ไม่ต้องการเนื้อหา (Content) จากไฟล์ JSON และ Export ไฟล์ JSON ที่ลบส่วนที่ไม่ต้องการออกแล้วกลับมาให้ผู้ใช้เข้าไป Import สร้างหน้าหรือบทความต่อไป

ลำดับการทำงาน
1. CSV file (Example Page: http://www.xxx.com/page1, Example Post: http://www.xxx.com/blog/post1, http://www.xxx.com/blog/post2)
2. Page Export / Post Export
3. JSON file
4. Remove Element
5. Page Import / Post Import
6. Page / Post (New site (Wordpress))

ผลการดำเนินการ
1. สามารถคัดลอกเนื้อหาไปเว็บไซต์ใหม่ได้หลายหน้าในครั้งเดียว
2. ลดขั้นตอนและเวลาในการทำงาน
3. ลดความผิดพลาดในการทำงาน
4. สามารถใช้ได้กับทุกงานที่มีลักษณะเดียวกัน

กิตติกรรมประกาศ
ขอขอบคุณบริษัทพรอนโต้ กรุ๊ป จำกัด ที่ให้การอนุเคราะห์ในการทำโครงการสหกิจครั้งนี้ ขอขอบคุณ คุณอภิญา สิริคุณพร ที่ให้ออกสทาร์รับแอปพลิเคชันที่ได้ใช้งานจริง ขอขอบคุณทุกๆ ในแผนกทุกคนที่คอยให้ความช่วยเหลือ ให้คำปรึกษาและดูแลเอาใจใส่ตลอดการดำเนินงาน

ภาพที่ ก.1 โปสเตอร์

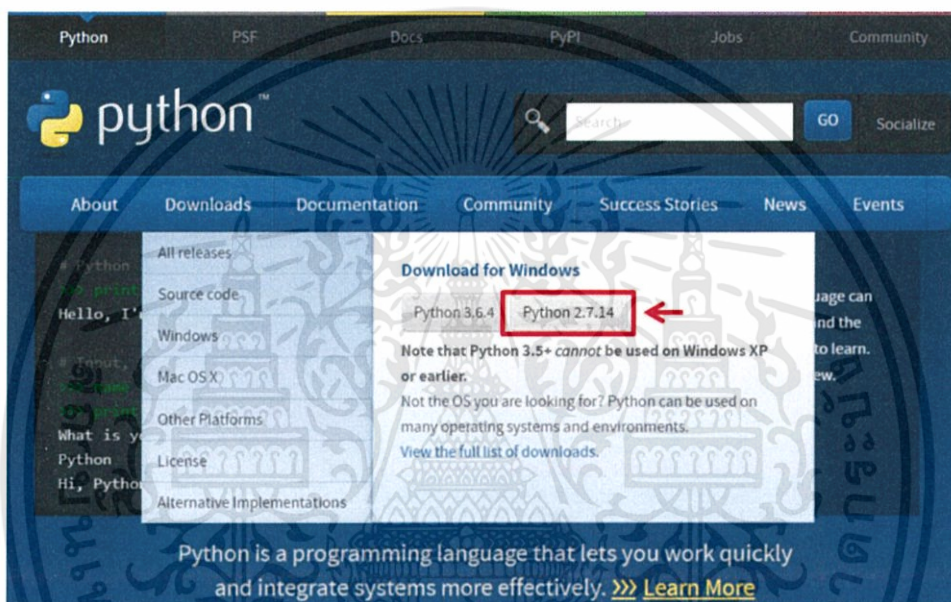
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข. การติดตั้งภาษาไพธอน

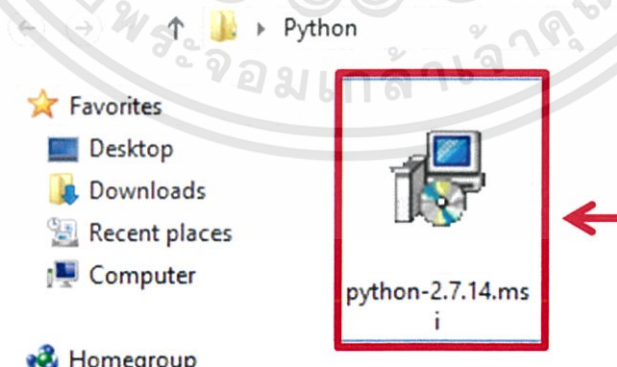
การติดตั้งภาษาไพธอน

1. ไปที่หน้าเว็บไซต์ <https://www.python.org> เพื่อดาวน์โหลดตัวติดตั้งภาษา โดยจะเลือกตามระบบปฏิบัติการของเครื่องคอมพิวเตอร์ ซึ่งในกรณีที่เป็น Windows สามารถดาวน์โหลดได้ ดังภาพที่ ข.1



ภาพที่ ข.1 หน้าเว็บไซต์สำหรับดาวน์โหลดตัวติดตั้งภาษา

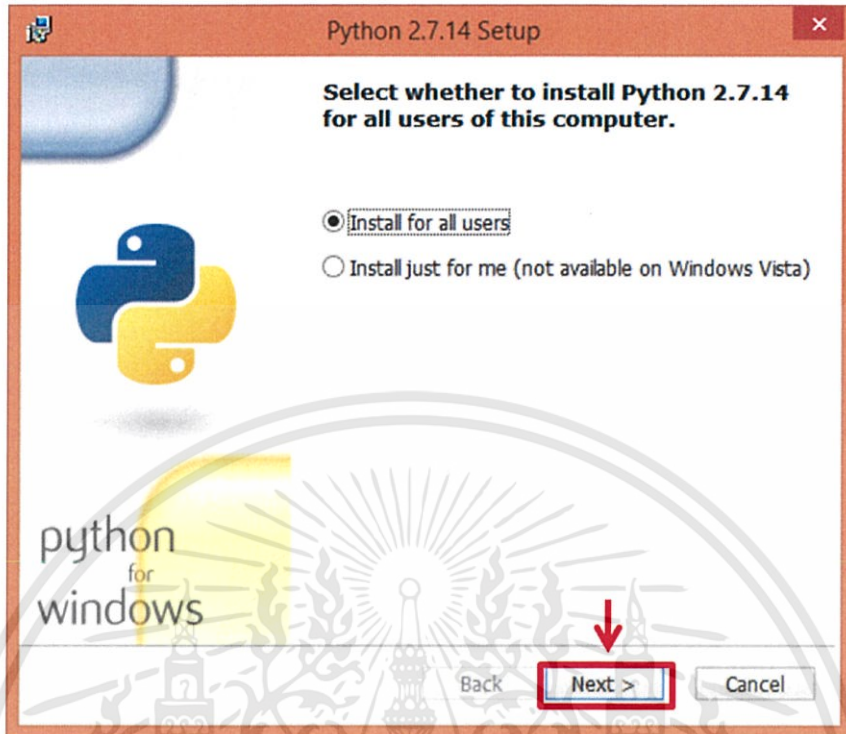
2. ไปยังโฟลเดอร์ที่ดาวน์โหลดตัวติดตั้งภาษา แล้วดับเบิลคลิกที่ตัวติดตั้ง ดังภาพที่ ข.2



ภาพที่ ข.2 ไฟล์ที่ใช้ในการติดตั้งภาษา

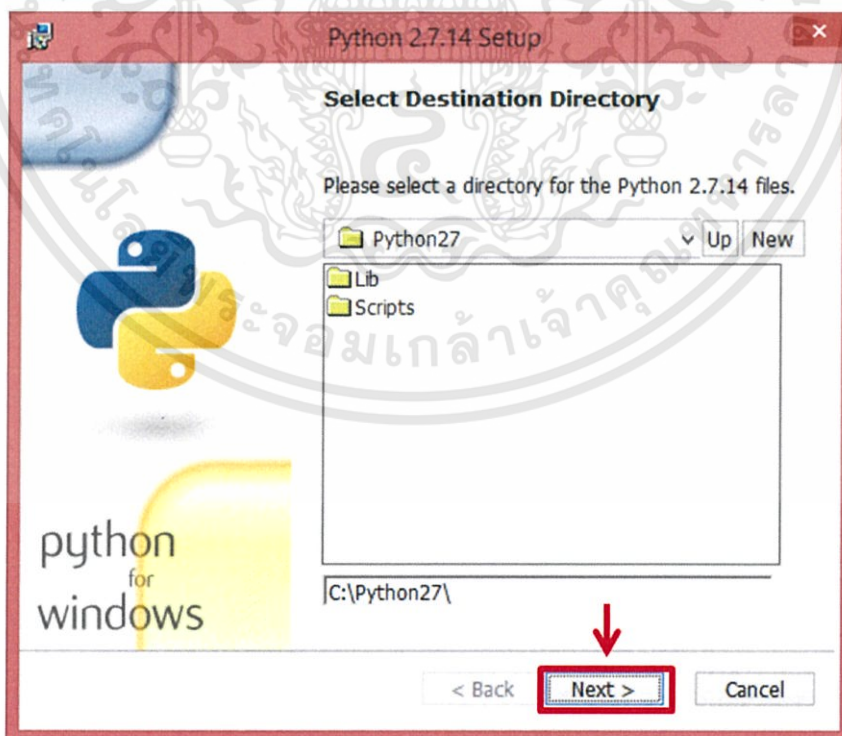
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อดับเบิลคลิกจะปรากฏหน้าต่าง ดังภาพที่ ข.3 แล้วกดปุ่ม Next



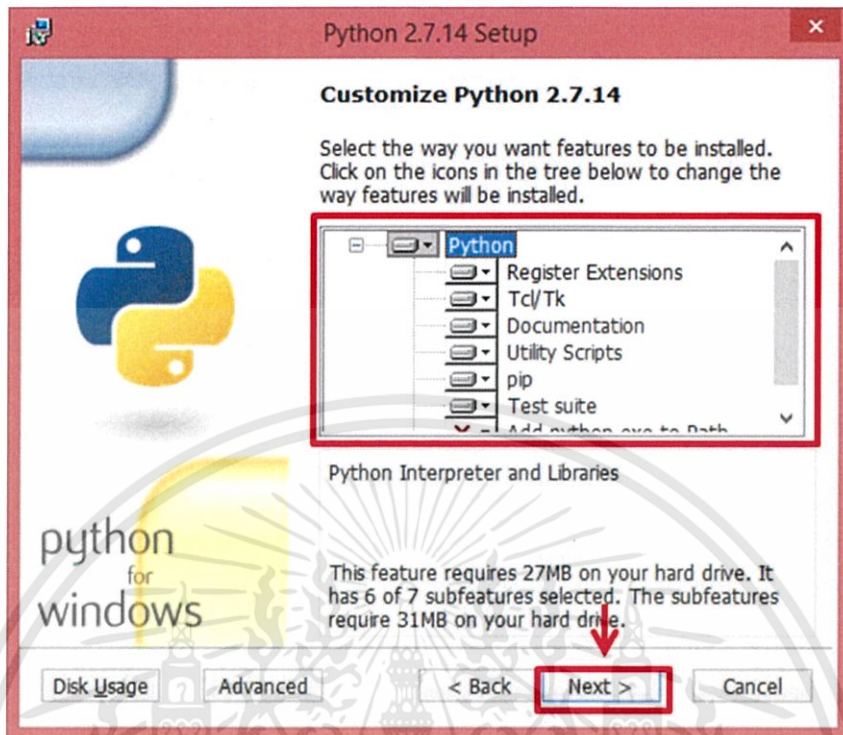
ภาพที่ ข.3 หน้าแรกของตัวติดตั้งภาษา

4. เลือกที่อยู่ของโปรแกรม แล้วกดปุ่ม Next ดังภาพที่ ข.4



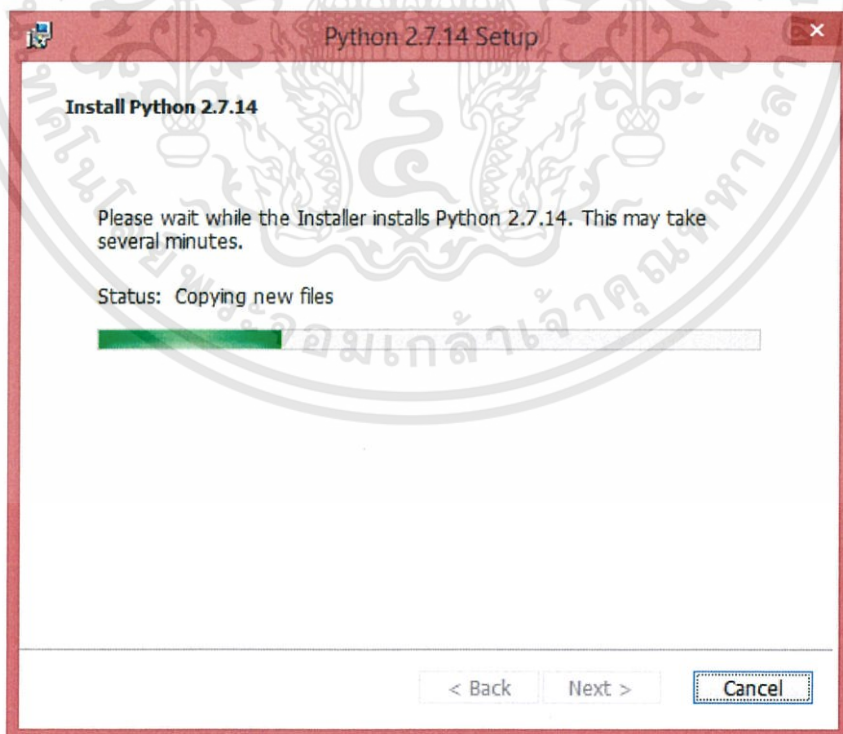
ภาพที่ ข.4 หน้าเลือกที่อยู่ตัวติดตั้งภาษา

5. เลือกฟีเจอร์ที่ต้องการใช้งาน แล้วกดปุ่ม Next ดังภาพที่ ข.5



ภาพที่ ข.5 หน้าเลือกฟีเจอร์ที่ใช้

6. รอการติดตั้งจนสำเร็จ ดังภาพที่ ข.6



ภาพที่ ข.6 หน้ากำลังติดตั้ง

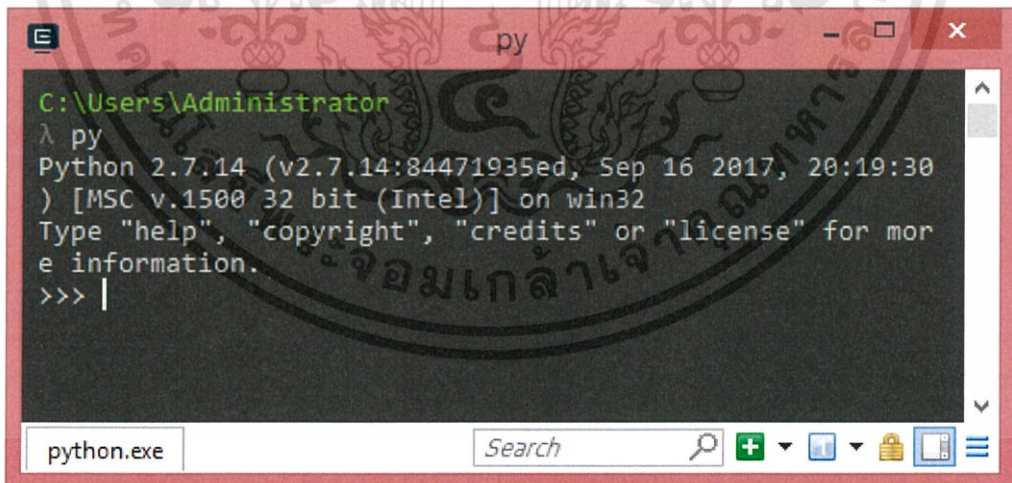
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ 65 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เมื่อติดตั้งเสร็จแล้ว กดปุ่ม Finish ดังภาพที่ ข.7



ภาพที่ ข.7 หน้าติดตั้งเสร็จสมบูรณ์

8. ทำการตรวจสอบการติดตั้ง โดยการเปิด Command line และพิมพ์คำสั่ง “python” หรือ “py” จะได้ผลลัพธ์ดังภาพที่ ข.8 ถือว่าการติดตั้งเสร็จสมบูรณ์



ภาพที่ ข.8 ตรวจสอบการติดตั้ง