รายงานสหกิจศึกษาฉบับสมบูรณ์

การเขียนซอฟต์แวร์เฟรมเวิร์คเพื่ออำนวยความสะดวกให้แก่ผู้ใช้

Creating a Software Framework to Improve the Usability for Users

นางสาวรุจิรวี คงดี

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

รายงานสหกิจศึกษาฉบับสมบูรณ์

การเขียนซอฟต์แวร์เฟรมเวิร์คเพื่ออำนวยความสะดวกให้แก่ผู้ใช้

Creating a Software Framework to Improve the Usability for Users

นางสาวรุจิรวี คงดี

ภาควิชาวิศกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงานสหกิจศึกษา การเขียนซอฟต์แวร์เฟรมเวิร์คเพื่ออำนวยความสะดวกให้แก่ผู้ใช้
ชื่อ-สกุล นักศึกษา นางสาวรุจิรวี คงดี
คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมคอมพิวเตอร์
ชื่อ-สกุล อาจารย์นิเทศ อาจารย์บัณฑิต พัสยา
อาจารย์จิระศักดิ์ สิทธิกร
ชื่อ – สกุล ผู้นิเทศงาน นายวีรยุทธ อินทร
ชื่อสถานประกอบการ บริษัท ทอมสัน รอยเตอร์ ซอฟต์แวร์ ประเทศไทย

บทคัดย่อ

การเขียนซอฟต์แวร์เฟรมเวิร์คในครั้งนี้ มีจุดประสงค์เพื่อปรับปรุงศูนย์ช่วยเหลือ (Help Center) ของโปรแกรม Eikon ซึ่งเป็นซอฟต์แวร์ที่ให้บริการวิเคราะห์ข้อมูลทางการเงินของบริษัท ทอมสัน รอยเตอร์ ซึ่งในการเขียนซอฟต์แวร์เฟรมเวิร์คนี้ได้ดำเนินการโดยการเขียนเว็บแอปพลิเคชัน (Web Application) เพื่อบอกวิธีการใช้งานและให้ข้อมูลเกี่ยวกับการแก้ปัญหาการใช้งานผลิตภัณฑ์ให้แก่ผู้ใช้งาน รวมทั้งให้ผู้ดูแลระบบสามารถแก้ไขหรือเพิ่มเติมข้อมูลได้ง่ายขึ้น และใช้เวลาในการ Deploy น้อยลง ซึ่งเว็บแอปพลิเคชันนี้เป็นเฟรมเวิร์คที่สามารถเรียกใช้ได้จากอีกหลากหลายผลิตภัณฑ์ของบริษัท

ในการดำเนินการเขียนซอฟต์แวร์เฟรมเวิร์ค ผู้จัดทำได้นำเอาหลักการวัฏจักรการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) เข้ามาใช้ คือ การวางแผน, การวิเคราะห์ความต้องการ, การออกแบบ, การพัฒนาซอฟต์แวร์, การทดสอบและนำไปใช้ และการบำรุงรักษาซอฟต์แวร์ โดยใช้ MEAN Stack เข้ามาใช้ในการเขียนเว็บแอปพลิเคชัน ทั้งนี้ ในขั้นตอนการออกแบบผู้จัดทำได้กำหนดโครงสร้างหลักที่จะดำเนินการทั้งหมด 16 โครงสร้าง ซึ่งมีโครงสร้างที่ดำเนินการได้สำเร็จทั้งหมด 12 โครงสร้าง และโครงสร้างที่ไม่ได้จัดทำทั้งหมด 4 โครงสร้าง คิดเป็นร้อยละ 75 ที่จัดทำสำเร็จ และคิดเป็นร้อยละ 25 ที่ไม่ได้ดำเนินการ เนื่องจากขณะที่ดำเนินการพัฒนาระบบผู้จัดทำได้รับคำแนะนำว่าเป็นโครงสร้างที่ไม่มีความจำเป็นกับระบบจึงไม่ได้ดำเนินการจัดทำ

ซอฟต์แวร์เฟรมเวิร์คที่พัฒนาขึ้นนี้ ช่วยทำให้การใช้งานศูนย์ช่วยเหลือมีความสะดวกมากขึ้น ไม่ว่าจะเป็นในส่วนของลูกค้าที่ใช้งานผลิตภัณฑ์จะได้รับความสะดวกในการใช้งานง่ายขึ้นและเข้าถึงข้อมูลได้รวดเร็วมากขึ้น รวมทั้งในส่วนของผู้ดูแลระบบจะสามารถแก้ไขหรือเพิ่มเติมข้อมูลได้ง่ายและรวดเร็วมากขึ้นซึ่งก็บรรลุตามวัตถุประสงค์ที่ผู้จัดทำได้กำหนดไว้

คำสำคัญ: เฟรมเวิร์ค เว็บแอปพลิเคชัน ศูนย์ช่วยเหลือ วัฏจักรการพัฒนาซอฟต์แวร์ MEAN Stack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Co-operative Title: Creating a Software Framework to Improve the Usability for Users

Student Intern Name: Ms. Rujiravee Kongdee

Faculty: Engineering **Department:** Computer

Advisor Name: Mr. Bundit Pasaya

Mr. Jirasak Sittigorn

Mentor Name: Mr. Veerayut Intorn

Company: Thomson Reuters Software Thailand

ABSTRACT

Creating a Software Framework to Improve the Usability for Users aims to improve the usability for users of the company's software product named Thomson Reuters Eikon, which is a software that provides information about the financial analysis of Thomson Reuters, by developing a web application to be a help center for telling users how to use the software product and providing information about troubleshooting. It also allows administrators to edit or add information more easily with less deployment time. This project is a framework that can be implemented from a wide variety of company software products.

In the development process, Software Development Life Cycle is used, which are Plan, Analysis, Design, Implementation, Testing and Integration, and Maintenance by using MEAN Stack to develop the web application. However, in the design process, there are 16 features expect to be completed but the result is 12 features completed and 4 features incompleted which are 75 percent completed and 25 percent incompleted because of the features were not necessary and not suitable for the system.

This software framework will make the usability of the help center to be more convenient whether for the customers that will experience the easiness of using and accessing to the content or the administrators that can edit the content easier and faster.

Keywords: Framework Web Application Help Center Software Development Life Cycle MEAN Stack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานสหกิจศึกษาฉบับนี้ สำเร็จได้ด้วยความอนุเคราะห์ข้อมูลที่เป็นประโยชน์จากหลายท่าน ซึ่งได้กรุณาให้การสนับสนุนและให้ความช่วยเหลือผู้จัดทำตั้งแต่เริ่มต้นโครงงานจนเสร็จสมบูรณ์

ขอกราบขอบพระคุณอาจารย์บัณฑิต พัสยา และอาจารย์จิระศักดิ์ สิทธิกร ซึ่งเป็นอาจารย์นิเทศของโครงงานสหกิจศึกษาที่ให้ความอนุเคราะห์ในการให้คำปรึกษา รวมไปถึงเสียสละเวลาให้คำแนะนำ ความคิดเห็นที่เป็นประโยชน์เกี่ยวกับแนวทางการจัดทำโครงงาน และคอยสอบถามความคืบหน้าไม่ว่าจะเป็นด้านข้อมูลเกี่ยวกับโครงงาน ด้านการปฏิบัติงาน การปฏิบัติตน การแก้ปัญหา ซึ่งทำให้การปฏิบัติงานสหกิจศึกษาของผู้จัดทำบรรลุตามวัตถุประสงค์ที่กำหนดไว้

ขอกราบขอบพระคุณคณาจารย์คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่าน ที่ได้กรุณาสั่งสอน ถ่ายทอด อบรมความรู้ ซึ่งผู้จัดทำก็ได้นำมาประยุกต์ใช้กับโครงงานชิ้นนี้

ขอขอบคุณบริษัท ทอมสัน รอยเตอร์ ซอฟต์แวร์ ประเทศไทย ที่ได้ให้โอกาสแก่ผู้จัดทำเพื่อเข้ามาปฏิบัติงานสหกิจศึกษา และขอขอบคุณนายวีรยุทธ อินทร พี่เลี้ยงที่ให้การดูแลตลอดการปฏิบัติงาน

ขอขอบคุณพี่และเพื่อนทุกคน ที่ได้มีส่วนช่วยให้ข้อคิดเห็น แลกเปลี่ยนความรู้ ตลอดจนให้กำลังใจแก่ผู้จัดทำเป็นอย่างดีเสมอมา

สุดท้ายนี้ซึ่งจะลืมไม่ได้เลยคือพระคุณของทุกคนในครอบครัว ขอกราบขอบพระคุณบิดามารดา และขอกราบระลึกถึงพระคุณของคุณยายที่ได้ล่วงลับไปในระหว่างการปฏิบัติงานสหกิจศึกษา ซึ่งทุกท่านได้อบรม สั่งสอน เลี้ยงดูให้ได้รับการศึกษาที่ดี รวมทั้งให้การสนับสนุนและเป็นกำลังใจอันสำคัญมาโดยตลอด จนสามารถประสบความสำเร็จในการจัดทำโครงงานสหกิจศึกษาชิ้นนี้

รุจิรวี คงดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ **หน้า**

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญภาพ VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญ 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 2

1.4 วิธีดำเนินการวิจัย 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 5

2.1 ทฤษฎีที่เกี่ยวข้อง 5

บทที่ 3 วิธีดำเนินการวิจัย 15

3.1 สถาปัตยกรรมของระบบ 15

3.2 การออกแบบ 19

3.3 ขั้นตอนการดำเนินงาน 28

บทที่ 4 ผลการวิจัย 29

4.1 ภาพรวมของระบบ 29

4.2 ผลการพัฒนาระบบ 30

4.3 ขั้นตอนการใช้งาน 31

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 41

5.1 สรุปผลการวิจัย 41

5.2 ประโยชน์ที่ได้จากการปฏิบัติงานสหกิจศึกษา 41

5.3 ปัญหาและอุปสรรคที่เกิดขึ้น 42

5.4 ข้อเสนอแนะ 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่ **หน้า**

เอกสารอ้างอิง 43

ภาคผนวก 45

ภาคผนวก ก 45

ภาคผนวก ข 47

ภาคผนวก ค 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

3.1 ตาราง Use case: View articles 20
3.2 ตาราง Use case: Search for articles 20
3.3 ตาราง Use case: Create articles 21
3.4 ตาราง Use case: Publish articles 21
3.5 ตาราง Use case: Edit articles 22
3.6 ตาราง Use case: Delete articles 22
3.7 ตาราง Test case 26
3.8 ตารางแสดงขั้นตอนการดำเนินงาน 28
4.1 ตารางผลการพัฒนาระบบ 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

1.1 โปรแกรม Thomson Reuters Eikon 1
1.2 หน้าช่วยเหลือของโปรแกรม Eikon แบบเก่า 2
1.3 Software Development Life Cycle 3
2.1 เฟรมเวิร์คต่าง ๆ 5
2.2 การทำงานของเว็บแอปพลิเคชัน 6
2.3 REST API 7
2.4 Bootstrap 8
2.5 ตัวอย่าง UI ของระบบปฏิบัติการ MacOS 8
2.6 ความแตกต่างระหว่าง UX และ UI 9
2.7 Django 9
2.8 ส่วนประกอบของ Software Stack 10
2.9 การทำงานของ MVC 11
2.10 WordPress 12
2.11 ตัวอย่างภาษา HTML ที่เขียนด้วยโปรแกรม Notepad 13
2.12 ตัวอย่างภาษา CSS 13
2.13 ตัวอย่างภาษา Javascript 14
3.1 สถาปัตยกรรมของ MEAN Stack 15
3.2 MongoDB 16
3.3 การเก็บข้อมูลในรูปแบบ JSON (Javascript Object Notation) 16
3.4 Express.js 17
3.5 Angular 6 17
3.6 MVC 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.7 Node.js 18
3.8 TinyMCE 19
3.9 Use case diagram ของระบบ 19
3.10 Database Schema ของระบบ 24
3.11 การออกแบบจอภาพของเว็บแอปพลิเคชันโดยใช้โปรแกรม Adobe XD CC 25
4.1 ภาพรวมของเว็บแอปพลิเคชันศูนย์ช่วยเหลือ 29
4.2 พิมพ์ข้อมูลที่ต้องการค้นหา 31
4.3 รายการของบทความที่ต้องการค้นหา 31
4.4 เนื้อหาของบทความ 22
4.5 เนื้อหาของบทความ (ต่อ) 22
4.6 เนื้อหาของบทความ (ต่อ) 33
4.7 หน้า Admin Portal 33
4.8 หน้า Create new 34
4.9 หน้าแก้ไขบทความ 34
4.10 บทความที่ทำการ Publish แล้ว 35
4.11 การแก้ไขบทความ 35
4.12 การแก้ไขบทความ (ต่อ) 36
4.13 การลบบทความ 36
4.14 การลบบทความ (ต่อ) 37
4.15 หน้า Admin Portal 37
4.16 หน้า Admin Portal (ต่อ) 38
4.17 หน้า Admin Profile 38
4.18 หน้าคำถามที่พบบ่อย 39
4.19 หน้าค้นหาโดย Tag 39
4.20 รายการบทความจากการค้นหาโดย Tag 40
4.21 หน้าติดต่อศูนย์บริการลูกค้า 40
ก.1 ตัวอย่างการเขียน Media Queries 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

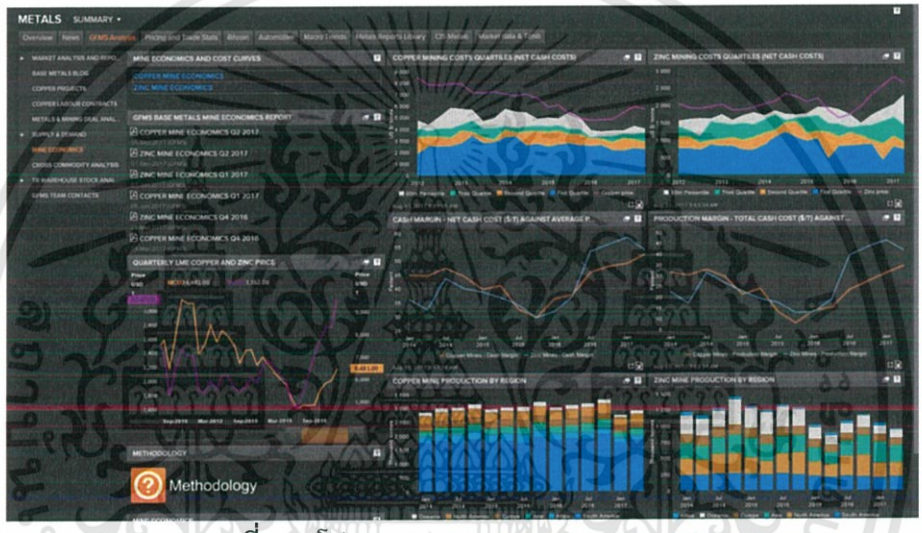
ก.2 ตัวอย่างการแบ่งคอลัมน์จาก Bootstrap 46
ก.3 Responsive Web Design 46
ข.1 โปรแกรม Visual Studio Code 47
ค.1 Git 48
ค.2 การทำงานของ Git 49
ค.3 โปรแกรม GitHub Desktop 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

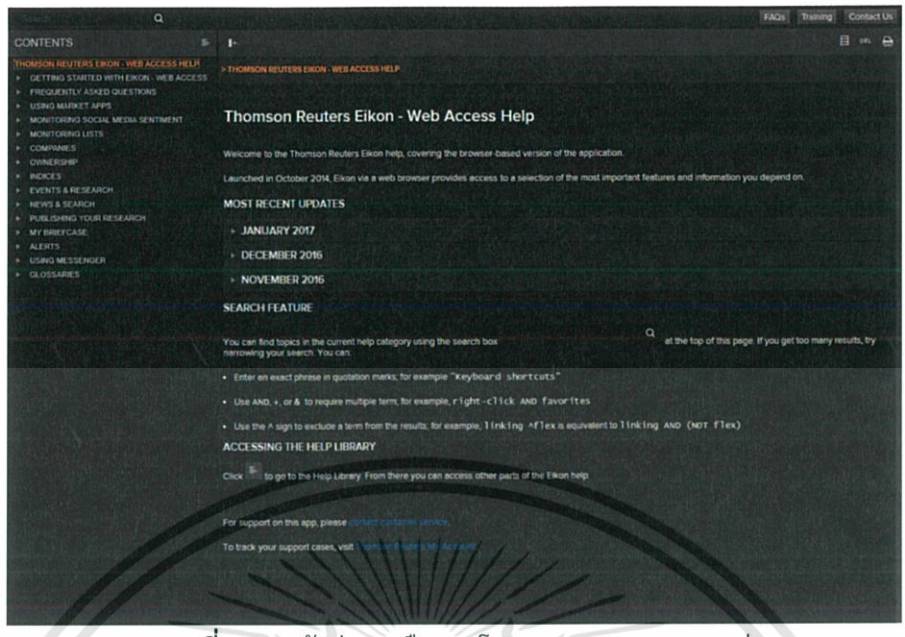
โปรแกรม Thomson Reuters Eikon (Eikon) เป็นซอฟต์แวร์ที่ให้บริการเกี่ยวกับการติดตามและวิเคราะห์ข้อมูลทางการเงิน มีการอัพเดตข้อมูลแบบเรียลไทม์ (Real time) ทั้งข้อมูลของตลาดหลักทรัพย์ อัตราแลกเปลี่ยนเงินตรา และข่าวสาร เป็นต้น เป็นผลิตภัณฑ์ที่อยู่ในฝ่ายโครงสร้าง Financial & Risk ของบริษัททอมสัน รอยเตอร์ ที่มีชื่อเสียงและได้รับการยอมรับทั่วโลก



ภาพที่ 1.1 โปรแกรม Thomson Reuters Eikon

ในตัวโปรแกรม Eikon นี้จะมีศูนย์ช่วยเหลือ (Help center) ที่ให้ข้อมูลเกี่ยวกับการใช้งานโปรแกรมเบื้องต้น เพื่ออำนวยความสะดวกและช่วยตอบคำถามเกี่ยวกับปัญหาการใช้งานให้แก่ลูกค้า แต่ปัญหาของศูนย์ช่วยเหลือนี้คือ เมื่อผู้ดูแลระบบต้องการที่จะแก้ไขหรือเพิ่มเติมข้อมูล จะต้องใช้เวลามากในการ Deploy อีกทั้งยังพบว่า การใช้งานศูนย์ช่วยเหลือนี้ค่อนข้างยาก การเข้าถึงข้อมูลที่ต้องการใช้เวลามาก และหน้าตาของบริษัทได้เล็งเห็นถึงปัญหาเหล่านี้ จึงมีแนวคิดที่จะปรับปรุงศูนย์ช่วยเหลือให้สามารถใช้งานได้ดีขึ้น เพื่ออำนวยความสะดวกทั้งแก่ลูกค้าผู้ใช้โปรแกรม และผู้ดูแลระบบที่ต้องการจะแก้ไขข้อมูล จึงเกิดเป็นโครงงานสหกิจศึกษาในเรื่องนี้ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 1.2 หน้าช่วยเหลือของโปรแกรม Eikon แบบเก่า

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อปรับปรุงศูนย์ช่วยเหลือของโปรแกรม Eikon ให้ใช้งานง่าย สะดวก และเข้าถึงข้อมูลได้รวดเร็วมากขึ้น
2. เพื่อปรับปรุงศูนย์ช่วยเหลือของโปรแกรม Eikon ให้สามารถแก้ไขหรือเพิ่มเติมข้อมูลได้ง่ายขึ้น และใช้เวลาในการ Deploy น้อยลง

1.3 ขอบเขตของการวิจัย

1. เว็บแอปพลิเคชันนี้ มีลักษณะเป็นเฟรมเวิร์คในโปรแกรม Eikon ที่แอปพลิเคชันอื่น ๆ สามารถเรียกใช้งานได้โดยไม่ต้องเขียนโปรแกรมใหม่ทุกครั้ง
2. ระบบฐานข้อมูลของเว็บแอปพลิเคชันนี้ ถูกออกแบบให้เหมาะสมกับการทำงานของระบบ ยังไม่ใช่ระบบฐานข้อมูลจริงของบริษัท
3. เว็บแอปพลิเคชันนี้ยังนำไปใช้ในโปรแกรม Eikon ไม่ได้ทันที เนื่องจากต้องผ่านการตรวจสอบจากหลายฝ่าย และปรับปรุง User Interface และ User Experience จากทีมงานอีกครั้ง
4. ข้อมูลที่แสดงในเว็บแอปพลิเคชันนี้เป็นข้อมูลจำลองเพื่อความเข้าใจเท่านั้น ยังไม่ใช่ข้อมูลจริงของบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีดำเนินการวิจัย

ในการดำเนินการโครงงานนี้ได้นำเอาหลักการวัฏจักรการพัฒนาซอฟต์แวร์ (Software Development Life Cycle) เข้ามาใช้ ซึ่งประกอบไปด้วย การวางแผน (Planning), การวิเคราะห์ความต้องการ (Analysis), การออกแบบ (Design), การพัฒนาซอฟต์แวร์ (Implementation), การทดสอบและนำไปใช้ (Testing and Integration) และการบำรุงรักษาซอฟต์แวร์ (Maintenance) ดังต่อไปนี้

ภาพที่ 1.3 Software Development Life Cycle

1. การวางแผน (Planning)
   - รับ Requirement จาก Project Manager
2. การวิเคราะห์ความต้องการ (Analysis)
   - วิเคราะห์ ตีความ Requirement ที่ได้ ว่าปัญหาอยู่ตรงไหน จะแก้ไขอย่างไร
3. การออกแบบ (Design)
   - ออกแบบขั้นตอนการทำงานของระบบ (Design Workflow)
   - ออกแบบหน้าจอตอบสนองผู้ใช้งาน (User Interface: UI)
   - ออกแบบความพึงพอใจของผู้ใช้ที่มีต่อระบบ (User eXperience: UX)
   - ออกแบบระบบฐานข้อมูลของระบบ (Database Design)
   - เลือกใช้เครื่องมือ (Tools) ที่จะนำมาใช้ในการพัฒนา
4. การพัฒนาซอฟต์แวร์ (Implementation)
   - พัฒนาโปรแกรมตามที่ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. การทดสอบและนำไปใช้ (Testing and Integration)
    - ทดสอบเพื่อให้มั่นใจว่าเว็บแอปพลิเคชันสามารถทำงานได้ตรงตามความสามารถพื้นฐาน
    - ทดสอบเว็บแอปพลิเคชัน ว่ามีส่วนใดที่ทำงานไม่ตรงตามความต้องการหรือไม่
6. การบำรุงรักษาซอฟต์แวร์ (Maintenance)
    - แก้ไขปรับปรุงระบบเมื่อพบข้อผิดพลาดเพิ่มเติมหลังการใช้งาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ใช้งานโปรแกรมจะได้รับความสะดวกจากการใช้ศูนย์ช่วยเหลือมากขึ้น
2. ผู้ดูแลระบบสามารถแก้ไขหรือเพิ่มเติมข้อมูลของศูนย์ช่วยเหลือได้ง่ายขึ้น และสามารถทำการ Deploy ได้เร็วขึ้น
3. ได้เรียนรู้การออกแบบ และทำเว็บแอปพลิเคชัน ทั้งระบบหน้าบ้าน (Front-end) และระบบหลังบ้าน (Back-end)
4. ได้ศึกษาการใช้งานเครื่องมือต่าง ๆ ที่เกี่ยวกับการพัฒนาเว็บแอปพลิเคชัน รวมไปถึงกระบวนการวางแผนการทำงาน และการปฏิบัติที่ดี (Best Practice)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 เฟรมเวิร์ค (Framework)

เฟรมเวิร์คคือ โครงสร้างของโค้ดที่มีฟังก์ชันการทำงานต่าง ๆ ที่มีความยืดหยุ่นสูง พร้อมจะนำไปใช้งานเพื่อประยุกต์ใช้ หรือต่อเติมในส่วนต่าง ๆ ของซอฟต์แวร์ได้ทันที เป็นการอำนวยความสะดวกแก่โปรแกรมเมอร์ มีข้อดีคือช่วยทำให้เข้าใจโครงสร้างของโปรแกรมได้ง่าย ไม่ต้องทำการเขียนโค้ดซ้ำ ๆ หากต้องการแก้ไข ก็แก้ไขที่เฟรมเวิร์คที่เดียว แล้วนำไปใช้ต่อได้ทันที สามารถเขียนโปรแกรมให้เสร็จได้ในเวลาที่รวดเร็วมากขึ้น และการใช้เฟรมเวิร์คจะช่วยในเรื่องของความปลอดภัยของซอฟต์แวร์ได้มากกว่า เพราะได้มีการวางโครงสร้างเอาไว้เรียบร้อยแล้ว การใช้เฟรมเวิร์คนั้นเหมาะสำหรับการพัฒนาซอฟต์แวร์ที่เป็นทีม เพราะจะทำให้การเขียนโค้ดเป็นไปในทางเดียวกัน

ยกตัวอย่างเฟรมเวิร์ค เช่น

- Web framework
- JavaScript framework
- PHP framework
- Application framework
- MVC framework

ภาพที่ 2.1 เฟรมเวิร์คต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 เว็บแอปพลิเคชัน (Web Application)

เว็บแอปพลิเคชันคือ เว็บที่นำเอาโปรแกรมประยุกต์ (Application) มารวมไว้ในเว็บโดยใช้ภาษาคอมพิวเตอร์เชิงโปรแกรมมิ่ง (Programming Language) เช่น Node.js, Python, PHP ฯลฯ โดยตัวเว็บแอปพลิเคชัน จะถูกติดตั้งไว้ที่เว็บเซิร์ฟเวอร์ (Web server) ที่คอยให้บริการและผู้ใช้ (Client) สามารถเรียกดูได้ผ่านเบราว์เซอร์ (Browser) เช่น Google Chrome, Safari, Internet Explorer โดยไม่ต้องติดตั้งโปรแกรมใด ๆ เพิ่มเติม ตัวอย่างของเว็บแอปพลิเคชัน เช่น เว็บเมล การพาณิชย์อิเล็กทรอนิกส์ กระดานสนทนา บล็อก วิกิ เป็นต้น

2.1.2.1 การทำงานของเว็บแอปพลิเคชัน

ภาพที่ 2.2 การทำงานของเว็บแอปพลิเคชัน

1. ผู้ใช้ส่งคำขอ (Request) ไปยังเว็บเซิร์ฟเวอร์ เพื่อเรียกใช้งานเว็บแอปพลิเคชันบนอินเทอร์เน็ตผ่านเว็บเบราว์เซอร์บนเครื่องคอมพิวเตอร์ของตนเอง

2. เว็บเซิร์ฟเวอร์ส่งต่อคำขอไปยังเว็บแอปพลิเคชันเซิร์ฟเวอร์ (Web application server) ที่เป็นเครื่องคอมพิวเตอร์ให้บริการ

3. เว็บแอปพลิเคชันเซิร์ฟเวอร์ทำตามคำขอ เช่น การขอข้อมูลจากฐานข้อมูล หรือประมวลผลต่าง ๆ เพื่อให้ออกมาเป็นผลลัพธ์ (Results)

4. เว็บแอปพลิเคชันเซิร์ฟเวอร์ส่งผลลัพธ์กลับไปยังเว็บเซิร์ฟเวอร์

5. เว็บเซิร์ฟเวอร์ส่งผลลัพธ์ไปยังผู้ใช้โดยการแสดงผลผ่านเว็บเบราว์เซอร์

2.1.2.2 เว็บเซิร์ฟเวอร์ (Web Server)

เว็บเซิร์ฟเวอร์คือ เครื่องคอมพิวเตอร์ที่ให้บริการเก็บเว็บไซต์ (Web site) และแสดงผลให้แก่ผู้ใช้ที่ร้องขอข้อมูลผ่านโปรโตคอล HTTP (Hyper Text Transfer Protocol) ผ่านเว็บเบราว์เซอร์ โดยเว็บเซิร์ฟเวอร์มักให้บริการอยู่บนพอร์ต 80 โปรแกรมที่นิยมนำมาใช้ให้บริการเว็บเซิร์ฟเวอร์ ได้แก่ อาปาเช่ (Apache Web Server) และ ไมโครซอฟต์ไอไอเอส (Microsoft IIS = Internet Information Server)

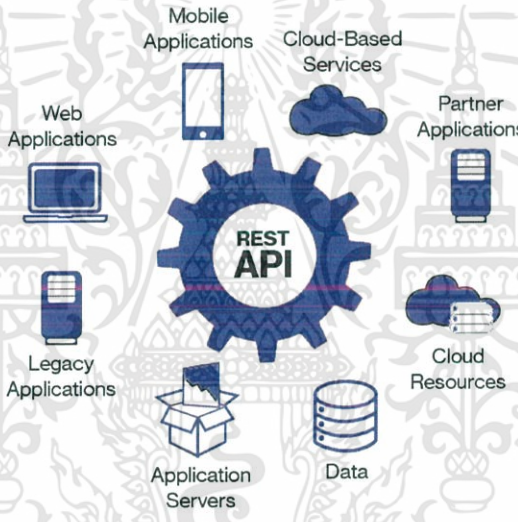
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.3 REST (Representational State Transfer)

REST คือ แนวทางการสร้างเว็บเซอร์วิส (Web Service) ชนิดหนึ่งที่ใช้สื่อสารกันบนอินเทอร์เน็ต โดยเรียกใช้งานผ่านทาง HTTP Method ดังนี้

- GET คือ การดึงข้อมูล
- POST คือ การสร้างข้อมูล
- PUT คือ การแก้ไขข้อมูล
- DELET คือ การลบข้อมูล

ข้อดีของ REST คือ ทำงานอยู่บนโปรโตคอล HTTP จึงทำให้พัฒนาได้ง่าย สนับสนุนข้อมูลหลายรูปแบบ เช่น XML, JSON, Plain text เป็นต้น อีกทั้งยังมีประสิทธิภาพการทำงานที่ดีและรองรับเรื่องการ caching ข้อมูลอีกด้วย



ภาพที่ 2.3 REST API

2.1.2.4 API (Application Programming Interface)

API คือ ช่องทางการเชื่อมต่อระหว่างเว็บไซต์กับเว็บไซต์ หรือผู้ใช้กับเซิร์ฟเวอร์ หรือระหว่างเซิร์ฟเวอร์กับเซิร์ฟเวอร์ก็ได้ เปรียบ API เป็นภาษาคอมพิวเตอร์ ที่ทำให้คอมพิวเตอร์สื่อสารแลกเปลี่ยนข้อมูลกันได้ โดยหน้าที่ของ API คือ ช่วยในการเข้าถึง ส่งออก นำเข้าข้อมูลต่าง ๆ โดยเจ้าของเว็บไซต์จะเป็นผู้กำหนดขอบเขตการใช้งานของ API ของตนเอง การใช้ API มีประโยชน์คือ ช่วยทำให้พัฒนาเว็บแอปพลิเคชันได้อย่างสะดวกรวดเร็วโดยไม่ต้องเข้าไปแก้ไขโค้ดคำสั่ง รวมไปถึงสามารถรับส่งข้อมูลข้ามเซิร์ฟเวอร์ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

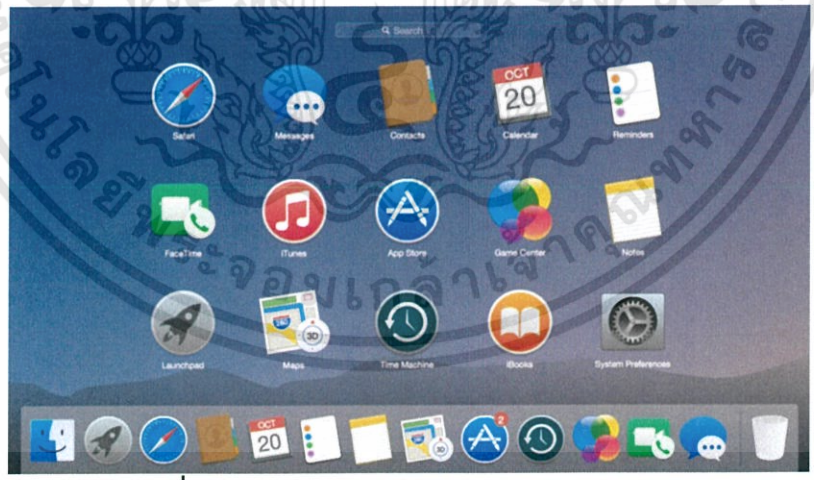
2.1.3 Front-end

Front-end คือ ส่วนติดต่อผู้ใช้ หรือที่เรียกว่าหน้าบ้าน เป็นส่วนแสดงผลหน้าจอต่าง ๆ ให้ผู้ใช้ได้เห็น เช่น หน้าเว็บเพจ หน้าโฮม โดยจะเป็นส่วนดึงดูดให้ผู้ใช้สนใจ และอยากเข้าเว็บไซต์นั้น ๆ มากขึ้น ส่วนมากจะใช้ภาษา HTML, CSS, Javascript ในการพัฒนา โดยเฟรมเวิร์คที่ใช้ในการเขียนมีมากมาย ยกตัวอย่างเช่น Bootstrap, Semantic-UI, Foundation, Materialize เป็นต้น

ภาพที่ 2.4 Bootstrap

2.1.3.1 UI (User Interface)

UI คือ ส่วนต่อประสานกับผู้ใช้ หรือหน้าจอผู้ใช้ ซึ่งก็คือสิ่งที่ผู้ใช้งานมองเห็นผ่านอุปกรณ์ทำงาน เป็นส่วนที่ควรออกแบบให้สวยงาม ดึงดูดผู้ใช้

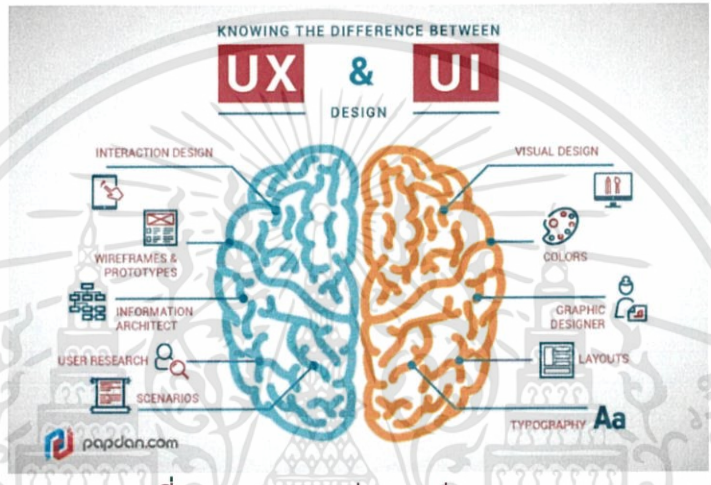


ภาพที่ 2.5 ตัวอย่าง UI ของระบบปฏิบัติการ MacOS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 UX (User eXperience)

UX คือ ประสบการณ์ของผู้ใช้ หรือความพึงพอใจของผู้ใช้ที่มีต่อระบบ ส่วนนี้จะเกี่ยวข้องกับการใช้งานง่าย – ยาก ดังนั้น UX จึงเป็นหัวใจของการออกแบบ ซึ่งความสำคัญของ UX ก็คือการคำนึงถึงผู้ใช้ว่าควรออกแบบเว็บไซต์อย่างไรระบบจึงจะออกมาใช้งานง่าย ตัวอย่างของความสำคัญของ UX ในด้านต่าง ๆ เช่น Visual Design คือการออกแบบหน้าตาของเว็บไซต์, Usability คือการใช้งานง่าย, Interaction Design คือรูปแบบการตอบสนองต่อการใช้งาน, Accessibility คือการเข้าถึงได้ง่าย เป็นต้น



ภาพที่ 2.6 ความแตกต่างระหว่าง UX และ UI

2.1.4 Back-end

Back-end คือ ระบบจัดการเว็บไซต์ หรือที่เรียกว่าหลังบ้าน จะมีหน้าที่ในการจัดการเว็บไซต์ ไม่ว่าจะเป็นในด้านการจัดการฐานข้อมูล การรับส่งข้อมูล โครงสร้างเว็บไซต์ ความปลอดภัยของเว็บไซต์ เป็นต้น ตัวอย่างเฟรมเวิร์คที่ใช้ในการเขียน Back-end เช่น Django, Express.js, Ruby on Rails, Flask เป็นต้น

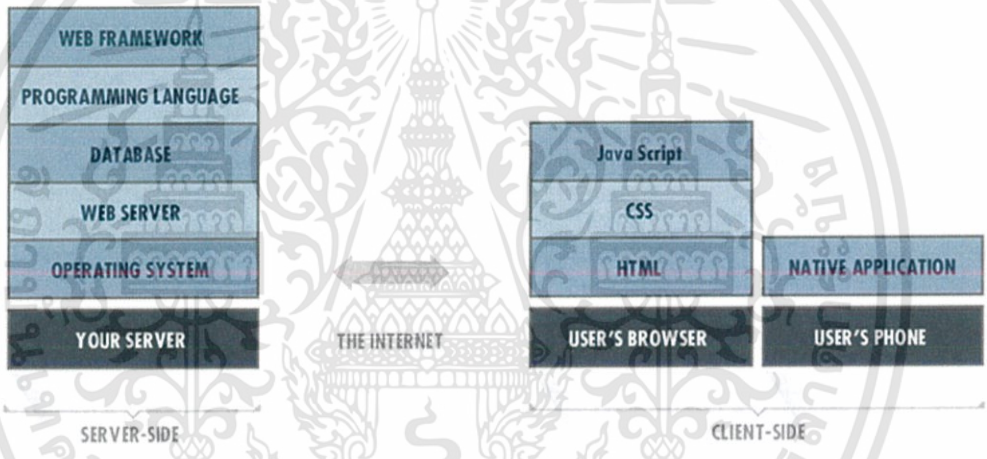
ภาพที่ 2.7 Django

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 Software Stack

Software Stack คือ กลุ่มของซอฟต์แวร์ที่อิสระต่อกัน นำมาทำงานร่วมกันเพื่อพัฒนาเป็นแอปพลิเคชันหรือโปรแกรมหนึ่งขึ้นมา โดยปกติแล้วมักจะมีซอฟต์แวร์ทั้งระดับบน (High-level) ที่ดูแลการจัดการงานต่าง ๆ ของผู้ใช้ และระดับล่าง (Low-level) ที่ติดต่อกับส่วนฮาร์ดแวร์มาทำงานร่วมกัน ช่วยทำให้การพัฒนานั้นใช้จำนวนซอฟต์แวร์น้อยที่สุด แต่ได้ประสิทธิภาพมาก

ตัวอย่าง Software Stack เช่น LAMP Stack ที่ประกอบไปด้วย Linux, Apache, MySQL และ PHP มักจะนำไปใช้ในการพัฒนาเว็บไซต์, MEAN Stack ที่ประกอบไปด้วย MongoDB, Express.js, Angular และ Node.js ที่ใช้สำหรับสร้างไดนามิกเว็บแอปพลิเคชัน (Dynamic Web Application) และเว็บไซต์ต่าง ๆ เป็นต้น จากตัวอย่างจะเห็นว่าแต่ละซอฟต์แวร์ที่นำมารวมกันเป็น Software Stack นั้น ไม่ได้มีส่วนเกี่ยวข้องกัน แต่สามารถนำมาสร้างโปรแกรมร่วมกันได้



ภาพที่ 2.8 ส่วนประกอบของ Software Stack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 MVC (Model-View-Controller)

MVC คือ รูปแบบของการเขียนโปรแกรม (Design Pattern) ชนิดหนึ่ง โดยจะแยกแต่ละส่วนของโปรแกรมให้เป็นออปเจ็กต์ (Object) ต่าง ๆ แบ่งเป็น 3 ส่วน คือ Model, View และ Controller

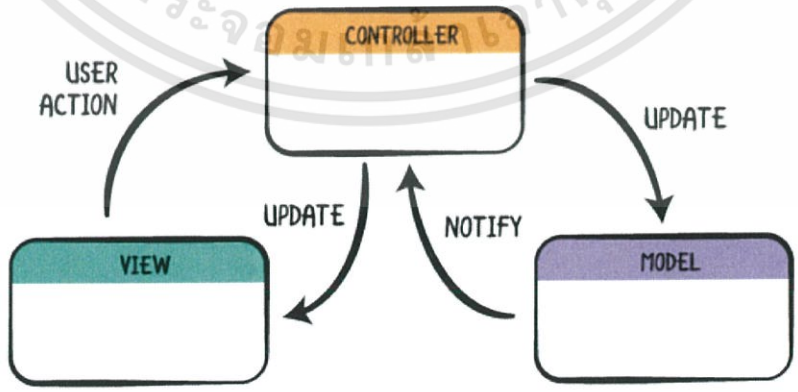
Model เป็นส่วนในการจัดการกับข้อมูลดิบ ดูแลการติดต่อสื่อสารระหว่างออปเจ็กต์กับฐานข้อมูล ตรวจสอบความสัมพันธ์ของข้อมูล (Association) เป็นต้น

View เป็นส่วนแสดงผลผ่านเบราว์เซอร์ ที่ติดต่อกับผู้ใช้ ซึ่งก็คือ User Interface นั่นเอง จะทำงานสัมพันธ์กับ Controller

Controller เป็นส่วนประมวลผลหลักของโปรแกรม จะเป็นส่วนที่ทำงานเป็นอันดับแรกเมื่อโปรแกรมถูกเรียกจากเว็บเบราว์เซอร์ Controller จะติดต่อกับฐานข้อมูลผ่าน Model และแสดงผลข้อมูลผ่าน View

2.1.6.1 การทำงานของ MVC

1. เริ่มจากผู้ใช้ส่งคำขอไปที่เว็บไซต์ ซึ่งจะถูกส่งต่อไปให้ Controller จากนั้น Controller จะทำการตรวจสอบข้อมูลที่ส่งมา
2. Controller จะเรียก Model เพื่อให้มาจัดการกับคำขอนั้น
3. Model จัดการกับคำขอ อาจมีการติดต่อกับฐานข้อมูล เมื่อเสร็จแล้วจะส่งผลลัพธ์กลับไปให้ Controller
4. เมื่อ Controller ได้รับผลลัพธ์จาก Model แล้ว จะส่งต่อให้ไปให้ View เพื่อทำการแสดงผลผ่านเว็บเบราว์เซอร์
5. View สร้างเว็บเพจ แล้วส่งเว็บเพจนี้กลับไปให้แก่ Controller
6. Controller ส่งเว็บเพจที่ได้จาก View กลับไปให้ผู้ใช้ที่ร้องขอ



ภาพที่ 2.9 การทำงานของ MVC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7 CMS (Content Management System)

CMS คือ ระบบที่นำมาใช้ในการสร้างเว็บไซต์แบบสำเร็จรูป โดยที่ผู้ใช้ไม่จำเป็นต้องมีความรู้ด้านการเขียนโปรแกรมแต่อย่างใด มีข้อดีคือผู้ใช้จะสามารถเพิ่ม แก้ไข ลบ เนื้อหา รวมทั้งตกแต่งหน้าตาของเว็บไซต์ได้เอง ไม่เสียเวลาในการพัฒนา ลดค่าใช้จ่าย แต่ข้อเสียคือไม่สามารถปรับแต่งได้ทุกส่วน เนื่องจากเป็นระบบที่สำเร็จรูปที่ถูกสร้างมาแล้ว ตัวอย่างของ CMS เช่น WordPress, Joomla, Drupal เป็นต้น

ภาพที่ 2.10 WordPress

2.1.8 NoSQL Database

NoSQL Database คือ ระบบฐานข้อมูลชนิดหนึ่งที่ไม่มีความสัมพันธ์กันระหว่างข้อมูล ต่างจากระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database) มีข้อดีคือสามารถขยายระบบได้ง่าย ซึ่งเป็นข้อด้อยข้อหลักของระบบฐานข้อมูลเชิงสัมพันธ์ และมีความยืดหยุ่นต่อข้อมูลประเภทต่าง ๆ สูง เพราะถูกออกแบบมาสำหรับข้อมูลที่ไม่มีโครงสร้าง (Unstructured Data)

NoSQL มี 4 ประเภทคือ

1. Key – value store คือ ฐานข้อมูลที่เก็บในรูปแบบ Key และ Value ซึ่งจะมีความสัมพันธ์กัน จะสามารถหา Value ได้จาก Key

2. Wide Column store คือ ฐานข้อมูลที่เก็บข้อมูลในรูปแบบตารางคล้ายกับฐานข้อมูลเชิงสัมพันธ์ แต่จะเก็บอยู่ในแบบคอลัมน์แทนแบบแถว

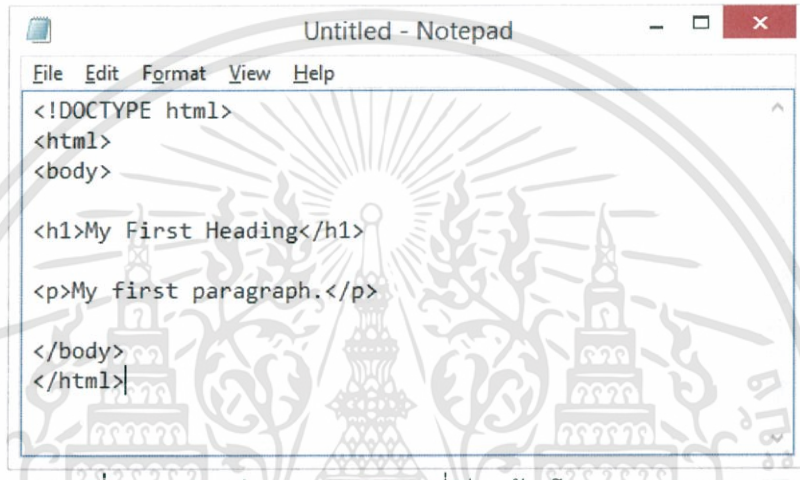
3. Document database คือ ฐานข้อมูลที่เก็บอยู่ในรูปของเอกสาร โดยแต่ละเอกสารสามารถมีจำนวนฟิลด์ไม่เท่ากันได้ ทำให้ง่ายต่อการแก้ไขโครงสร้าง

4. Graph database คือ ฐานข้อมูลที่เก็บข้อมูลในรูปแบบของกราฟที่มีความสัมพันธ์กันทำให้สามารถใช้ทฤษฎีกราฟได้ในการจัดการกับข้อมูลได้เป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.9 ภาษา HTML (Hypertext Markup Language)

ภาษา HTML คือ ภาษาคอมพิวเตอร์ที่ใช้ในการเขียนเว็บเพจ ถูกพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) การเขียนภาษา HTML สามารถเขียนโดยใช้โปรแกรมเขียนเอกสาร (Text Editor) ต่าง ๆ เช่น Notepad, EditPlus เป็นต้น การทดสอบผลการเขียนภาษา HTML นั้น สามารถทำได้โดยการเปิดไฟล์ HTML ด้วยเว็บเบราว์เซอร์ใด ๆ ก็ได้

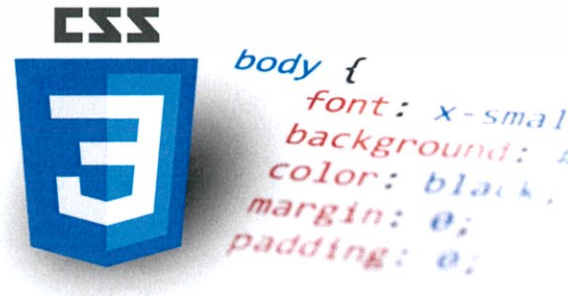


ภาพที่ 2.11 ตัวอย่างภาษา HTML ที่เขียนด้วยโปรแกรม Notepad

2.1.10 ภาษา CSS (Cascading Style Sheet)

ภาษา CSS คือ ภาษาที่ใช้ตกแต่งเอกสาร HTML/XHTML ให้มีลักษณะตามที่ต้องการเพิ่มเติมจากภาษา HTML ที่ทำไม่ได้ โดยทีภาษา CSS สามารถระบุรูปแบบของเนื้อหาในเอกสารได้หลายรูปแบบ เช่น สีของตัวอักษร สีพื้นหลัง การจัดวางข้อความ ประเภทตัวอักษร เป็นต้น

ประโยชน์ของภาษา CSS คือ มีคุณสมบัติในการตกแต่งมากกว่าภาษา HTML อีกทั้งยังสามารถกำหนดไว้ที่เอกสาร HTML ที่ตำแหน่งใด ๆ ก็ได้ จะมีผลกับทั้งเอกสาร ทำให้การแก้ไขเป็นไปได้อย่างสะดวก หรือจะแยกไฟล์ภาษา CSS และนำมาใช้กับไฟล์ภาษา HTML ก็ได้



ภาพที่ 2.12 ตัวอย่างภาษา CSS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.11 ภาษา Javascript

ภาษา Javascript คือ ภาษาสำหรับเขียนโปรแกรมบนอินเทอร์เน็ต ใช้ในการพัฒนาเว็บไซต์ร่วมกับภาษา HTML ทำให้เว็บไซต์สามารถตอบสนองกับผู้ใช้ได้มากขึ้น สามารถเขียนหรือเปลี่ยนแปลงส่วนประกอบต่าง ๆ ใน HTML ได้ (HTML Element) และยังสามารถใช้ตรวจสอบข้อมูล ต่าง ๆ ที่รับมาจากผู้ใช้ได้อีกด้วย

ภาษา Javascript เป็นภาษาประเภท Interpreted Language คือ ไม่ต้องมีการ Compile ก่อน หมายถึง คอมพิวเตอร์จะแปลและทำงานตามคำสั่งแบบทีละบรรทัด และเป็นภาษาประเภท Client-side Script ซึ่งจะประมวลผลบนเครื่องคอมพิวเตอร์ของผู้เยี่ยมชมเว็บไซต์ โดยใช้โปรแกรมเว็บเบราเซอร์ต่าง ๆ การที่ภาษา Javascript ไม่ได้ถูกประมวลผลบนเครื่องเว็บเซิร์ฟเวอร์จะช่วยแบ่งเบาภาระการทำงานของเซิร์ฟเวอร์ และทำให้ทำงานได้อย่างรวดเร็ว

```
function setLocation(city) {
  var country = "France";

  function printLocation() {
    console.log("You are in " + city + ", " + country);
  }

  return printLocation;
}

var currentLocation = setLocation ("Paris");

currentLocation();   // output: You are in Paris, France
```

ภาพที่ 2.13 ตัวอย่างภาษา Javascript

2.1.12 ภาษา Typescript

ภาษา Typescript คือ ภาษาคอมพิวเตอร์ที่เป็นเซตใหญ่ของภาษา Javascript ถูกออกแบบมาเพื่อพัฒนาโปรแกรมขนาดใหญ่ สามารถใช้พัฒนา Javascript Application ได้ทั้งส่วนผู้ใช้ (Client-side) และส่วนเซิร์ฟเวอร์ (Server-side)

ข้อดีของการใช้ภาษา Typescript คือ มีความสามารถมากกว่า Javascript เพราะได้รวมความสามารถต่าง ๆ ของ EcmaScript 2016 (ES6) เอาไว้ ในส่วนของการเขียนโปรแกรม จะมีข้อผิดพลาดด้านชนิดของข้อมูลลดลง เพราะตัวแปรที่ประกาศชนิดในภาษา Typescript จะไม่สามารถเปลี่ยนแปลงชนิดได้อีก และภาษา Typescript จะมีการตรวจสอบความผิดพลาดของโค้ดในช่วง Compile Time ซึ่งหากเกิดข้อผิดพลาดขึ้น ข้อผิดพลาดนั้นจะไม่ปรากฏขึ้นขณะที่ใช้งาน (Runtime)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

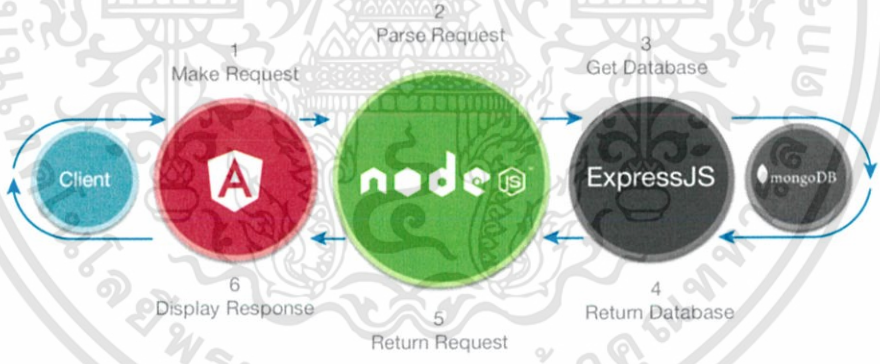
บทที่ 3
วิธีดำเนินการวิจัย

การดำเนินการครั้งนี้ มุ่งที่จะปรับปรุงศูนย์ช่วยเหลือของโปรแกรม Eikon ให้ใช้งานง่าย สะดวก และเข้าถึงข้อมูลได้รวดเร็วมากขึ้นและสามารถแก้ไขหรือเพิ่มเติมข้อมูลได้ง่ายขึ้น โดยใช้เวลาในการ Deploy น้อยลง ซึ่งผู้จัดทำได้ดำเนินการทั้งในส่วนของสถาปัตยกรรมของระบบและการออกแบบระบบดังนี้

3.1 สถาปัตยกรรมของระบบ

โครงงานนี้มีลักษณะเป็นเว็บแอปพลิเคชัน โดยส่วนหลัก ๆ ประกอบไปด้วย หน้าเว็บเพจ เพื่อให้ผู้ใช้เข้ามาใช้บริการ, เซิร์ฟเวอร์ สำหรับเป็นเครื่องให้บริการ และฐานข้อมูลเพื่อเก็บข้อมูลต่าง ๆ ของตัวเว็บแอปพลิเคชัน ซึ่งแต่ละส่วนจะมีการจัดการที่แตกต่างกัน แต่ทั้งหมดนี้ต้องนำมาทำงานร่วมกันได้ จึงเลือก MEAN Stack เข้ามาใช้ในการพัฒนา

MEAN Stack เป็น Javascript software stack ที่ใช้ในการสร้างเว็บแอปพลิเคชันตั้งแต่ระดับบนสุดที่เป็นหน้าเว็บเพจ ไปจนถึงระดับล่างสุดที่เป็นฐานข้อมูล โดย MEAN นั้นย่อมาจาก MongoDB, Express.js, Angular และ Node.js



ภาพที่ 3.1 สถาปัตยกรรมของ MEAN Stack

การทำงานของ MEAN Stack คือเมื่อฝั่งผู้ใช้งานต้องการข้อมูลบางอย่าง Angular ก็จะจัดการประมวลผลสร้างคำขอและส่งคำขอนั้นไปยัง Node.js ที่เป็นเซิร์ฟเวอร์ จากนั้น Node.js จะส่งต่อไปยัง Express.js เพื่อให้ Express.js ขอข้อมูลที่ต้องการจาก MongoDB ที่เป็นฐานข้อมูล เมื่อ MongoDB รับคำขอแล้ว ก็จะทำการประมวลผลหาข้อมูลที่ต้องการ และส่งกลับไปยัง Express.js ซึ่งข้อมูลจะถูกส่งกลับไปต่อยัง Node.js และส่งให้ Angular แสดงผลในที่สุด โดยรายละเอียดของแต่ละองค์ประกอบของ MEAN Stack มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 MongoDB

MongoDB เป็นฐานข้อมูลชนิดหนึ่ง จัดอยู่ในประเภท NoSQL มีการเก็บข้อมูลในลักษณะ JSON (Javascript Object Notation) คือมีการเก็บข้อมูลเป็นแบบ Key และ Value ข้อดีคือสามารถรองรับข้อมูลขนาดใหญ่ได้ มีความยืดหยุ่นสูง เหมาะสำหรับข้อมูลที่กึ่งมีโครงสร้างและไม่มีโครงสร้าง สามารถเข้าถึงข้อมูลได้อย่างรวดเร็ว จึงทำให้ได้เปรียบกว่าระบบฐานข้อมูลเชิงสัมพันธ์ (Relational Database)

ภาพที่ 3.2 MongoDB


```
{
  "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",
  "grades": [
    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```


ภาพที่ 3.3 การเก็บข้อมูลในรูปแบบ JSON (Javascript Object Notation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

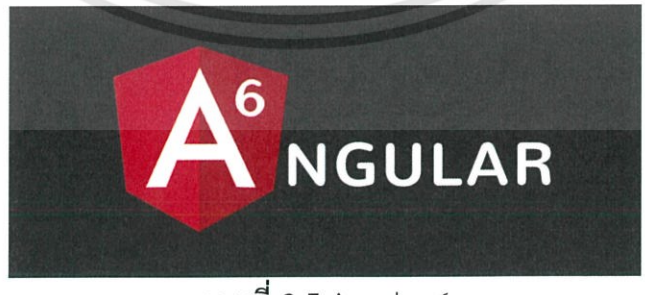
3.1.2 Express.js

Express.js เป็นเว็บแอปพลิเคชันเฟรมเวิร์ค ที่ทำงานอยู่บน Node.js ทำหน้าที่เป็นเซิร์ฟเวอร์ตัวหนึ่ง มีคุณสมบัติช่วยจัดการ Request และ Response ของเว็บไซต์ รวมไปถึงการจัดการเส้นทางของระบบ (Routing) และการรับส่งข้อมูลของระบบ (Middleware) ต่าง ๆ ช่วยอำนวยความสะดวกสบายให้แก่การจัดการเซิร์ฟเวอร์ที่จากเดิมนั้นต้องใช้ความรู้เกี่ยวกับภาษาเซิร์ฟเวอร์ เช่น ภาษา PHP แต่หากใช้ Express.js เพียงแค่มีความรู้ของภาษา Javascript ก็สามารถจัดการกับเซิร์ฟเวอร์ได้ ช่วยลดความยุ่งยากลงได้มาก

ภาพที่ 3.4 Express.js

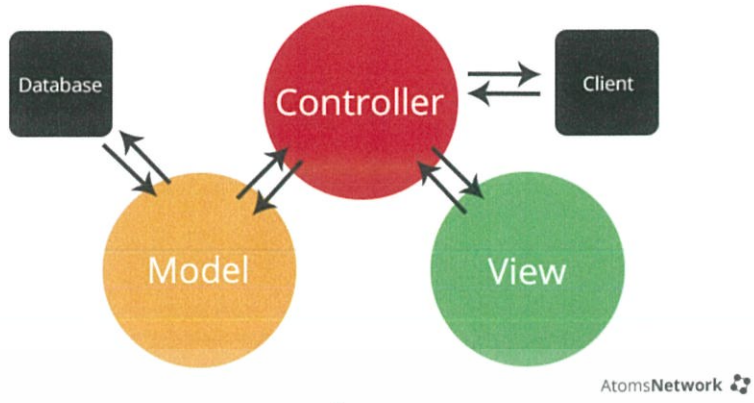
3.1.3 Angular

Angular คือ Front-end Framework สำหรับทำเว็บแอปพลิเคชัน มีการทำงานบน Client Side คือจะมีการประมวลผลทางฝั่งเบราว์เซอร์ ทำให้มีการทำงานที่รวดเร็ว Angular ถูกพัฒนาโดยบริษัท Google ออกแบบมาเพื่อสร้างเว็บไซต์แบบ SPA (Single Page Application) หรือเว็บแอปพลิเคชันที่มีเพียงหน้าเดียว มีโครงสร้างเป็นแบบ MVC โดย Model จะทำงานเกี่ยวกับฐานข้อมูล และจัดการด้าน Business Logic ส่วน View จะเป็นส่วนที่ทำหน้าที่แสดงผลหน้าตาของเว็บไซต์ และ Controller จะเป็นส่วนที่จัดการระหว่าง Model และ View รวมไปถึงการรับอินพุตและการจัดการความสัมพันธ์ของออปเจ็กต์ต่าง ๆ ใน Model หากเปรียบการทำงานของ MVC เข้ากับของ Angular จะพบว่า Model คือ Service, View คือ Component template และ Controller คือ Component Class โดยเวอร์ชันที่ใช้งานในโครงงานนี้จะเป็น Angular เวอร์ชัน 6



ภาพที่ 3.5 Angular 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.6 MVC

3.1.4 Node.js

Node.js เป็น Cross Platform Runtime Environment ที่ทำงานบน Server Side พัฒนาโดยภาษา Javascript ทำหน้าที่อยู่ในฝั่ง Back-end สามารถทำงานได้ทุกระบบปฏิบัติการ โดยในที่นี้นำมาใช้เป็นเว็บเซิร์ฟเวอร์ การทำงานของ Node.js จะเป็นแบบ Asynchronous และ Event Driven คือทำงานแบบไม่ต้องรอให้งานอื่น ๆ ทำหน้าที่เสร็จก่อนจึงประมวลผล แต่สามารถทำงานได้อย่างทันที ทำให้การเรียกใช้งาน API ทำได้อย่างสะดวก

ภาพที่ 3.7 Node.js

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

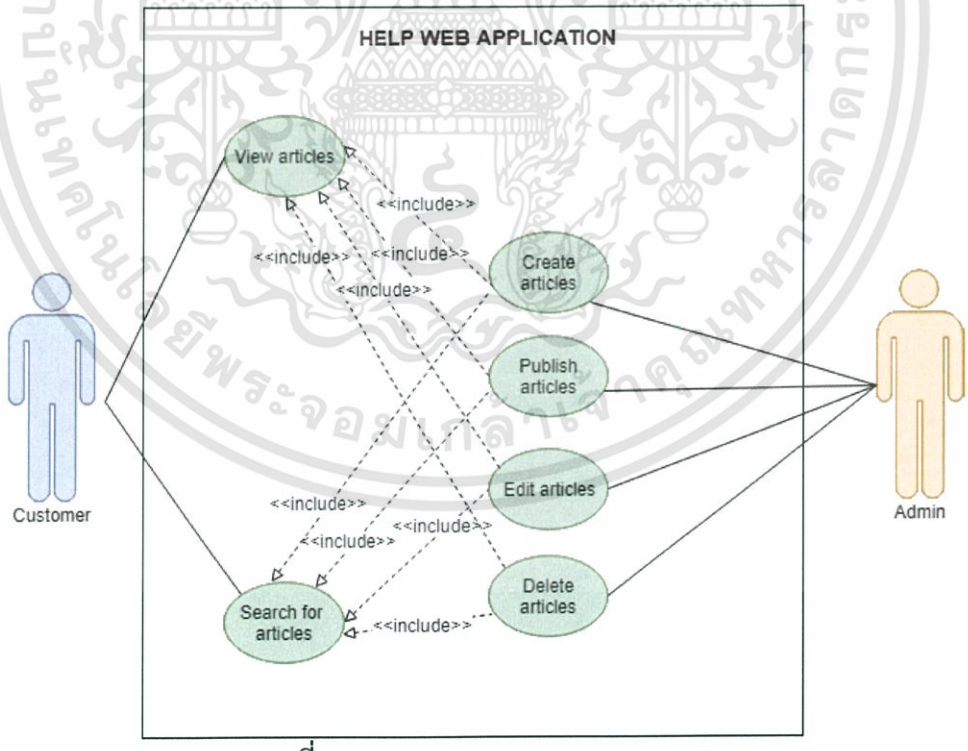
3.1.5 TinyMCE

TinyMCE เป็น WYSIWYG Editor (What You See Is What You Get Editor) หรือ HTML Editor เป็น Open source ที่ใช้สร้าง แก้ไข เนื้อหาของข้อมูล มักนำไปใช้กับเว็บไซต์ที่มีการจัดการกับข้อมูล เช่น CMS (Content Management Systems), LMS (Learning Management Systems) เป็นต้น โดยสาเหตุที่เลือกใช้เพราะเป็น WYSIWYG Editor ที่มีความเป็นมาตรฐาน มีความสามารถพื้นฐานเกี่ยวกับการแก้ไขข้อมูลของบทความ อีกทั้งยังมี Javascript API ที่ใช้งานง่าย ทำให้สามารถเพิ่มฟีเจอร์ (Feature) และกำหนดปลั๊กอิน (Plugin) ต่าง ๆ ได้ตามต้องการ และมีเอกสาร (Documentation) ที่ละเอียดและอธิบายได้ดี

ภาพที่ 3.8 TinyMCE

3.2 การออกแบบ

3.2.1 Use Case Diagram



ภาพที่ 3.9 Use case diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Requirement ของ Project Manger สามารถนำมาเขียนเป็น Use case diagram ได้ดังภาพที่ 3.9 โดยในส่วนของลูกค้า มี Use case ดังนี้

1. View articles คือ ลูกค้าสามารถเข้าชมบทความได้
2. Search for articles คือ ลูกค้าสามารถค้นหาบทความที่ต้องการได้

ส่วนของผู้ดูแลระบบจะมี Use case เหมือนกับส่วนของลูกค้า โดยจะมี Use case เพิ่มมา ดังนี้

1. Create articles คือ ผู้ดูแลระบบสามารถสร้างบทความได้
2. Publish articles คือ ผู้ดูแลระบบสามารถตีพิมพ์บทความได้
3. Edit articles คือ ผู้ดูแลระบบสามารถทำการแก้ไขบทความได้
4. Delete articles คือ ผู้ดูแลระบบสามารถทำการลบบทความได้

ตารางที่ 3.1 Use case: View articles

Use case	View articles
Primary Actor	Customer, Admin
Brief Description	ลูกค้าหรือผู้ดูแลระบบสามารถเข้าชมบทความที่ต้องการได้
Related Use Case	Search for articles
Main Success Scenario	ลูกค้าหรือผู้ดูแลระบบพบบทความที่สนใจ สามารถกดเพื่อเข้าไปชมบทความนั้นได้
Extensions	-

ตารางที่ 3.2 Use case: Search for articles

Use case	Search for articles
Primary Actor	Customer, Admin
Brief Description	ลูกค้าหรือผู้ดูแลระบบสามารถค้นหาบทความที่ต้องการได้
Related Use Case	View articles
Main Success Scenario	1. ลูกค้าหรือผู้ดูแลระบบพิมพ์ข้อความที่ต้องการค้นหาในช่องค้นหา 2. รายการของบทความที่เกี่ยวข้องกับคำที่พิมพ์ค้นหาปรากฏขึ้น 3. ลูกค้าหรือผู้ดูแลระบบเลือกชมบทความที่ต้องการได้
Extensions	1. ลูกค้าหรือผู้ดูแลระบบพิมพ์ข้อความที่ต้องการค้นหาในช่องค้นหา 2. ไม่มีบทความใดที่มีความเกี่ยวข้องกับคำที่พิมพ์ค้นหา 3. ปรากฏข้อความเพื่อแจ้งว่าไม่พบบทความที่เกี่ยวข้อง ให้ลูกค้าหรือผู้ดูแลระบบพิมพ์คำใหม่ หรือติดต่อศูนย์บริการลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 Use case: Create articles

Use case	Create articles
Primary Actor	Admin
Brief Description	ผู้ดูแลระบบสามารถสร้างบทความได้
Related Use Case	Publish articles
Main Success Scenario	1. ผู้ดูแลระบบเลือกเมนู Create new articles 2. ผู้ดูแลระบบทำการใส่ข้อมูลต่าง ๆ ที่เกี่ยวข้องกับบทความ และทำการเขียนเนื้อหาของบทความ 3. ผู้ดูแลระบบกดปุ่ม Draft เพื่อทำให้เป็นบทความร่างในกรณีที่ยังไม่ต้องการตีพิมพ์ หรือกดปุ่ม Publish เพื่อตีพิมพ์บทความ
Extensions	1. ผู้ดูแลระบบไม่ได้ใส่ชื่อบทความ 2. ระบบขึ้นแจ้งเตือนให้กรอกชื่อบทความ

ตารางที่ 3.4 Use case: Publish articles

Use case	Publish articles
Primary Actor	Admin
Brief Description	ผู้ดูแลระบบสามารถตีพิมพ์บทความได้
Related Use Case	Create articles
Main Success Scenario	1. ผู้ดูแลระบบสร้างบทความ และทำการพิมพ์เนื้อหาของบทความ 2. ผู้ดูแลระบบกดปุ่ม Publish เพื่อทำการตีพิมพ์บทความ
Extensions	1. ผู้ดูแลระบบกดปุ่ม Draft เพื่อทำให้เป็นบทความร่าง ในกรณีที่ยังไม่ต้องการตีพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 Use case: Edit articles

Use case	Edit articles
Primary Actor	Admin
Brief Description	ผู้ดูแลระบบสามารถแก้ไขบทความได้
Related Use Case	Publish articles
Main Success Scenario	1. ผู้ดูแลระบบเลือกบทความที่ต้องการแก้ไข 2. ผู้ดูแลกดปุ่ม Edit เพื่อทำการแก้ไข 3. ผู้ดูแลระบบทำการแก้ไขบทความ 4. ผู้ดูแลระบบกดปุ่ม Publish เพื่อทำการตีพิมพ์บทความ
Extensions	1. ผู้ดูแลระบบกดปุ่ม Draft เพื่อทำให้เป็นบทความร่าง ในกรณีที่ยังไม่ต้องการตีพิมพ์

ตารางที่ 3.6 Use case: Delete articles

Use case	Delete articles
Primary Actor	Admin
Brief Description	ผู้ดูแลระบบสามารถลบบทความได้
Related Use Case	-
Main Success Scenario	1. ผู้ดูแลระบบเลือกบทความที่ต้องการลบ 2. ผู้ดูแลระบบกดปุ่ม Delete เพื่อทำการลบบทความ 3. ผู้ดูแลระบบกดปุ่มยืนยันการลบบทความ
Extensions	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Database Schema

โครงสร้างของระบบฐานข้อมูลเป็นไปตามภาพที่ 3.10 เนื่องจากเป็นระบบฐานข้อมูลแบบ NoSQL จึงไม่มี ER Diagram โดยโครงสร้างจะแบ่งออกเป็น Collection ซึ่งเปรียบเสมือนตาราง (Table) ในระบบฐานข้อมูลเชิงสัมพันธ์ และแต่ละ Collection จะมีหลาย Field ซึ่งเปรียบเสมือนคอลัมน์ (Column) นั่นเอง โดยในระบบนี้จะมี 3 Collection ด้วยกัน

Collection Article จะเก็บข้อมูลของบทความทั้งหมด โดยจะมี 11 ฟิลด์ ประกอบไปด้วย

- id: เลขประจำของแต่ละบทความ
- name: ชื่อของบทความ
- author: เจ้าของบทความ
- date_published: วันที่สร้างบทความ
- version: เวอร์ชันของบทความ
- dev_team: ทีม Development ของบทความนั้น ๆ
- status: สถานะของบทความ ประกอบไปด้วย published ที่หมายถึงทำการตีพิมพ์แล้ว และ draft คือบทความร่าง
- last_modified: วันที่ที่ทำการแก้ไขบทความล่าสุด
- content: เนื้อหาของบทความ
- views: ยอดเข้าชมบทความ
- tag: แท็กต่าง ๆ ของบทความนั้น ๆ โดยจะเก็บแบบ one-to-many ที่เป็นแบบ child referencing กับ Collection Tag

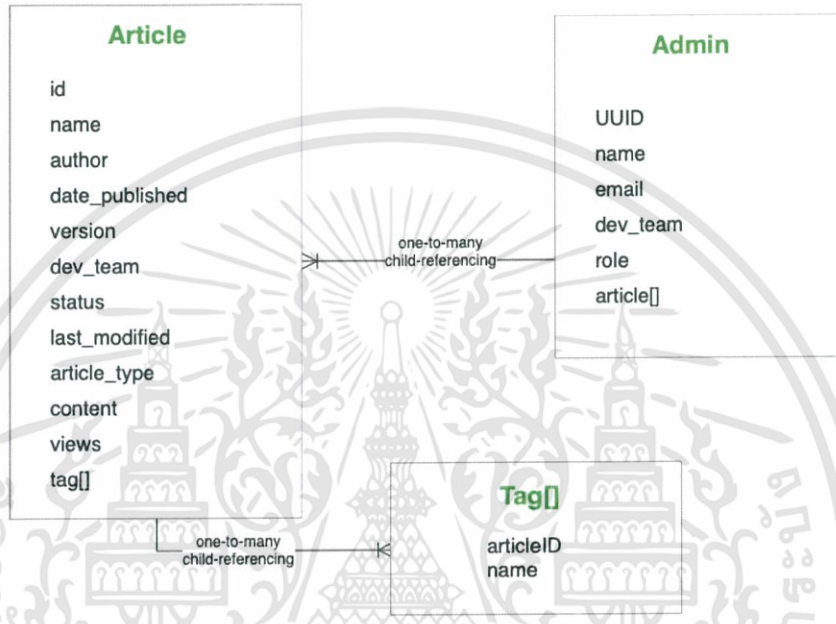
Collection Tag จะเก็บแท็กทั้งหมดที่ปรากฏอยู่ในบทความ โดยจะมี 2 ฟิลด์ ด้วยกัน ประกอบไปด้วย

- id: เลขประจำของแต่ละแท็ก
- name: ชื่อของแท็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Collection Admin จะเก็บข้อมูลของผู้ดูแลระบบแต่ละคน ซึ่งในที่นี้เป็นเพียงแค่การออกแบบเท่านั้น เพราะไม่สามารถนำไปใช้จริงได้ เนื่องด้วยขอบเขตของโครงงานที่ไม่สามารถเข้าไปจัดการข้อมูลเฉพาะของทางบริษัทได้

- id: เลขประจำของแต่ละแท็ก
- name: ชื่อของแท็ก

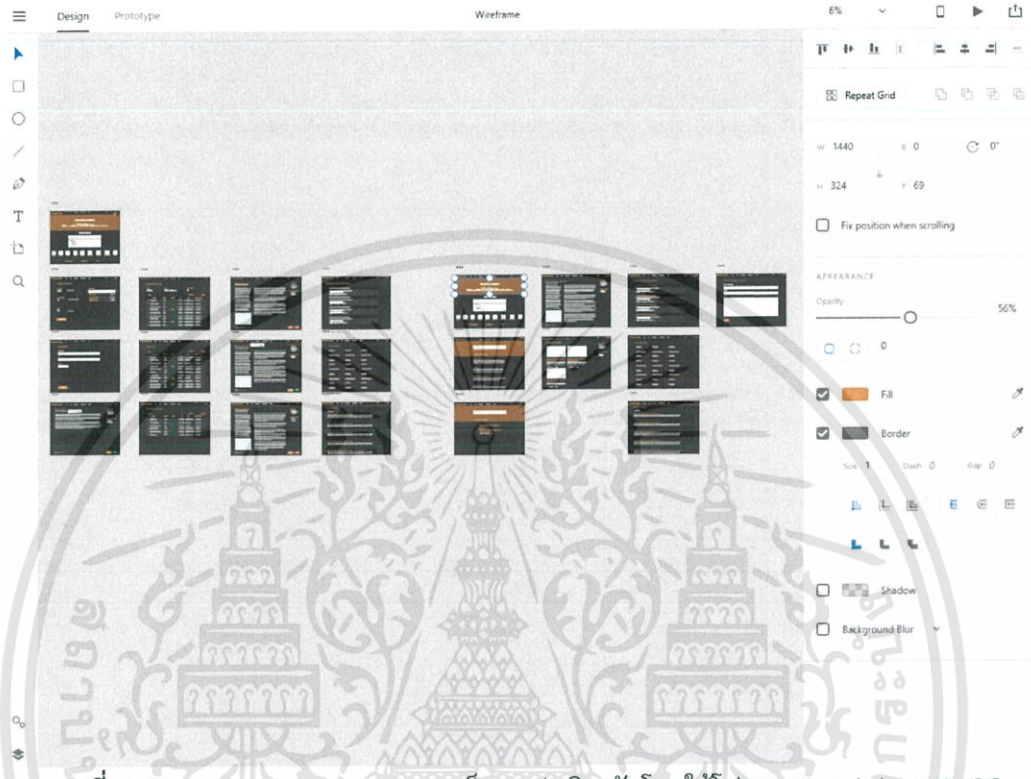


ภาพที่ 3.10 Database Schema ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบบจอภาพของเว็บแอปพลิเคชัน (Visual Design)

การออกแบบหน้าตาของเว็บแอปพลิเคชันนี้ ใช้โปรแกรม Adobe XD CC มาช่วยในการทำ โดยเริ่มจากการวางโครงร่าง (Wireframe) ของเว็บแอปพลิเคชัน และออกแบบตามหลักการของ UI/UX



ภาพที่ 3.11 การออกแบบจอภาพของเว็บแอปพลิเคชันโดยใช้โปรแกรม Adobe XD CC

3.2.4 การเขียนตารางกรณีที่ใช้ในการทดสอบ (Test case)

ออกแบบตาราง Test case เพื่อนำมาใช้ในการทดสอบระบบหลังจากที่ได้ทำการเขียนโปรแกรมเรียบร้อยแล้ว เพื่อเป็นการทำให้แน่ใจว่าระบบมีคุณสมบัติตาม Requirement ที่ต้องการ ทำงานได้ถูกต้อง และไม่มีข้อผิดพลาดในการใช้งาน โดยผลการทดสอบต้องผ่านในทุกกรณี หากมีกรณีใดที่ไม่ผ่าน จะต้องทำการแก้ไขส่วนนั้น

กรณีทดสอบใน Test case จะมีจำนวนมาก เพราะต้องทดลองกับสถานการณ์จำลองทั้งหมดที่เป็นไปได้ ทั้งกรณีที่ทำรายการถูกต้อง และกรณีที่ทำรายการไม่ถูกต้อง เพื่อทดสอบการทำงานของระบบว่าสามารถรองรับข้อผิดพลาดได้หรือไม่ โดย Test case ที่ได้ออกแบบมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 ตาราง Test case

Test case ID	Test scenario	Test steps	Test data	Expected results	Actual results	Pass/Fail
1	Customer can view articles	1. Go to home page 2. Click any app icon 3. Click an article from the list	-	The article's content appears		
2	Customer can search for articles	1. Go to home page 2. Type a keyword in search box 3. Press enter	Keyword = login	The articles that contain the keyword appear		
		1. Go to any page 2. Type a keyword in search box at navigation bar 3. Press enter	Keyword = login	The articles that contain the keyword appear		
		1. Go to home page 2. Type any word at search box 3. Press enter 4. Type a keyword in search box at the top of the page	Keyword = login	The articles that contain the keyword appear		
3	Admin can view article	1. Go to home page 2. Click any app icon 3. Click an article from the list	-	The article's content appears		
4	Admin can search for articles	1. Go to home page 2. Type a keyword in search box 3. Press enter	Keyword = login	The articles that contain the keyword appear		
		1. Go to any page 2. Type a keyword in search box at navigation bar 3. Press enter	Keyword = login	The articles that contain the keyword appear		
		1. Go to home page 2. Type a keyword at search box 3. Press enter 4. Type a keyword in search box at the top of the page	Keyword = login	The articles that contain the keyword appear		
		1. Go to admin portal 2. Type a keyword at search box at all article section	Keyword = getting	The article's names that contain the keyword appear		
		1. Go to admin profile 2. Type a keyword at search box at the top of all article list section	Keyword = getting	The article's names that contain the keyword appear		
5	Admin can create articles	1. Go to admin portal 2. Click 'New article' button 3. Enter article name 4. Enter article tag 5. Enter article version 6. Click 'Create' button	Article name = test Article tag = tag Article version = 1.0	Can create the article		
		1. Go to admin portal 2. Click 'New article' button 3. Do not enter article name	Article tag = tag Article version = 1.0	Can not the create article, the warning message appears		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		4. Enter article tag 5. Enter article version 6. Click 'create' button				
		1. Go to admin portal 2. Click 'New article' button 3. Enter article name 4. Do not enter article tag 5. Enter article version 6. Click 'create' button	Article name = test Article version = 1.0	Can create the article		
		1. Go to admin portal 2. Click 'New article' button 3. Enter article name 4. Enter article tag 5. Do not enter article version 6. Click 'create' button	Article name = test Article tag = tag	Can create the article		
		1. Go to admin portal 2. Click 'New article' button 3. Do not enter article name 4. Do not enter article tag 5. Enter article version 6. Click 'create' button	Article version = 1.0	Can not the create article, the warning message appears		
		1. Go to admin portal 2. Click 'New article' button 3. Enter article name 4. Do not enter article tag 5. Do not enter article version 6. Click 'create' button	Article name = test	Can create the article		
		1. Go to admin portal 2. Click 'New article' button 3. Do not enter article name 4. Enter article tag 5. Do not enter article version 6. Click 'create' button	Article tag = tag	Can not the create article, the warning message appears		
		1. Go to admin portal 2. Click 'New article' button 3. Do not enter article name 4. Do not enter article tag 5. Do not enter article version 6. Click 'create' button	-	Can not the create article, the warning message appears		
6	Admin can publish articles	1. Go to admin portal 2. Click 'New article' button 3. Enter article name 4. Enter article tag 5. Enter article version 6. Click 'Create' button 7. Type any content 8. Click 'publish' button	Article name = test Article tag = tag Article version = 1.0 Content = test content	Can publish the article		
7	Admin can edit articles	1. Go to any article 2. Click 'Edit' button 3. Type new content	New content = test new content	Can edit the article		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		4. Type new tag 5. Type new version 6. Press 'Publish' button	New tag = new New version = 2.0			
8	Admin can delete article	1. Go to any article 2. Click 'Delete' button 3. Click 'Yes' button at the modal	-	Can delete the article		
		1. Go to admin profile 2. Check any one checkbox in front of an article's name 3. Click 'DELETE button	-	Can delete the article		
		1. Go to admin profile 2. Check any three checkboxes in front of an article's name 3. Click 'DELETE button	-	Can delete the article		

3.2 ขั้นตอนการดำเนินงาน

ตารางที่ 3.8 ตารางแสดงขั้นตอนการดำเนินงาน

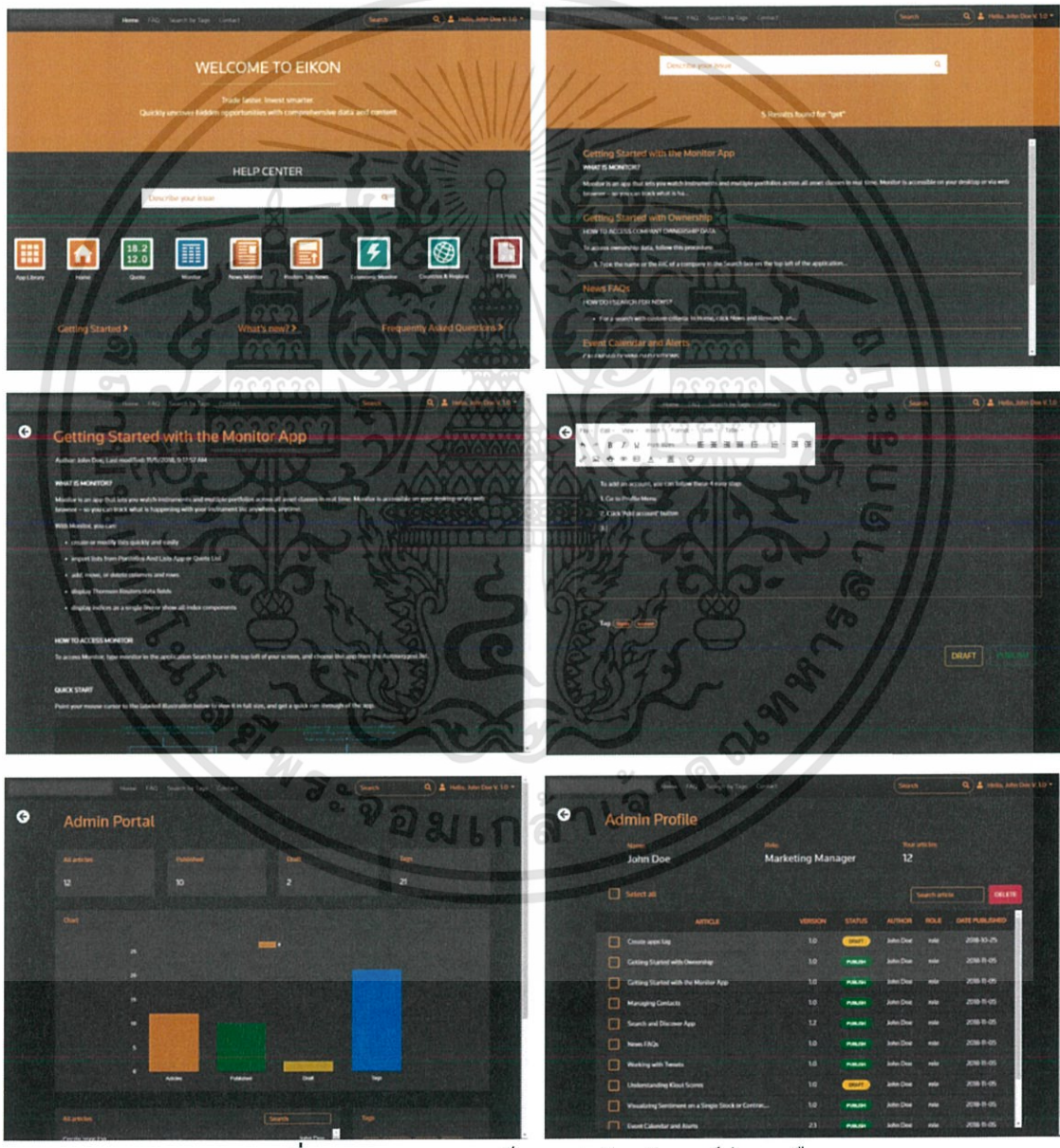
	กันยายน				ตุลาคม				พฤศจิกายน				ธันวาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
วางแผน วิเคราะห์ระบบ																
ออกแบบ																
พัฒนาและทดสอบซอฟต์แวร์																
จัดทำเอกสาร																
สอบโครงงานสหกิจ																

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการวิจัย

การจัดทำโครงงานครั้งนี้ มีวัตถุประสงค์ที่จะปรับปรุงศูนย์ช่วยเหลือของโปรแกรม Eikon ให้ใช้งานง่าย สะดวก และเข้าถึงข้อมูลได้รวดเร็วมากขึ้นและสามารถแก้ไขหรือเพิ่มเติมข้อมูลได้ง่ายขึ้น โดยใช้เวลาในการ Deploy น้อยลง ผู้จัดทำได้นำเสนอผลการดำเนินการ ดังนี้

4.1 ภาพรวมของระบบ



ภาพที่ 4.1 ภาพรวมของเว็บแอปพลิเคชันศูนย์ช่วยเหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการพัฒนาระบบ

ตารางที่ 4.1 ตารางผลการพัฒนาระบบ

ที่คาดว่าจะพัฒนา	ผลการพัฒนา	
	สำเร็จ	ไม่สำเร็จ
สร้างบทความ	/	
แก้ไขบทความ	/	
ลบบทความ	/	
บทความร่าง	/	
เพิ่ม Tag แต่ละบทความ	/	
เข้าชมบทความ	/	
ค้นหาบทความ	/	
ค้นหาบทความโดย Tag	/	
คำถามที่พบบ่อย	/	
ติดต่อศูนย์บริการลูกค้า	/	
หน้าจัดการของผู้ดูแลระบบ	/	
หน้าโปรไฟล์ของผู้ดูแลระบบ	/	
บทความแนะนำ		/
บทความที่เกี่ยวข้อง		/
ให้คะแนนบทความ		/
มีเทมเพลตของบทความ		/

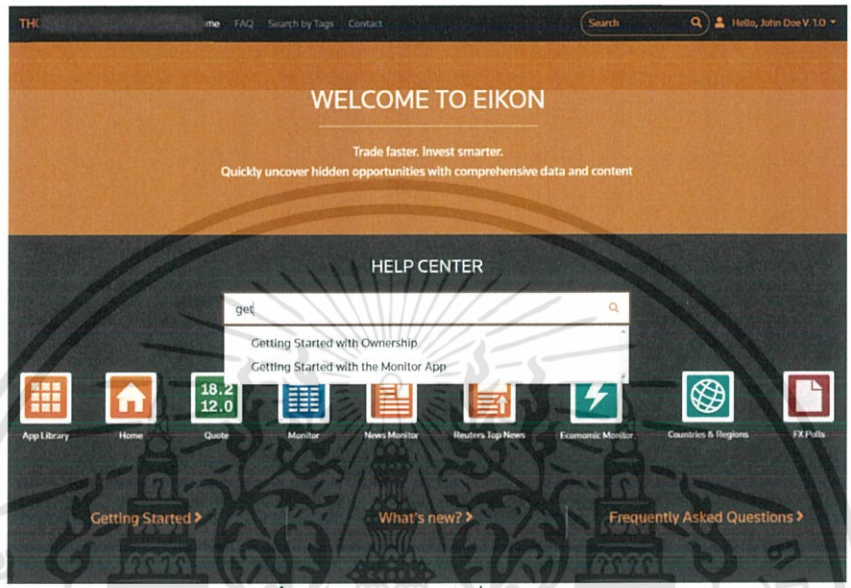
ในขั้นตอนการออกแบบการจัดทำโครงงาน ผู้จัดทำได้กำหนดหลักที่คาดว่าจะทำทั้งหมด 16 ซึ่งมีที่ดำเนินการได้สำเร็จทั้งหมด 12 และที่ไม่ได้ดำเนินการทั้งหมด 4 ด้วยกัน คิดเป็นร้อยละ 75 ที่จัดทำสำเร็จ และร้อยละ 25 ที่ไม่ได้จัดทำ เนื่องจากขณะที่ดำเนินการพัฒนาระบบผู้จัดทำได้รับคำแนะนำว่าเป็นที่ไม่มีความจำเป็นและไม่มีความเหมาะสมกับระบบจึงไม่ได้ดำเนินการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ขั้นตอนการใช้งาน

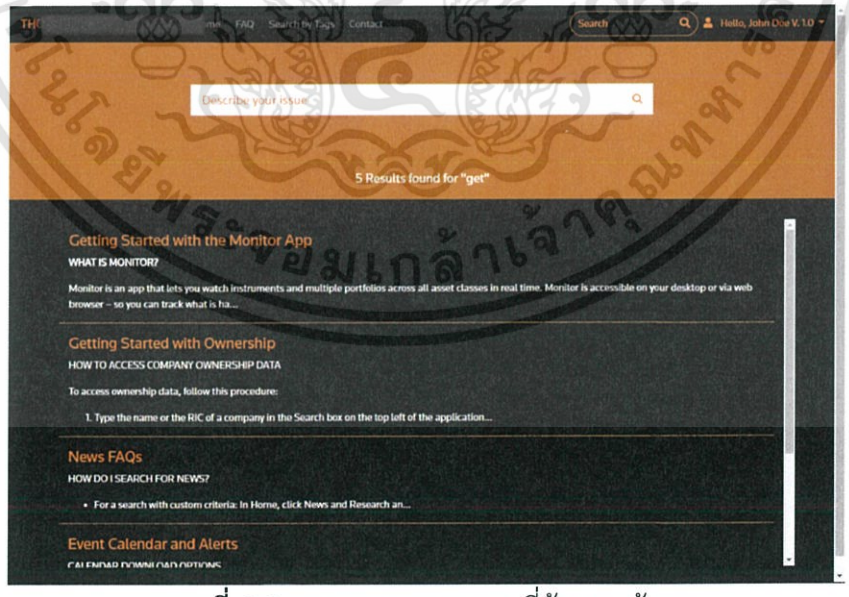
4.3.1 การค้นหาเข้าชมบทความ

1. พิมพ์ข้อมูลที่ต้องการหาในช่องค้นหาที่หน้าแรกของศูนย์ช่วยเหลือ จากนั้นกด Enter ที่แป้นพิมพ์ หรือคลิกที่สัญลักษณ์รูปแว่นขยายทางด้านขวาของช่องค้นหา



ภาพที่ 4.2 พิมพ์ข้อมูลที่ต้องการค้นหา

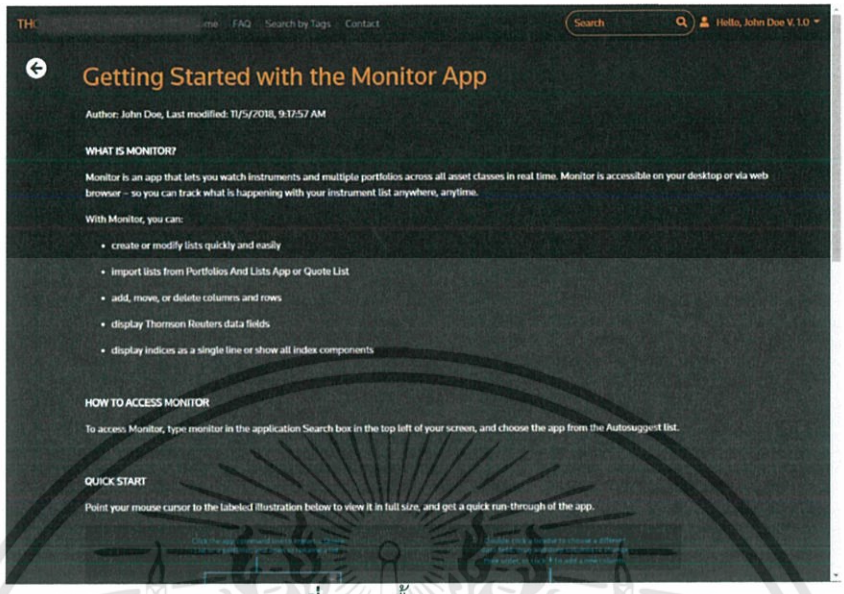
2. รายการของบทความที่เกี่ยวข้องกับข้อมูลที่ต้องการค้นหาจะปรากฏขึ้น



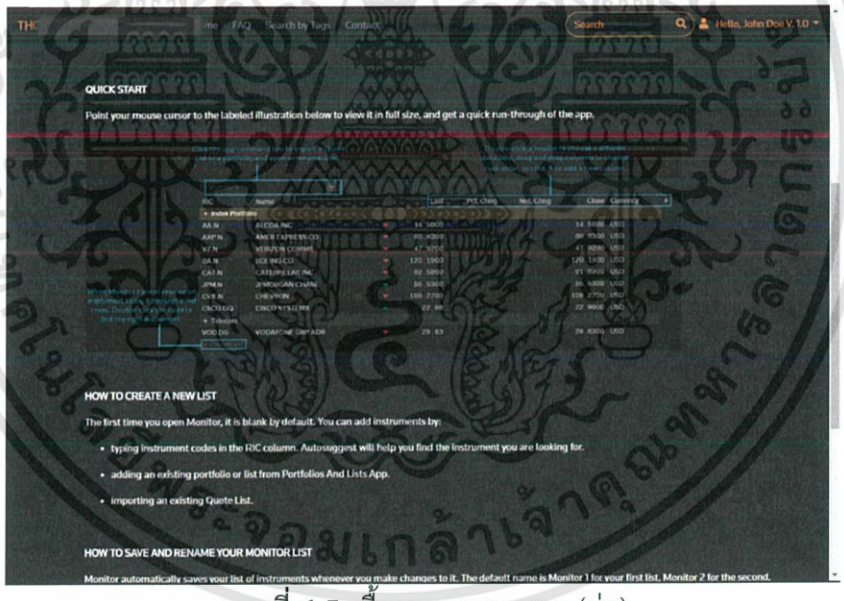
ภาพที่ 4.3 รายการของบทความที่ต้องการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกเลือกบทความที่ต้องการ จากนั้นบทความจะปรากฏขึ้น

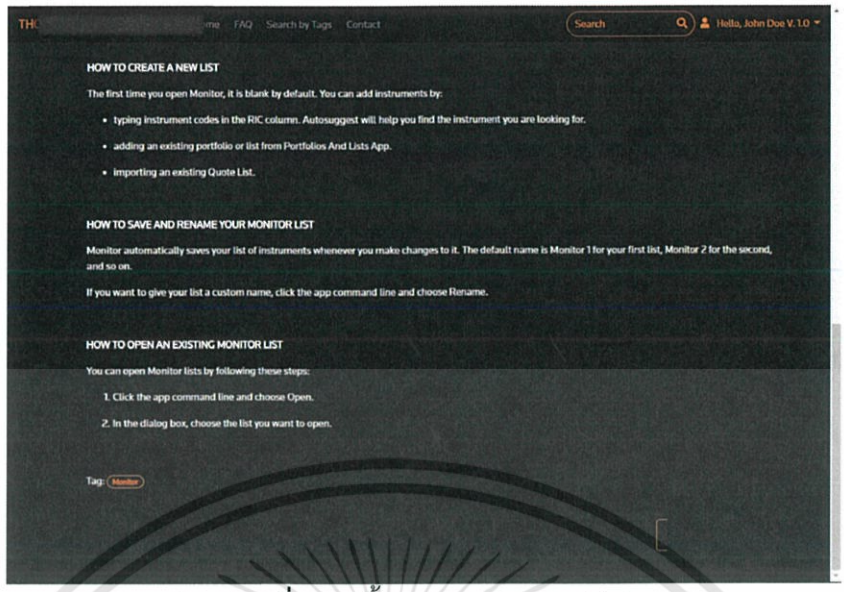


ภาพที่ 4.4 เนื้อหาของบทความ



ภาพที่ 4.5 เนื้อหาของบทความ (ต่อ)

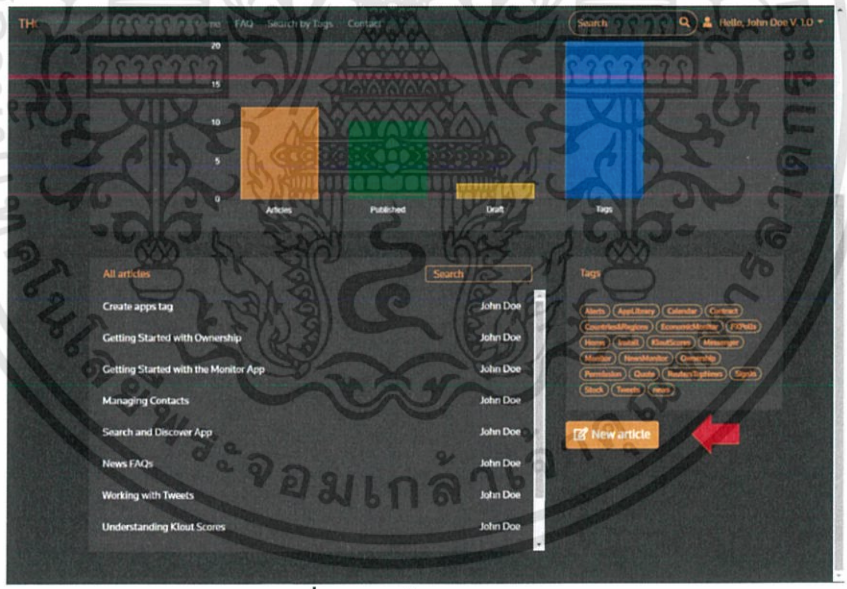
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.6 เนื้อหาของบทความ (ต่อ)

4.3.2 การสร้างบทความ สำหรับผู้ดูแลระบบ

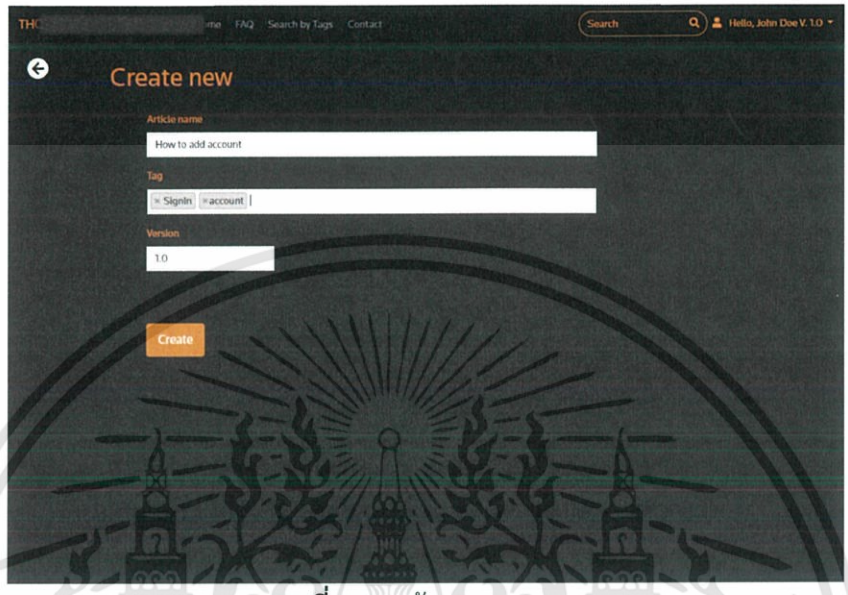
1. ที่หน้า Admin Portal คลิกที่ปุ่ม New Article



ภาพที่ 4.7 หน้า Admin portal

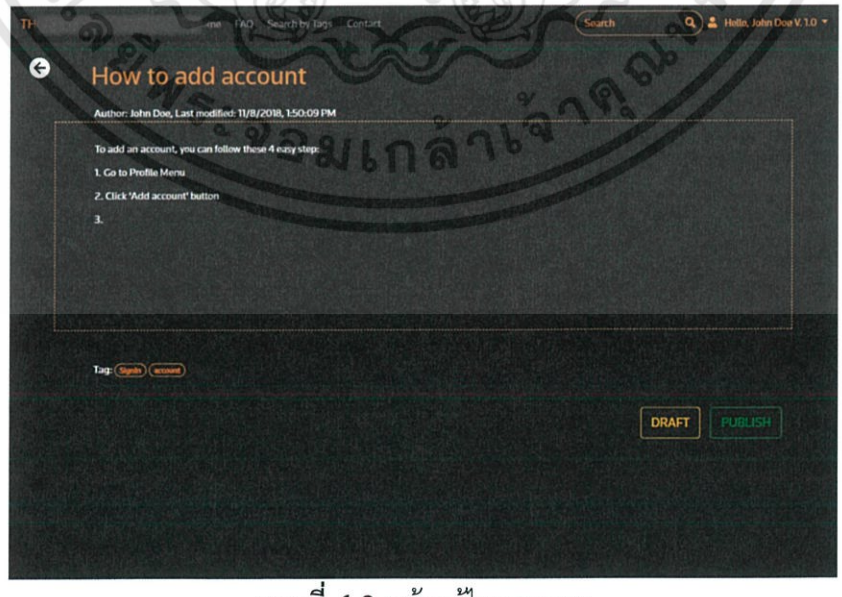
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. หน้า Create new จะปรากฏขึ้น พิมพ์ชื่อบทความ เพิ่ม Tag และใส่เวอร์ชันของบทความได้ตามต้องการ โดยในหน้านี้จะบังคับให้พิมพ์ชื่อบทความ หากไม่ได้กรอก จะมีข้อความแจ้งเตือนขึ้น และไม่สามารถไปยังขั้นตอนถัดไปได้ เมื่อกรอกข้อมูลเสร็จแล้ว คลิกปุ่ม Create



ภาพที่ 4.8 หน้า Create new

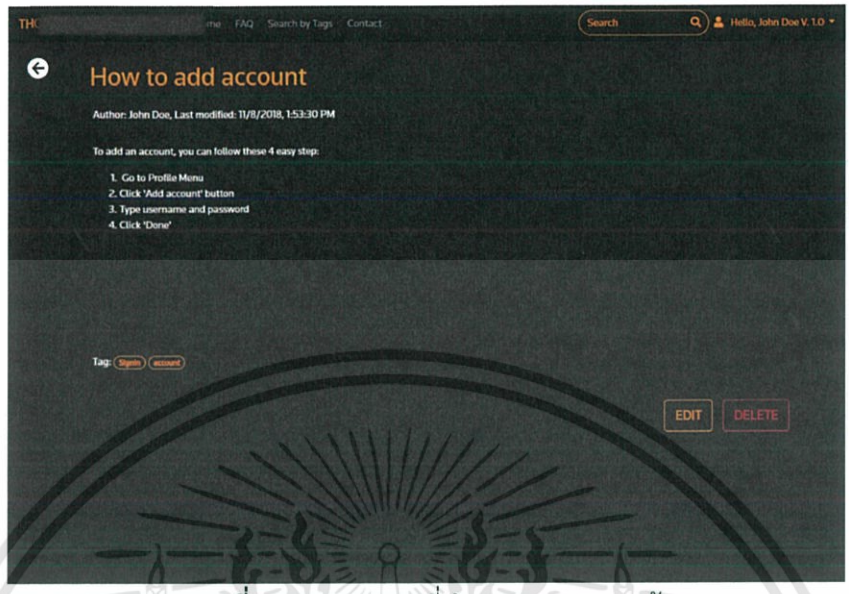
3. หน้าแก้ไขบทความจะปรากฏขึ้น ที่หน้านี้ผู้ใช้จะสามารถแก้ไขเพิ่มเติมข้อความได้ รวมไปถึงใส่รูปภาพ ตกแต่งสี จัดข้อความได้ เป็นต้น เมื่อเสร็จแล้วให้กดปุ่ม Publish เพื่อทำการลงบทความให้ลูกค้าเห็นได้ทันที หรือหากต้องการให้เป็นร่างบทความก็สามารถกดปุ่ม Draft ได้ ซึ่งลูกค้าจะยังไม่เห็นร่างบทความนี้จนกว่าจะกด Publish



ภาพที่ 4.9 หน้าแก้ไขบทความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

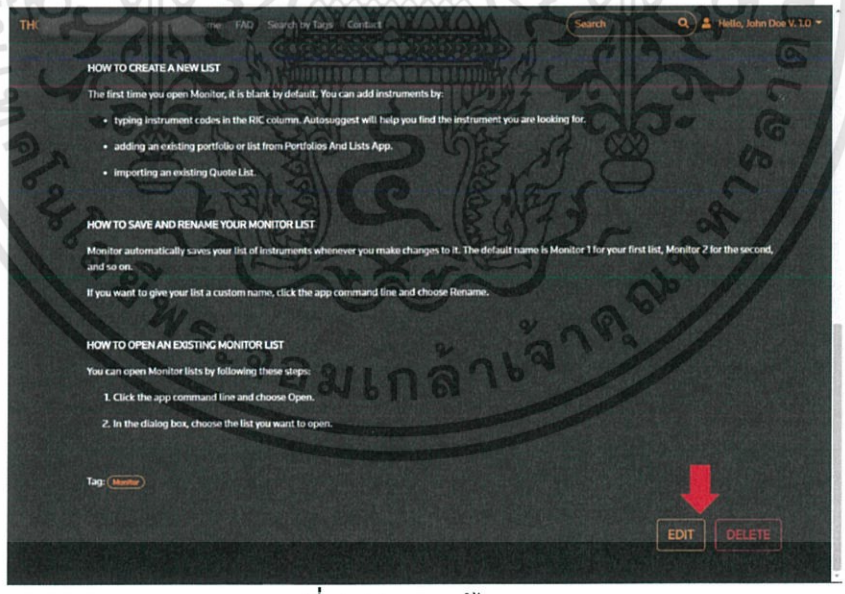
4. บทความที่ทำการ Publish แล้ว



ภาพที่ 4.10 บทความที่ทำการ Publish แล้ว

4.3.3 การแก้ไขบทความ สำหรับผู้ดูแลระบบ

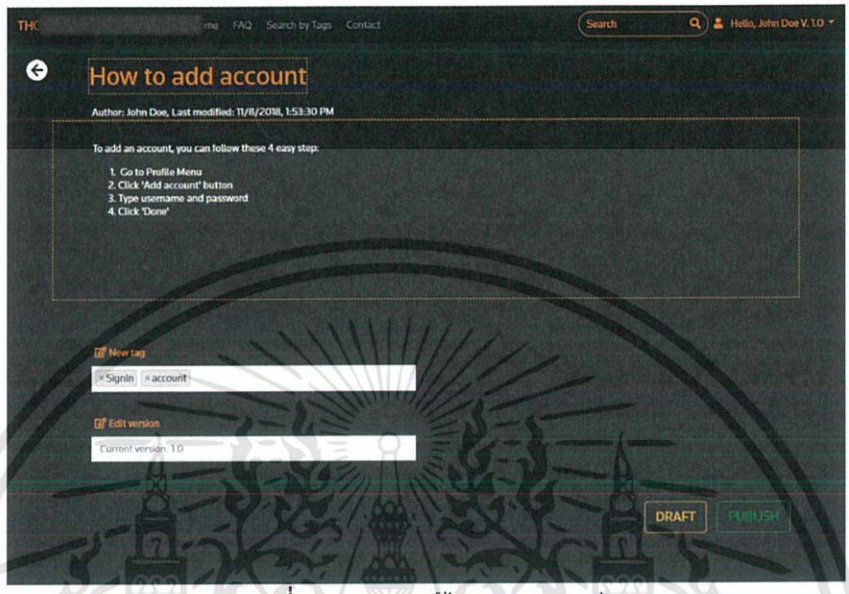
1. ที่หน้าบทความ คลิกปุ่ม Edit เพื่อทำการแก้ไขบทความ



ภาพที่ 4.11 การแก้ไขบทความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

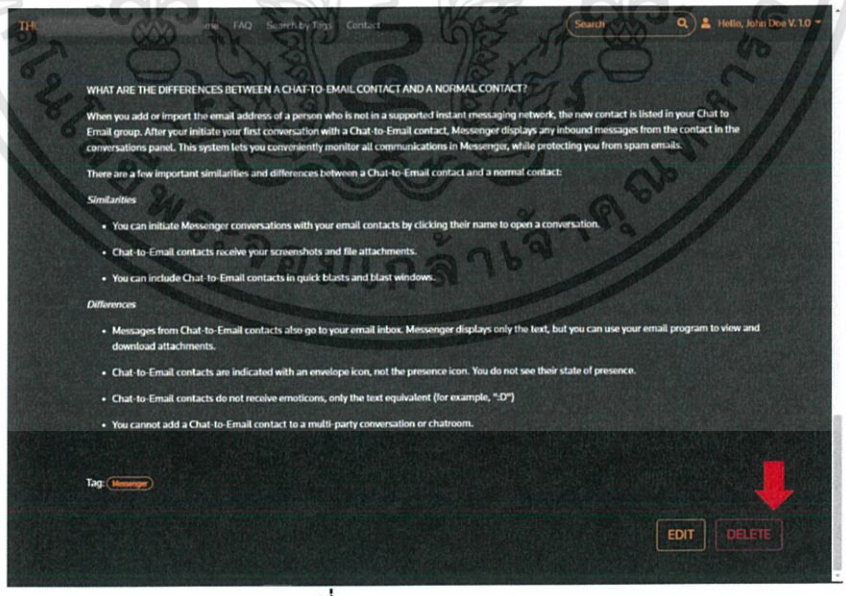
2. หน้าแก้ไขบทความจะปรากฏขึ้น สามารถแก้ไขข้อความ รวมไปถึง Tag และเวอร์ชันของบทความได้ เมื่อเสร็จแล้วคลิกปุ่ม Publish หรือถ้าหากต้องการให้เป็นบทความร่างก็สามารถกดปุ่ม Draft ได้



ภาพที่ 4.12 การแก้ไขบทความ (ต่อ)

4.3.4 การลบบทความ สำหรับผู้ดูแลระบบ

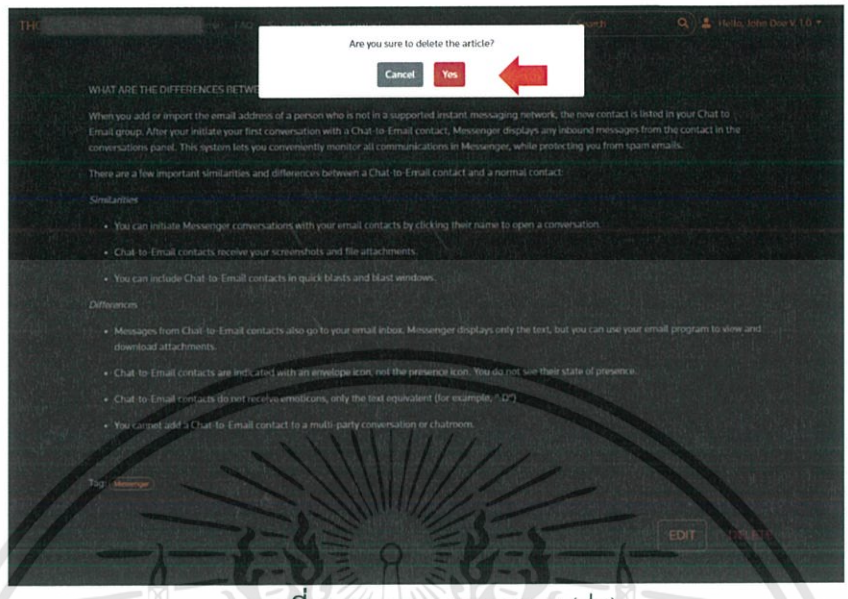
1. ที่หน้าบทความ คลิกปุ่ม Delete เพื่อทำการลบบทความ



ภาพที่ 4.13 การลบบทความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

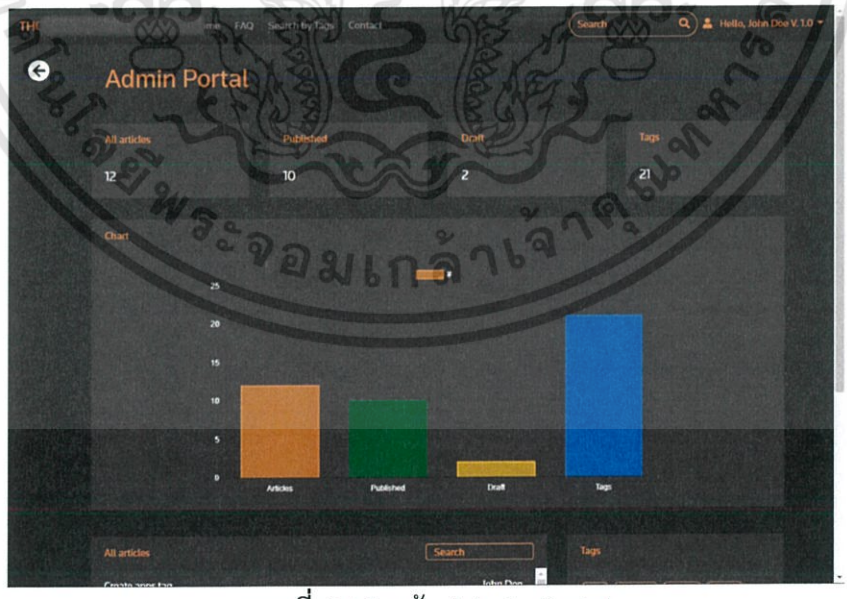
2. หน้าต่างยืนยันการลบจะปรากฏขึ้นมา กด Yes เพื่อยืนยัน จากนั้นบทความจะถูกลบทันที



ภาพที่ 4.14 การลบบทความ (ต่อ)

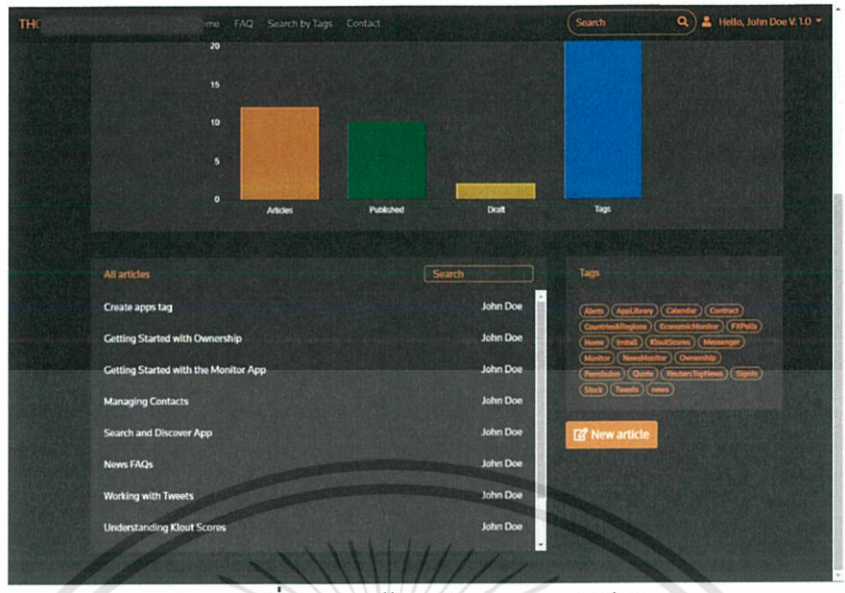
4.3.5 หน้า Admin Portal สำหรับผู้ดูแลระบบ

หน้า Admin Portal นี้ ด้านบนจะแสดงสถิติต่าง ๆ ของบทความ ประกอบไปด้วย จำนวนบทความทั้งหมด จำนวนบทความที่ถูก Publish แล้ว จำนวนบทความร่าง จำนวน Tag และแสดงผลออกมาเป็นกราฟ ด้านล่างจะแสดงรายการของบทความทั้งหมด และ Tag ทั้งหมด



ภาพที่ 4.15 หน้า Admin Portal

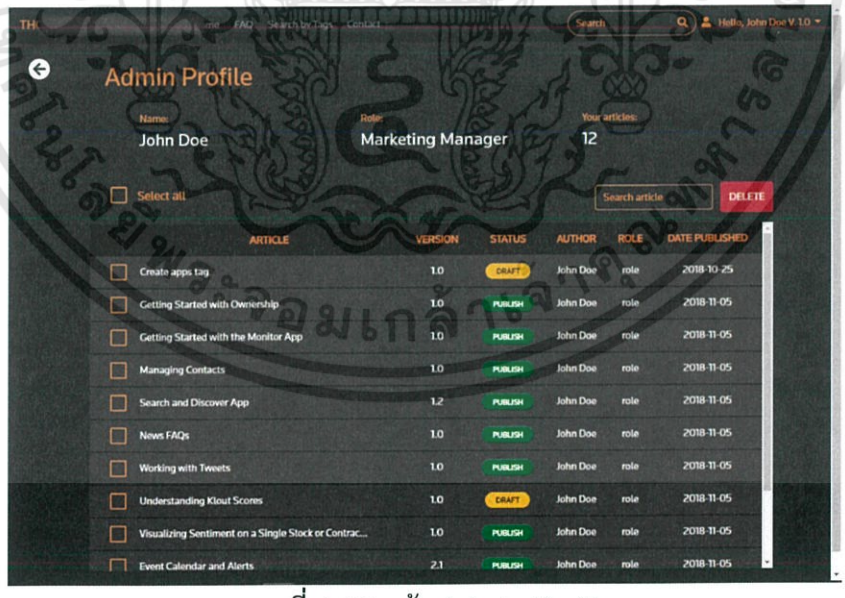
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.16 หน้า Admin Portal (ต่อ)

4.3.6 หน้า Admin Profile

หน้า Admin Profile นี้ จะแสดงชื่อ ตำแหน่ง และจำนวนบทความที่ผู้ดูแลระบบคนนั้น ๆ เป็นเจ้าของ พร้อมตารางแสดงรายการบทความทั้งหมดที่ผู้ดูแลระบบเป็นเจ้าของอยู่ ในหน้านี้ผู้ดูแลระบบสามารถเลือกหลายบทความเพื่อลบได้ โดยการคลิกที่ Checkbox ด้านซ้ายของชื่อบทความและคลิกปุ่ม Delete

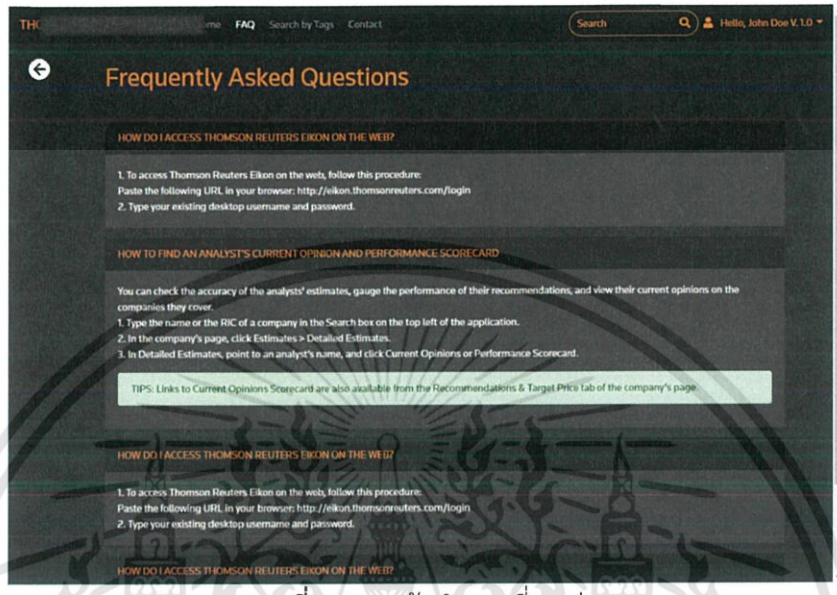


ภาพที่ 4.17 หน้า Admin Profile

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.7 หน้าคำถามที่พบบ่อย (Frequently Asked Question: FAQ)

หน้านี้จะรวบรวมคำถามที่ลูกค้ามักจะค้นหาหรือสอบถามบ่อย ๆ เพื่ออำนวยความสะดวกในการตอบคำถาม และลดเวลาการค้นหาบทความ

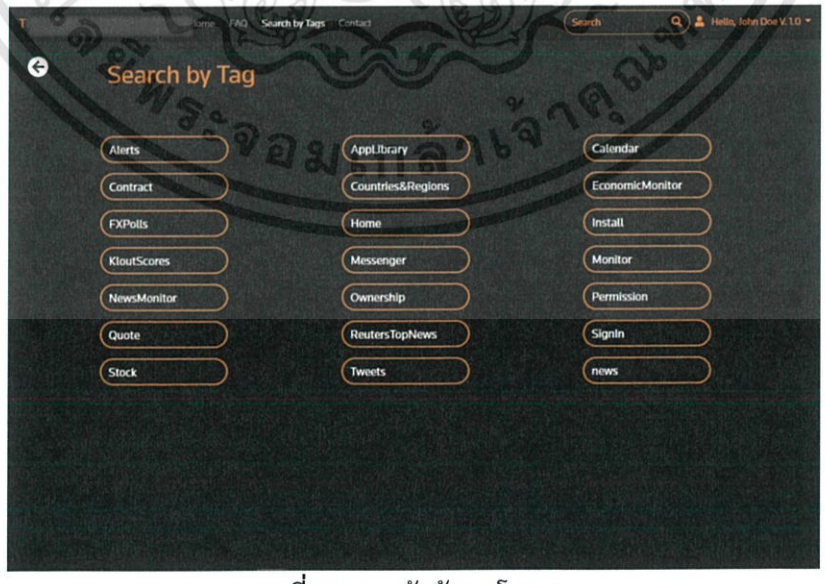


ภาพที่ 4.18 หน้าคำถามที่พบบ่อย

4.3.8 หน้าค้นหาโดย Tag (Search by Tags)

หน้านี้จะรวบรวม Tag ของทุกบทความไว้ ผู้ใช้สามารถค้นหาบทความโดยแยกประเภทตาม Tag ได้

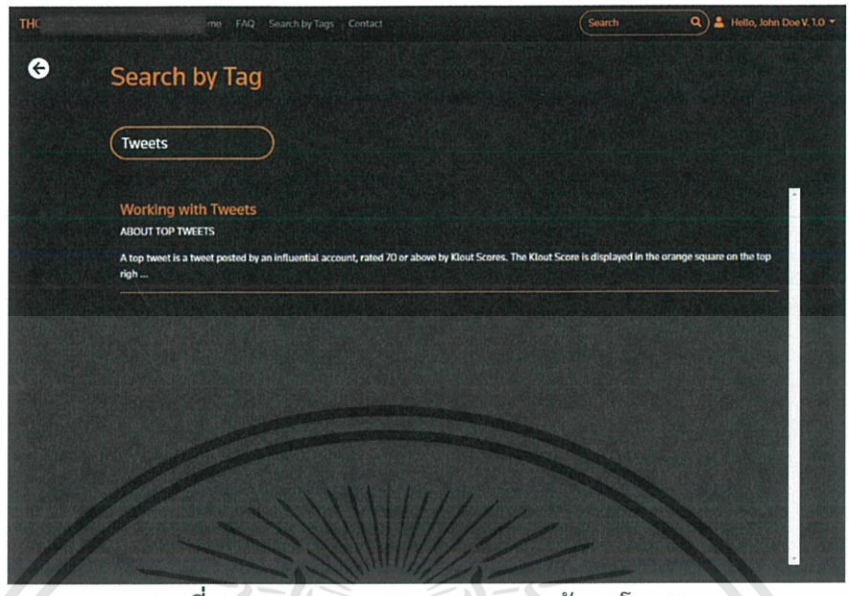
1. ที่หน้า Search by Tags คลิกเลือก Tag ที่ต้องการ



ภาพที่ 4.19 หน้าค้นหาโดย Tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

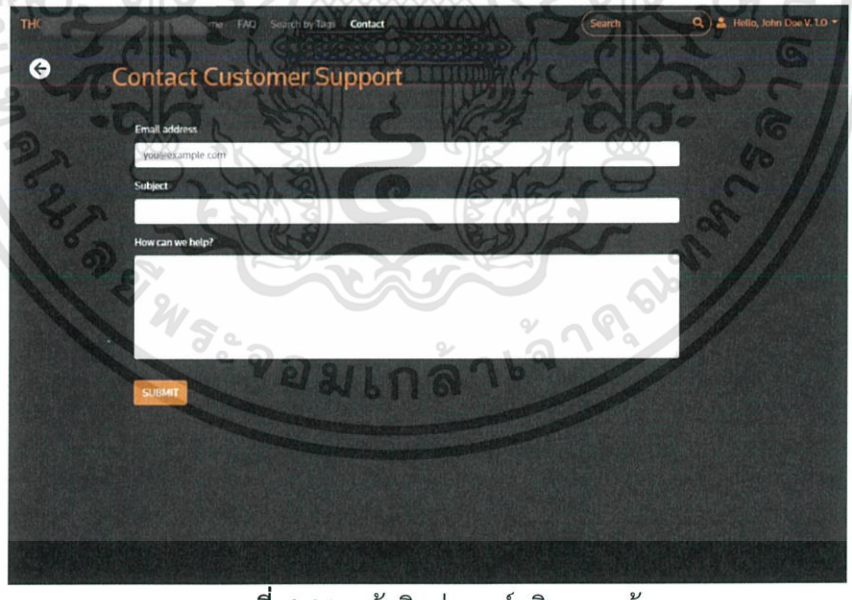
2. รายการของบทความจะปรากฏขึ้นมา คลิกเลือกบทความที่ต้องการ



ภาพที่ 4.20 รายการบทความจากการค้นหาโดย Tag

4.3.9 หน้าติดต่อศูนย์บริการลูกค้า (Contact Customer Support)

หน้าติดต่อศูนย์บริการลูกค้านี้ ลูกค้าสามารถอีเมลเพื่อสอบถามเรื่องที่ต้องการทราบที่อาจจะยังไม่มีเป็นบทความ หรือสอบถามเรื่องอื่น ๆ ได้



ภาพที่ 4.21 หน้าติดต่อศูนย์บริการลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

โครงงานสหกิจศึกษาที่ได้จัดทำขึ้นตามทฤษฎีที่เกี่ยวข้อง ได้แก่ เว็บแอปพลิเคชัน Front-end, Back-end, Software Stack, MVC, CMS, NoSQL Database, ภาษา HTML, ภาษา CSS, ภาษา Javascript และภาษา Typescript จึงทำให้สามารถใช้งานได้ตามความสามารถพื้นฐานที่ต้องการของระบบ โดยรวมถือว่าผลการดำเนินการบรรลุตามวัตถุประสงค์ที่กำหนดไว้ สามารถช่วยพัฒนาระบบเดิมของทางบริษัทให้ดียิ่งขึ้น รวมทั้งมีโอกาสได้นำเทคโนโลยีใหม่ ๆ มาใช้ในการจัดทำ ทั้งนี้ ระบบยังไม่ได้ถูกนำไปใช้จริงกับลูกค้าและผู้ดูแลระบบ จึงอาจจะยังไม่พบปัญหาจากการใช้งาน ซึ่งในส่วนนี้จะต้องทำการแก้ไขและพัฒนาให้ดียิ่งขึ้นในโอกาสต่อไป

5.2 ประโยชน์ที่ได้จากการปฏิบัติงานสหกิจศึกษา

1. ได้เรียนรู้การทำงานในบริษัทใหญ่ ๆ ในเรื่องของระบบการทำงาน วัฒนธรรมองค์กร การแบ่งงาน และการทำงานเป็นทีม ซึ่งทำให้เข้าใจวิถีของการทำงานมากขึ้น

2. ได้รับความรู้เชิงเทคนิคเกี่ยวกับสายงานด้านซอฟต์แวร์เพิ่มขึ้นเป็นอย่างมาก ตั้งแต่ด้านกระบวนการทำงานไปจนถึงการเขียนโปรแกรม ทำให้มีทักษะในการพัฒนาโปรแกรมเพิ่มขึ้น เพราะมีเทคโนโลยีใหม่ ๆ มากมายที่จะต้องศึกษาค้นคว้า และต้องตามให้ทันอยู่ตลอดเสมอ เพื่อนำมาใช้ในการจัดทำโครงงาน

3. ได้นำความรู้จากการศึกษาในสถาบันมาประยุกต์ใช้ในการทำงาน ทำให้เข้าใจในเนื้อหามากขึ้นว่าสามารถนำไปใช้จริงได้อย่างไร มีประโยชน์ต่อการทำงานอย่างไร และได้ศึกษาเรียนรู้เพิ่มเติมนอกเหนือจากความรู้ในเนื้อหาอีกด้วย

4. นอกจากทักษะด้านความรู้ (Hard Skill) ที่ได้ฝึกฝนมากขึ้นแล้ว ยังมีทักษะด้านสังคม (Soft Skill) ที่ได้ฝึกฝนด้วย คือได้ฝึกฝนทักษะด้านการสื่อสาร การใช้ภาษา มนุษยสัมพันธ์ ความอดทน มารยาททางสังคม และฝึกให้มีความรับผิดชอบมากขึ้น ซึ่งทั้ง Hard Skill และ Soft Skill เป็นสิ่งที่จำเป็นมากในการทำงานในยุคปัจจุบัน

5. ได้รู้จักและได้รับมิตรภาพจากเพื่อนต่างสถาบันมากขึ้น ไม่ว่าจะเป็นเพื่อนจากจุฬาลงกรณ์มหาวิทยาลัย, มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และมหาวิทยาลัยเกษตรศาสตร์ ได้แลกเปลี่ยนความรู้กันและประสบการณ์ระหว่างกัน ซึ่งเป็นการเพิ่มคอนเนคชันให้แก่ตนเอง

6. ได้รับประสบการณ์จริงจากชีวิตการทำงาน ทำให้ต้องมีความรับผิดชอบต่อตนเองมากขึ้น ตั้งแต่เรื่องการบริหารเวลา การเข้าทำงานตามกำหนดเวลาทุกวัน ซึ่งแตกต่างจากการเรียนใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาบันที่จะมีความยืดหยุ่นกว่า รวมทั้ง ยังได้เรียนรู้ว่าในการทำงานต้องมีรอบคอบให้มากขึ้น เพราะหากเกิดความผิดพลาดอาจจะมีผลกระทบทำให้งานในส่วนอื่นผิดพลาดตามไปด้วย นอกจากนี้ ยังได้ฝึกวินัยในตนเอง ฝึกเรื่องการทำงานเป็นทีมและการทำงานร่วมกับผู้อื่น ได้พบผู้คนที่หลากหลาย จึงต้องรู้จักการวางตัวที่ดี และการปรับตัวเข้ากับผู้อื่น

5.3 ปัญหาและอุปสรรคที่พบ

1. เนื่องจากผู้จัดทำยังไม่มีความรู้ด้านการทำเว็บแอปพลิเคชันมากนัก ทำให้ใช้เวลาในการศึกษาค่อนข้างมาก ทั้งในส่วนของการเลือกใช้เครื่องมือ หรือไลบรารี่ต่าง ๆ ที่มีจำนวนมาก จึงต้องลองนำมาใช้ก่อน รวมไปถึงการออกแบบโครงสร้างของเว็บแอปพลิเคชัน ทำให้การดำเนินงานในช่วงแรกเป็นไปค่อนข้างช้า

2. ในช่วงแรกที่เริ่มทำโครงงาน ยังไม่เข้าใจภาพรวมและระบบการทำงานของบริษัท จึงทำให้เกิดความสับสน ซึ่งกระทบต่อการออกแบบระบบ ทำให้ต้องปรับแก้อยู่หลายครั้ง

3. ในด้านการออกแบบที่ต้องออกแบบเองทั้งหมด แต่เนื่องจากยังไม่มีความชำนาญมากนัก อาจจะทำให้มีบางส่วนที่ยังไม่ครบถ้วนสมบูรณ์

5.4 ข้อเสนอแนะ

1. ควรปรับแต่ง UX ของตัวเว็บแอปพลิชันให้ดีขึ้น เพื่อตอบสนองการใช้งานของผู้ใช้ให้ผู้ใช้มีความสะดวกสบาย และใช้งานได้อย่างราบรื่นมากขึ้น

2. ควรเพิ่มฟีเจอร์ให้มากกว่าเดิม เพื่อทำให้เว็บแอปพลิเคชันมีความหลากหลายและมีประโยชน์ต่อผู้ใช้งานมากขึ้น

3. ควรปรับการเรียกข้อมูลจากฐานข้อมูลให้ใช้เวลาน้อยลง เพราะในกรณีที่มีข้อมูลมาก ๆ อาจจะทำให้การแสดงผลช้ากว่าปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

Software Development Life Cycle. (2017). [Online]. **Available from :** https://www.9chor.com/2017/11/19/software-development-life-cycle/ [accessed 6 November 2018]

Eikon. (2018). [Online] **Available from :** https://en.wikipedia.org/wiki/Eikon [accessed 6 November 2018]

Eikon Financial Analysis. [Online] **Available from :** https://financial.thomsonreuters.com/en/products/tools-applications/trading-investment-tools/eikon-trading-software.html [accessed 6 November 2018]

Atthakorn. (2015). *Framework คืออะไร ทำไมเราถึงต้องใช้ Framework?.* [Online]. **Available from** : http://www.atthakorn.com/ทำไมเราถึงต้องใช้-framework/ [accessed 6 November 2018]

8 Phases of the Web Design Process. (2014). [Online]. Available from : http://www.printmag.com/featured/phases-of-the-web-design-process/

เว็บแอปพลิเคชันคืออะไร. [Online]. **Available from :** http://aicomputer.co.th/sArticle/002-เว็บแอพพลิเคชั่น-Web-Application-คืออะไร.aspx [accessed 7 November 2018]

Amose Ndegwa. (2016). *What is a Web Application?.* [Online]. **Available from :** https://www.maxcdn.com/one/visual-glossary/web-application/ [accessed 7 November 2018]

อภิรัตน์ รัตนไพบูลย์. (2018). *UI UX คืออะไร.* [Online]. **Available from :** https://medium.com/@apirat/ui-ux-d2a5889f92bc [accessed 7 November 2018]

Software Stack. [Online]. **Available from :** https://www.techopedia.com/definition/27268/software-stack [accessed 7 November 2018]

Margaret Rouse. (2018). *Software Stack.* [Online]. **Available from :** https://searchmicroservices.techtarget.com/definition/software-stack [accessed 7 November 2018]

Cabot Technology Solution. (2017). *How to Pick the Right Web Technology Stack for Your Product.* [Online]. **Available from :** https://hackernoon.com/how-to-pick-the-right-web-technology-stack-for-your-product-f6d94440af2f [accessed 7 November 2018]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arnondora. (2015). *What is MVC?.* [Online]. **Available from :** https://arnondora.in.th/what-is-mvc

What is NoSQL?. [Online]. **Available from :** https://www.softmelt.com/article.php?id=579

Nuttavut Thongjor. (2016). *Typescript.* [Online]. **Available from :** https://www.babelcoder.com/blog/posts/typescript-data-types

Sebastian Eschweiler. (2018). *Angular 6 – Mean Stack Crash Course*. [Online]. **Available from :** https://medium.com/codingthesmartway-com-blog/angular-6-mean-stack-crash-course-part-4-completing-the-user-interface-94f2d4da3490

Database Schema Example. [Online]. **Available from :** https://www.mongodb.com/scale/database-schema-example

TinyMCE Documentation. [Online]. **Available from :** https://www.tiny.cloud/docs/

Pakin Phuhinkong. (2017). *Git.* [Online]. **Available from :** https://medium.com/@pakin/git-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-git-is-your-friend-c609c5f8efea

Tobias Günther. (2017). *8 Reasons for Switching to Git.* [Online]. **Available from :** https://www.git-tower.com/blog/8-reasons-for-switching-to-git

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การเขียนเว็บไซต์แบบ Responsive

การเขียนเว็บไซต์แบบ (Responsive Web Design) เป็นการเขียนเว็บไซต์เพื่อรองรับการแสดงผลบนอุปกรณ์พกพาต่าง ๆ ให้มีการแสดงผลที่สวยงาม เช่น โทรศัพท์มือถือ, แท็บเลต เป็นต้น โดยใช้ URL เดียวกันกับที่แสดงผลบนคอมพิวเตอร์ ซึ่งการเขียนเว็บไซต์แบบ Responsive มีความสำคัญมาก เพราะในปัจจุบันมี Mobile Internet User เพิ่มขึ้นเป็นจำนวนมาก ดังนั้นผู้พัฒนาจึงควรให้ความสำคัญกับการออกแบบเว็บไซต์ในลักษณะนี้

การทำ Responsive นั้น จะใช้การเขียน Media Queries ใน CSS เข้ามาช่วยกำหนดขอบเขต (Breakpoint) ของอุปกรณ์พกพาต่าง ๆ แบบมีลักษณะเป็นช่วง หรือระยะขนาดตามแต่อุปกรณ์นั้น ๆ โดยตามหลักการที่ถูกต้องคือการออกแบบให้แก่อุปกรณ์โทรศัพท์มือถือก่อน (Mobile First) จากนั้นค่อยใช้ Media Queries เข้ามาช่วยในอุปกรณ์ที่ใหญ่ขึ้น

```
// Extra small devices (portrait phones, less than 576px)
// No media query for `xs` since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }

// Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

// Extra large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }
```

ภาพที่ ก.1 ตัวอย่างการเขียน Media Queries

นอกจาก Media Queries แล้ว ยังมีการแบ่งพื้นที่แบบ Grid โดยใช้เทคนิคคอลัมน์ (Column) เข้ามาจัดการ โดยใน 1 แถว (Row) จะมีทั้งหมด 12 Column เท่า ๆ กัน เราสามารถแบ่งพื้นที่เป็นจำนวน Column ได้ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

One of three columns	One of three columns	One of three columns

Copy

```
<div class="container">
  <div class="row">
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
    <div class="col-sm">
      One of three columns
    </div>
  </div>
</div>
```

ภาพที่ ก.2 ตัวอย่างการแบ่งคอลัมน์จาก Bootstrap

ประโยชน์ของการทำ Responsive คือ ทำให้เว็บไซต์สามารถแสดงผลได้หลายอุปกรณ์โดยใช้โค้ดเพียงชุดเดียว อีกทั้งยังทำให้เว็บไซต์สวยงาม มีประสิทธิภาพ ส่งผลดีต่อ Search Engine เป็นอย่างมาก เพราะจะทำให้ Google สามารถเก็บข้อมูลจากเว็บไซต์ได้รวดเร็วและง่ายขึ้น และที่สำคัญที่สุดคือเป็นการตอบสนองความต้องการของผู้ใช้ในยุคที่มีการใช้ Smart Phone อย่างแพร่หลายในปัจจุบัน

ภาพที่ ก.3 Responsive Web Design

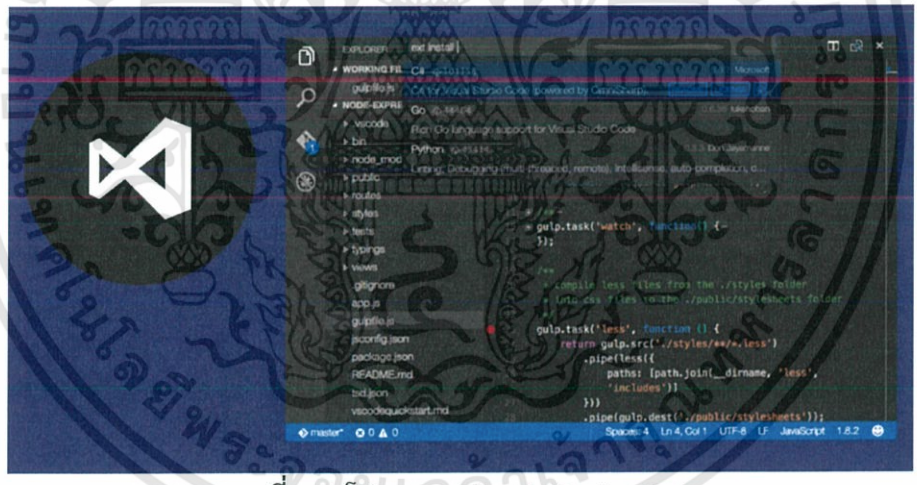
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
โปรแกรมเท็กซ์อิดิเตอร์ (Text Editor)

โปรแกรมเท็กซ์อิดิเตอร์ คือ โปรแกรมเขียนเอกสารประเภทโค้ด ที่มักใช้ในการเขียนเว็บไซต์ ตัวอย่าง Text Editor พื้นฐาน เช่น Notepad, EditPlus เป็นต้น โดย Text Editor ที่ผู้จัดทำใช้ในการพัฒนาโครงงานนี้คือโปรแกรม Visual Studio Code จากบริษัทไมโครซอฟต์ (Microsoft)

Visual Studio Code เป็น Text Editor ที่ได้รับความนิยมอย่างมากในปัจจุบัน เนื่องจากมีประสิทธิภาพสูง ใช้งานง่าย สวยงาม มี Plugin ให้เลือกมากมาย และใช้ขนาดพื้นที่คอมพิวเตอร์ไม่มาก สามารถดาวน์โหลดได้แบบไม่มีค่าใช้จ่าย

นอกจากนี้ Visual Studio Code ยังเหมาะกับผู้พัฒนาโปรแกรมที่ต้องการพัฒนาข้ามแพลตฟอร์ม (Platform) อีกด้วย ตัวโปรแกรมรองรับการใช้งานบนระบบปฏิบัติการ Windows, MacOS และ Linux รองรับการเขียนโปรแกรมหลากหลายภาษา เช่น Javascript, HTML, Typescript, C# ฯลฯ และสามารถเชื่อมต่อกับระบบ Git ได้



ภาพที่ ข.1 โปรแกรม Visual Studio Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

Git

Git คือ Version Control ประเภทหนึ่งที่ใช้จัดเก็บการเปลี่ยนแปลงของเอกสารประเภทใดก็ได้ สามารถใช้นำมาเก็บสำรอง Source Code แล้วเรียกดูหรือย้อนกลับไปยังเวอร์ชันใด ๆ ของโค้ดก็ได้ ช่วยในการจัดการโค้ดของการทำงานเป็นทีมสามารถทำได้สะดวกมากยิ่งขึ้น

ภาพที่ ค.1 Git

สาเหตุที่ควรนำ Git มาใช้ในการพัฒนาซอฟต์แวร์มีดังนี้

1. ประหยัดเวลา เพราะสามารถเก็บโค้ดทั้งหมดที่เขียนไว้ได้โดยใช้เวลาเพียงไม่กี่วินาทีต่อคำสั่ง Git 1 คำสั่ง
2. สามารถทำงานโดยไม่พึ่งพาอินเทอร์เน็ตได้ เพราะ Git สามารถทำให้เราเลือกได้ว่าเราต้องการจะทำงานในเครื่อง หรือรีโมต (Remote)
3. เมื่อเกิดความผิดพลาดใด ๆ ก็สามารถย้อนกลับไปยังเวอร์ชันที่ต้องการได้
4. สามารถดูการเปลี่ยนแปลงของโค้ดได้ ไม่ว่าจะเป็นเวลาที่เปลี่ยนแปลง ตำแหน่งใดของโค้ดที่เปลี่ยนแปลง และสมาชิกคนใดในทีมเป็นเปลี่ยนแปลงโค้ด เป็นต้น
5. สมาชิกทุกคนในทีมสามารถทำงานไปพร้อม ๆ กัน โดยการแยกโค้ดออกมาเขียนได้โดยไม่กระทบกับโค้ดของคนอื่นในทีมและโค้ดหลัก
6. มั่นใจได้ว่าโค้ดไม่สูญหายอย่างแน่นอน เพราะถูกเก็บไว้ใน Git หากโค้ดของเราเสียหายหรือไม่ได้เก็บสำรองไว้ ก็สามารถดึงข้อมูลมาจาก Git หรือจากสมาชิกคนใดคนหนึ่งในทีมได้

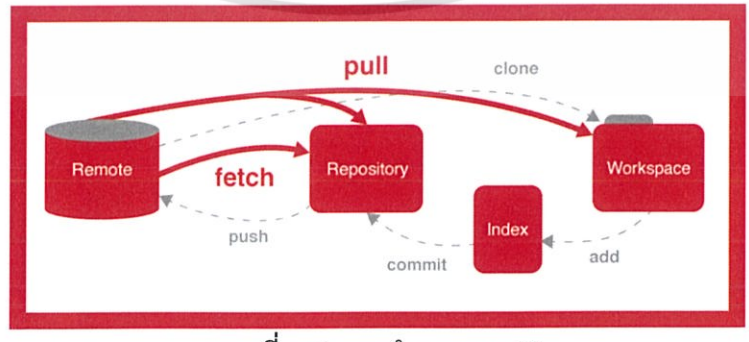
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้ง Git

- Windows ดาวน์โหลดตัวติดตั้งได้จาก https://git-scm.com/downloads
- Mac OS X Yosemite or later ติดตั้งผ่าน Terminal ให้พิมพ์ $ git version
- Linux ติดตั้งผ่าน Terminal ให้พิมพ์ $ sudo apt-get install git

การใช้งานคำสั่งพื้นฐาน Git ที่ใช้งานบ่อย บน Terminal

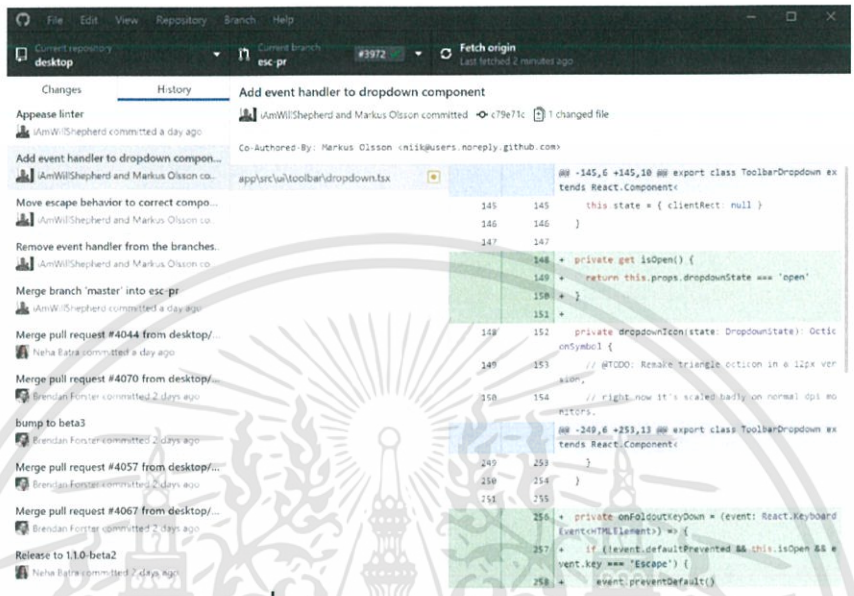
1. git status คือ แสดงไฟล์ที่มีการเปลี่ยนแปลง หรือยังต้องการ add หรือ commit
2. git add คือ การเพิ่มไฟล์เป็นสถานะ stage
3. git reset คือ การ unstage ไฟล์ที่เคยถูก stage
4. git commit คือการ commit ไฟล์ที่ stage โดยมักจะใช้ git commit -m เพื่อเพิ่มข้อความในสิ่งที่ทำไป
5. git log คือการดูประวัติที่เคย commit ไว้
6. git pull คือการ remote ไฟล์มายัง local โดยคำสั่ง git pull นั้นจะทำการ git fetch และ git merge ไปด้วย โดยเราจะมักเห็นใช้ git pull –rebase เพื่อทำการเปลี่ยนฐานแทนการ merge
7. git merge (branch) คือการรวม branch ที่เจาะจงมายัง local
8. git fetch คือ การตรวจสอบไฟล์ภายใน local และ remote ว่าตรงกันหรือไม่
9. git push คือ ส่งการเปลี่ยนแปลงของไฟล์ไปบน remote repository
10. git stash คือ การเก็บซ่อนการเปลี่ยนแปลงทั้งหมดไว้ทั้งหมด และสามารถนำกลับมาโดยใช้ git stash pop
11. git checkout (branch) คือ การเปลี่ยน branch
12. git clone (url) คือการ คัดลอกโปรเจคจาก remote มายัง local



ภาพที่ ค.2 การทำงานของ Git

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากการใช้งาน Git ผ่าน Terminal แล้ว ยังมีโปรแกรมที่นำมาใช้ในการจัดการระบบ Git ที่มี GUI (Graphic User Interface) อีกมากมาย ช่วยให้นักพัฒนาสามารถใช้งาน Git ได้สะดวกขึ้น ตัวอย่างเช่น GitHub Desktop, Source Tree, GitKraken เป็นต้น



ภาพที่ ค.3 โปรแกรม GitHub Desktop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้