

[เว็บแอพพลิเคชั่น](https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/3664-web-application-%E0%B9%80%E0%B8%A7%E0%B9%87%E0%B8%9A-%E0%B9%81%E0%B8%AD%E0%B8%9E%E0%B8%9E%E0%B8%A5%E0%B8%B4%E0%B9%80%E0%B8%84%E0%B8%8A%E0%B8%B1%E0%B9%88%E0%B8%99-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)แบบคำร้องสำหรับนักศึกษา

PETITION FORM WEB APPLICATION FOR STUDENT

**ธีระยุทธ์ เติมแต้ม**

**ตะวัน เข็มทอง**

**ศุภกิจ กิจนะบำรุงศักดิ์**

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะเทคโนโลยีอุตสาหกรรม**

**มหาวิทยาลัยราชภัฏสวนสุนันทา**

**ปีการศึกษา 2561**

**บทที่ 1**

**บทนำ**

* 1. **ความเป็นมาและความสำคัญของปัญหา**

ในอดีตการทำธุรกรรมต่าง ๆ นั้นจะต้องทำผ่านเอกสารที่เป็นกระดาษ ข้อเสียของการส่งเอกสารแบบกระดาษ คือ ล่าช้าเพราะต้องใช้คนเป็นผู้ดำเนินการส่งเอกสารถึงผู้รับ และอาจเกิดความเสียหายต่อเอกสาร เช่น เอกสารชำรุดและสูญหายจากผู้ดำเนินการเอง หรือเอกสารมีข้อความไม่ชัดเจน

ในปัจจุบัน สำนักวิชาการศึกษาทั่วไปฯ มหาวิทยาลัยราชภัฏสวนสุนันทา นั้นใช้เอกสารในการส่งแบบคำร้องแบบฟอร์มการกรอกเป็นกระดาษและเกิดข้อผิดพลาดต่าง ๆ เช่น เกิดความล่าช้าในการดำเนินการส่งเอกสารจากเจ้าหน้าที่ หรือปัญหาที่เกิดจากข้อมูลมีความหมายผิดไปจากเดิม ซึ่งเกิดจากลายมือผู้ส่งเอกสาร และรวมถึงการต้องเดินทางมายังมหาลัยเพื่อกรอกหรือส่งแบบคำร้องนั้น ๆ อาจจะทำให้เสียเวลาและค่าใช้จ่ายในการเดินทางอีกด้วย

ดังนั้นเราจึงต้องการสร้างระบบการส่งแบบคำร้องของ สำนักวิชาการศึกษาทั่วไปฯ มหาวิทยาลัยราชภัฏสวนสุนันทา ให้มีความสะดวกรวดเร็ว ลดการใช้กระดาษ ลดภาระค่าเดินทางและลดเวลาในการดำเนินงานและใช้งานได้ง่ายต่อผู้ใช้งานมากยิ่งขึ้น ไม่ว่าจะเป็นนักศึกษาที่สามารถส่งแบบคำร้องได้จากทุกที่ผ่านเว็บแอพพลิเคชันของเรา และผู้ดูแลระบบจะลดภาระและระยะเวลาในการส่งไปยังเจ้าหน้าที่ที่เกี่ยวข้อง  อีกทั้งเว็บแอพพลิเคชันของเราจะเข้ามาแก้ปัญหาต่าง ๆ

* 1. **วัตถุประสงค์ของโครงงาน**
     1. เพื่อสร้างเว็บไซต์
     2. ศึกษาปัญหาการส่งแบบคำร้องของสำนักงานวิชาการฯ
     3. เพื่อออกแบบและศึกษาฐานข้อมูล
     4. ประเมินประสิทธิภาพจากผู้ใช้งานจริง
  2. **ขอบเขตของโครงงาน**
     1. ระบบใช้สำหรับสำนักวิชาการศึกษาทั่วไปฯ มหาวิทยาลัยราชภัฏสวนสุนันทา
     2. ภาษาที่ใช้ในการพัฒนา CSS, HTML, JavaScript, PHP,SQL
     3. ทำงานบนเซิร์ฟเวอร์เดียว
     4. มีการประเมินความพอใจจากผู้ใช้งานได้มากกว่า 80%
     5. มีระบบ 2 ภาษา (ไทย,อังกฤษ)
     6. จะมีการแจ้งเตือนไปเมื่อเอกสารมีการเปลี่ยนแปลง
     7. รูปแบบไฟล์ที่สามารถแนบพร้อมเอกสารคือ .PDF,.JPG, .PNG, .DOC
     8. สามารถใช้ได้ทั้งคอมพิวเตอร์ และสมาร์ทโฟน
     9. เอกสารใบคำร้อง
        1. ใบคำร้องขอรหัสผ่านเข้าระบบ
        2. แบบคำร้องทั่วไป
        3. แบบใบลาป่วย ลากิจ
        4. แบบคำร้องขอตรวจสอบผลการเรียน
        5. แบบฟอร์มแจ้งสาเหตุการขาดสอบรายวิชาศึกษาทั่วไป
        6. แบบคำร้องขอแก้ไขผลการเรียน
        7. แบบคำร้องขอสอบภายหลัง
     10. ข้อมูลส่วนตัวของผู้ใช้งาน
         1. รหัสผ่าน
         2. ชื่อ – นามสกุล
         3. รหัสนศ.
         4. อีเมล
         5. เบอร์โทร
         6. คณะ
         7. สาขา
     11. ประเภทของผู้ใช้งาน
         1. นักศึกษา
         2. เจ้าหน้าที่(พนักงาน,อาจารย์)
         3. ผู้ดูแลระบบ
     12. โดยที่นักศึกษาสามารถใช้งานได้ดังนี้
         1. ส่งเอกสารแบบคำร้องเป็นแบบฟอร์ม
         2. ติดตามเอกสารว่าขณะนั้นดำเนินการอยู่ในขั้นตอนใด
         3. แก้ไขข้อมูลส่วนตัวของนักศึกษา เช่น ชื่อ-นามสกุล รหัสผ่าน เบอร์โทรศัพท์และอีเมล เป็นต้น
         4. ติดต่อผู้ดูแลระบบเมื่อต้องการสอบถามปัญหาต่าง ๆ
     13. โดยที่เจ้าหน้าที่สามารถใช้งานได้ดังนี้
         1. ตรวจสอบแบบคำร้องที่ยังไม่ถูกดำเนินการได้ง่ายและรวดเร็วยิ่งขึ้น
         2. ตรวจสอบแบบคำร้องที่ถูกดำเนินการไปแล้วได้
         3. โดยที่ผู้ดูแลระบบสามารถทำได้ดังนี้
         4. ค้นหาเอกสารแบบคำร้องทั้งหมดหรือกำหนดประเภท
         5. ค้นหาสมาชิกทั้งหมดหรือค้นหารายบุคคล
         6. จัดการลบหรือแก้ไขข้อมูลสมาชิก
         7. แก้ไขข้อมูลเจ้าหน้าที่
         8. จัดการข่าวประชาสัมพันธ์ที่ด้านหน้าเว็บไซต์
         9. ตั้งค่าเส้นทางเอกสารแบบคำร้องแบบระบุบุคคลตรวจสอบ
         10. รับและส่งข้อความการติดต่อจากผู้ใช้
  3. **ประโยชน์ที่คาดว่าจะได้รับ**
     1. เพื่อสร้างความสะดวกสบายให้กับผู้ใช้งานมากขึ้น
     2. ลดเวลาในการดำเนินงานเอกสาร
     3. ลดค่าใช้จ่ายในการเดินทาง
     4. สามารถติดตามคำร้องได้ง่าย
     5. เจ้าหน้าที่สามารถทำงานนอกสถานที่ได้
     6. เพื่อให้สำนักงานวิชาการศึกษาทั่วไปฯมหาวิทยาลัยราชภัฏสวนสุนันทาได้นำไปใช้งานจริง
  4. **นิยามศัพท์เฉพาะ**
     1. เว็บแอปพลิเคชั่น (Web Application) คือ [โปรแกรมประยุกต์](https://th.wikipedia.org/wiki/%E0%B9%82%E0%B8%9B%E0%B8%A3%E0%B9%81%E0%B8%81%E0%B8%A3%E0%B8%A1%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B8%A2%E0%B8%B8%E0%B8%81%E0%B8%95%E0%B9%8C)ที่เข้าถึงด้วย[โปรแกรมค้นดูเว็บ](https://th.wikipedia.org/wiki/%E0%B9%82%E0%B8%9B%E0%B8%A3%E0%B9%81%E0%B8%81%E0%B8%A3%E0%B8%A1%E0%B8%84%E0%B9%89%E0%B8%99%E0%B8%94%E0%B8%B9%E0%B9%80%E0%B8%A7%E0%B9%87%E0%B8%9A)ผ่าน[เครือข่ายคอมพิวเตอร์](https://th.wikipedia.org/wiki/%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B7%E0%B8%AD%E0%B8%82%E0%B9%88%E0%B8%B2%E0%B8%A2%E0%B8%84%E0%B8%AD%E0%B8%A1%E0%B8%9E%E0%B8%B4%E0%B8%A7%E0%B9%80%E0%B8%95%E0%B8%AD%E0%B8%A3%E0%B9%8C)อย่าง[อินเทอร์เน็ต](https://th.wikipedia.org/wiki/%E0%B8%AD%E0%B8%B4%E0%B8%99%E0%B9%80%E0%B8%97%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B9%80%E0%B8%99%E0%B9%87%E0%B8%95)หรือ[อินทราเน็ต](https://th.wikipedia.org/wiki/%E0%B8%AD%E0%B8%B4%E0%B8%99%E0%B8%97%E0%B8%A3%E0%B8%B2%E0%B9%80%E0%B8%99%E0%B9%87%E0%B8%95)เว็บแอปพลิเคชันเป็นที่นิยมเนื่องจากความสามารถในการอัปเดตและดูแล โดยไม่ต้องแจกจ่ายและติดตั้งบนเครื่องผู้ใช้
     2. เซิร์ฟเวอร์ (Server) คือ คอมพิวเตอร์ชนิดหนึ่งที่มีไว้สำหรับเก็บข้อมูลเพื่อแสดงเว็บไซต์[ระบบปฏิบัติการ](https://th.wikipedia.org/wiki/%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A%E0%B8%9B%E0%B8%8F%E0%B8%B4%E0%B8%9A%E0%B8%B1%E0%B8%95%E0%B8%B4%E0%B8%81%E0%B8%B2%E0%B8%A3)อีกชั้นหนึ่ง
     3. อีเมล (E-mail) คือ จดหมายอิเล็กทรอนิกส์ที่เราสามารถส่งและรับข้อมูลได้โดยการสื่อสารผ่านเครือข่าย Internet ซึ่งมีความรวดเร็วมาก ไม่ว่าคุณจะอยู่ที่ไหนในโลกนี้ถ้ามีอีเมล์คุณก็สามารถที่จะส่งหรือรับอีเมล์ได้จากทุกที่เพียงแค่คุณมี Internet ใช้งานโดยข้อมูลที่ส่งผ่านอีเมล์นั้นสามารถที่จะอยู่ในรูปของข้อความ, เสียง, รูปภาพ และ วิดีโอ
     4. **เจพีจี (JPG) คือ** รูปแบบการบีบอัดแฟ้มภาพแบบสูญเสีย โดยยังให้เสียความละเอียดน้อยที่สุด รูปแบบแฟ้มสำหรับวิธีการนี้ได้แก่ .jpeg, .jpg, .jpe, .jfif, .jfi (จะเป็นตัวเล็กหรือตัวใหญ่ก็ได้)
     5. **พีเอนจี (PNG) คือ** เป็นรูปแบบรูปภาพที่พัฒนาขึ้นมามาจากรูปแบบรุปภาพ GIF  เพื่อแก้ปัญหาด้านสิทธิบัตรของภาพแบบ GIF โดยที่ ไฟล์ภาพแบบ PNG ไลบรารีสำหรับ PNG คือ libpng ซึ่งเขียนด้วยภาษาซี ปัจจุบัน PNG สนับสนุนโดยเว็บเบราว์เซอร์เกือบทุกตัว
     6. **ดีโอซี (DOC) คือ** นามสกุลไฟล์ของแฟ้มเอกสาร ซึ่งสร้างขึ้นโดยใช้โปรแกรม Microsoft Word ซึ่งเป็นโปรแกรมประมวลผลคำ (word processing) แฟ้มเอกสารประเภทนี้จะประกอบด้วยข้อความที่เป็นตัวหนังสือ อาจมีภาพด้วยหรือไม่ก็ได้
     7. **พีดีเอฟ (PDF) คือ** ไฟล์ประเภทหนึ่งที่สร้างมาจากโปรแกรม ประเภท PDF Creator ซึ่งเดิมทีจะรู้จักไฟล์ PDF จาก Acrobat ที่ถูกพฒั นาข้ึนจากทีมงานของ Adobe ด้วยโปรแกรม Adobe Acrobat ซึ่งคุณสมบตัิเบ้ืองตน้ของไฟล์ PDF อันเป็นไฟล์งานที่ไม่สามารถ แกไ้ขได้และรูปแบบก็เหมือนต้นฉบับเดิมจึงเหมาะที่จะใช้สำหรับ
     8. **เฮชทีเอมแอล (HTML) คือ** เป็นภาษาประเภท Markup Language ที่ใช้ในการสร้างเว็บเพจ   มีแม่แบบมาจากภาษา SGML (Standard Generalized Markup Language) ที่ตัดความสามารถบางส่วนออกไป เพื่อให้สามารถทำความเข้าใจและเรียนรู้ได้ง่าย   ปัจจุบันมีการพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C)
     9. **ซีเอสเอส (CSS) คือ** ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง พื้นหลัง เส้นขอบและอื่น ๆ ตามที่ต้องการ CSS ย่อมาจาก Cascading Style Sheets มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกกำหนดมาตรฐานโดย W3C เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ ได้รับความนิยมอย่างแพร่หลาย
     10. **จาวาสคริป (JavaScript) คือ** ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง Java JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช่ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียลเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา [HTML](https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2026-html-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)และภาษา [Java](https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2185-java-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)
     11. **พีเอชพี (PHP) คือ** ภาษาคอมพิวเตอร์จำพวก scripting language ภาษาจำพวกนี้คำสั่งต่างๆจะเก็บอยู่ในไฟล์ที่เรียกว่า script และเวลาใช้งานต้องอาศัยตัวแปรชุดคำสั่ง ตัวอย่างของภาษาสคริปก็เช่น [JavaScript](https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2187-java-javascript-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html), Perl เป็นต้น ลักษณะของ PHP ที่แตกต่างจากภาษาสคริปต์แบบอื่นๆ คือ PHP ได้รับการพัฒนาและออกแบบมา เพื่อใช้งานในการสร้างเอกสารแบบ [HTML](https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2026-html-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)โดยสามารถสอดแทรกหรือแก้ไขเนื้อหาได้โดยอัตโนมัติ ดังนั้นจึงกล่าวว่า PHP เป็นภาษาที่เรียกว่า server-side หรือ HTML-embedded scripting language นั้นคือในทุกๆ ครั้งก่อนที่เครื่องคอมพิวเตอร์ซึ่งให้บริการเป็น [Web server](https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2053-web-server-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.html)จะส่งหน้าเว็บเพจที่เขียนด้วย PHP ให้เรา มันจะทำการประมวลผลตามคำสั่งที่มีอยู่ให้เสร็จเสียก่อน แล้วจึงค่อยส่งผลลัพธ์ที่ได้ให้เรา ผลลัพธ์ที่ได้นั้นก็คือเว็บเพจที่เราเห็นนั่นเอง  ถือได้ว่า PHP เป็นเครื่องมือที่สำคัญชนิดหนึ่งที่ช่วยให้เราสามารถสร้าง Dynamic Web pages (เว็บเพจที่มีการโต้ตอบกับผู้ใช้) ได้อย่างมีประสิทธิภาพและมีลูกเล่นมากขึ้น
     12. **เอสคิวแอล (SQL) คือ** เป็นภาษามาตรฐานในการเข้าถึง ฐานข้อมูล เราสามารถใช้งานภาษา SQL ได้จากโปรแกรมต่าง ๆ ที่ต้องทำการกับระบบฐานข้อมูล เช่น ใช้ SQL ในการทำการดึงข้อมูล (Retrieve Data) จากฐานข้อมูล และมันเป็นมาตรฐานกลางที่ใช้ในระบบฐานข้อมูลต่าง ๆ โดยเป็นมาตรฐานของ ANSI (American National Standard Institute)
  5. **ระยะเวลาในการดำเนินงาน**

ในการดำเนินงานจะแบ่งออกเป็น 3 ขั้นตอนหลัก คือ

* + 1. ขั้นเตรียมการ
       1. คนคว้าข้อมูลเกี่ยวกับเรื่องที่สนใจเพื่อเลือกหัวข้อโครงงาน
       2. ค้นคว้าข้อมูลเกี่ยวกับหัวข้อที่เลือก
       3. ศึกษาและทำความเข้าใจข้อมูล
    2. ขั้นดำเนินการ
       1. ออกแบบเขียนโปรแกรม

- ออกแบบหน้าเว็บแอพพลิเคชั่น

* + - 1. ลงมือปฏิบัติงาน

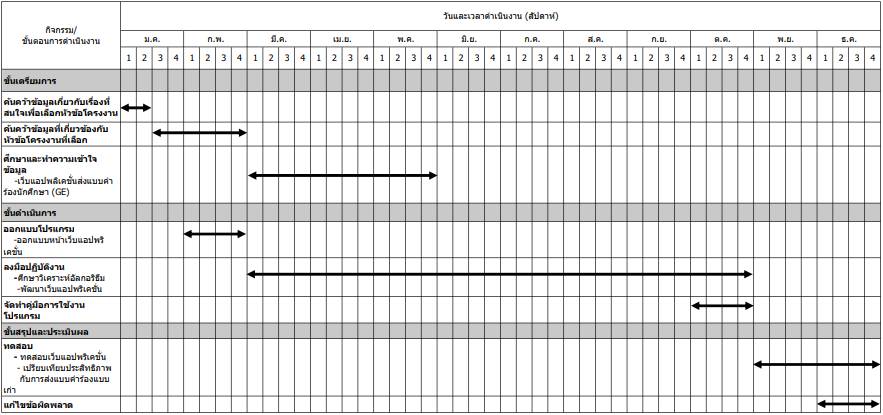
- ศึกษาวิเคราะห์อัลกอริทึม

* + - 1. จัดทำคู่มือการใช้งานโปรแกรม
    1. ขั้นตอนประเมินผล และสรุป
       1. ทดสอบ

- ทดสอบเว็บแอพพลิเคชั่น

- เปรียบเทียบประสิทธิภาพกับการส่งคำร้องแบบเก่า

* + - 1. แก้ไขข้อผิดพลาด

**รูปที่ 1.1** ตารางวันเวลาดำเนินงาน

**บทที่ 2**

**ทฤษฎีที่เกี่ยวข้อง**

ในการดำเนินโครงการเว็บแอพพลิเคชั่นในการตกแต่งและจัดการข้อมูลโดยใช้หน่วยประมวลผลกราฟฟิกบนเว็บแอพพลิเคชั่นซึ่งสิ่งที่สำคัญคือผู้จัดทำต้องรวบรวม

และค้นคว้าหาข้อมูลที่เกี่ยวข้องเพื่อที่จะบรรลุตามจุดประสงค์ของโครงการ

โดยทำการศึกษาข้อมูลตามลำดับดังนี้

2.1 เอชทีเอ็มแอล 5

2.2 พีเอชพี

2.3 เอสคิวแอล

2.4 มายเอสคิวแอลเซอเวอร์

2.5 ซีเอสเอส

2.6 จาวาสคริปต์

2.7 เจเควียรี

2.8 ลาร้าเวล

2.9 แซมป์

2.10 คอมโพเซอร์

2.11 บูทสแตป

2.12 เอ็มพีดีเอฟ

2.13 กูเกิล ฟอนต์

2.14 โหนดเจเอส

**2.1 เอชทีเอ็มแอล 5**

เอชทีเอ็มแอล คือ ภาษาหลักที่ใช้ในการเขียนเว็บเพจ โดยใช้ แทก ในการกำหนดการแสดงผล เอชทีเอ็มแอล ย่อมาจากคำว่า Hypertext Markup Language โดย Hypertext หมายถึง ข้อความที่เชื่อมต่อกันผ่านลิ้ง (Hyperlink) Markup language หมายถึงภาษาที่ใช้ Tag ในการกำหนดการแสดงผลสิ่งต่าง ๆ ที่แสดงอยู่บนเว็บเพจ ดังนั้น HTML จึงหมายถึง ภาษาที่ใช้ Tag ในการกำหนดการแสดงผลเว็บเพจที่ต่างก็เชื่อมถึงกันใน Hyperspace ผ่าน Hyperlink

2.1.1 ความสามารถของ เอชทีเอ็มแอล 5

2.1.1.1 ส่วนของคำสั่ง (tag) เป็นส่วนที่กำหนดรูปแบบของข้อความที่แสดง ซึ่งเราเรียกว่า Tag โดยจะอยู่ในเครื่องหมาย < ... >

2.1.1.2 ส่วนของบทความทั่วไปเป็นส่วนของข้อความที่เราต้องการแสดงผลบนบราวเซอร์ [2]

**ตัวอย่าง**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

**2.2 พีเอชพี**

พีเอชพี คือ ภาษาคอมพิวเตอร์ในลักษณะเซิร์ฟเวอร์-ไซด์ สคริปต์ โดยลิขสิทธิ์อยู่ในลักษณะโอเพนซอร์ส ภาษาพีเอชพีใช้สำหรับจัดทำเว็บไซต์ และแสดงผลออกมาในรูปแบบ HTML โดยมีรากฐานโครงสร้างคำสั่งมาจากภาษา ภาษาซี ภาษาจาวา และ ภาษาเพิร์ล ซึ่ง ภาษาพีเอชพี นั้นง่ายต่อการเรียนรู้ ซึ่งเป้าหมายหลักของภาษานี้ คือให้นักพัฒนาเว็บไซต์สามารถเขียน เว็บเพจ ที่มีการตอบโต้ได้อย่างรวดเร็ว

2.2.1 ความสามารถของ พีเอชพี

2.2.1.1 การสร้างเนื้อหาอัตโนมัติจัดการคำสั่ง

2.2.1.2 การอ่านข้อมูลจากผู้ใช้และประมวลผล

2.2.1.3 การอ่านข้อมูลจากดาต้าเบส

2.2.1.4 ความสามารถจัดการกับคุกกี้ [2]

**ตัวอย่าง**

<!DOCTYPE html>

<html>

<body>

<?php

echo "My first PHP script!";

?>

</body>

</html>

**2.3 เอสคิวแอล**

SQL ย่อมาจาก structured query language คือภาษาที่ใช้ในการเขียนโปรแกรม เพื่อจัดการกับฐานข้อมูลโดยเฉพาะ เป็นภาษามาตราฐานบนระบบฐานข้อมูลเชิงสัมพันธ์และเป็นระบบเปิด (open system) หมายถึงเราสามารถใช้คำสั่ง sql กับฐานข้อมูลชนิดใดก็ได้ และ คำสั่งงานเดียวกันเมื่อสั่งงานผ่าน ระบบฐานข้อมูลที่แตกต่างกันจะได้ ผลลัพธ์เหมือนกัน ทำให้เราสามารถเลือกใช้ฐานข้อมูล ชนิดใดก็ได้โดยไม่ติดยึดกับฐานข้อมูลใดฐานข้อมูลหนึ่ง นอกจากนี้แล้ว SQL ยังเป็นชื่อโปรแกรมฐานข้อมูล ซึ่งโปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์

2.3.1 ความสามารถของ เอสคิวแอล

2.3.1.1 สร้างฐานข้อมูลและ ตาราง

2.3.1.2 สนับสนุนการจัดการฐานข้อมูล ซึ่งประกอบด้วย การเพิ่ม การปรับปรุง และการลบข้อมูล

2.3.1.3 สนับสนุนการเรียกใช้หรือ ค้นหาข้อมูล [2]

**ตัวอย่าง**

SELECT Country FROM Customers;

SELECT \* FROM Customers WHERE Country='Mexico';

UPDATE Customers SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'

WHERE CustomerID = 1;

DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

**2.4 มายเอสคิวแอลเซอเวอร์**

MySQL (มายเอสคิวแอล) เป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System) โดยใช้ภาษา SQL แม้ว่า MySQL เป็นซอฟต์แวร์โอเพนซอร์ส แต่แตกต่างจากซอฟต์แวร์โอเพนซอร์สทั่วไป โดยมีการพัฒนาภายใต้บริษัท MySQL AB ในประเทศสวีเดน โดยจัดการ MySQL ทั้งในแบบที่ให้ใช้ฟรี และแบบที่ใช้ในเชิงธุรกิจ

2.4.1 ความสามารถของ มายเอสคิวแอลเซอเว่อร์

2.4.1.1 CSV เก็บข้อมูลจาก Text ไฟล์โดยอาศัยเครื่องหมาย คอมมา (comma) เป็นตัวแบ่งฟิลด์

2.4.1.2 InnoDB สนับสนุนการทำ ทรานแซคชั่น (transaction) แบบ ACID

2.4.1.3 Memory การจัดเก็บในหน่วยความจำ ใช้เป็นตารางชั่วคราวเพื่อความรวดเร็ว เนื่องจากเก็บไว้ในหน่วยความจำ ทำให้มีความเร็วในการทำงานสูงมาก

2.4.1.4 Archive เหมาะสำหรับการจัดเก็บข้อมูลพวก log file,ข้อมูลที่ไม่ต้องมีการ คิวรี่ (query) หรือใช้บ่อย ๆ เช่น log file เพื่อประโยชน์ในการตรวจสอบย้อนหลัง (Security Audit Information)

2.4.1.5 Federated สำหรับการจัดเก็บแบบปลายทาง (remote server) แทนที่จะเป็นการจัดเก็บแบบ local เหมือนการจัดเก็บ (Storage) แบบอื่น ๆ

2.4.1.6 Merge เป็นการรวม Table หลาย ๆ ตัวให้แสดงผล หรือแก้ไข เสมือนเป็นข้อมูลจาก Table เดียว

2.4.1.7 NDB สำหรับการจัดเก็บแบบ คลัสเตอร์ (cluster) [2]

**ตัวอย่าง**

Summary of MySQL Commands

-- Database-Level

DROP DATABASE *databaseName* -- Delete the database (irrecoverable!)

DROP DATABASE IF EXISTS *databaseName* -- Delete if it exists

CREATE DATABASE *databaseName* -- Create a new database

CREATE DATABASE IF NOT EXISTS *databaseName* -- Create only if it does not exists

SHOW DATABASES -- Show all the databases in this server

USE *databaseName* -- Set the default (current) database

SELECT DATABASE() -- Show the default database

SHOW CREATE DATABASE *databaseName* -- Show the CREATE DATABASE statement

**ตัวอย่าง (ต่อ)**

-- Table-Level

DROP TABLE [IF EXISTS] *tableName*, ...

CREATE TABLE [IF NOT EXISTS] *tableName* (

*columnName columnType columnAttribute*, ...

PRIMARY KEY(*columnName*),

FOREIGN KEY (*columnNmae*) REFERENCES *tableName* (*columnNmae*)

)

SHOW TABLES -- Show all the tables in the default database

DESCRIBE|DESC *tableName* -- Describe the details for a table

ALTER TABLE *tableName* ... -- Modify a table, e.g., ADD COLUMN and DROP COLUMN

ALTER TABLE *tableName* ADD *columnDefinition*

ALTER TABLE *tableName* DROP *columnName*

ALTER TABLE *tableName* ADD FOREIGN KEY (*columnNmae*) REFERENCES *tableName* (*columnNmae*)

ALTER TABLE *tableName* DROP FOREIGN KEY *constraintName*

SHOW CREATE TABLE *tableName* -- Show the CREATE TABLE statement for this *tableName*

-- Row-Level

INSERT INTO *tableName*

VALUES (*column1Value*, *column2Value*,...) -- Insert on all Columns

INSERT INTO *tableName*

VALUES (*column1Value*, *column2Value*,...), ... -- Insert multiple rows

INSERT INTO *tableName* (*column1Name*, ..., *columnNName*)

VALUES (*column1Value*, ..., *columnNValue*) -- Insert on selected Columns

DELETE FROM *tableName* WHERE *criteria*

UPDATE *tableName* SET *columnName* = *expr*, ... WHERE *criteria*

SELECT \* | *column1Name* AS *alias1*, ..., *columnNName* AS *aliasN*

FROM *tableName*

WHERE *criteria*

GROUP BY *columnName*

ORDER BY *columnName* ASC|DESC, ...

HAVING *groupConstraints*

LIMIT *count* | *offset count*

-- Others

SHOW WARNINGS; -- Show the warnings of the previous statement

**2.5 ซีเอสเอส**

CSS คือ ภาษาที่ใช้สำหรับตกแต่งเอกสาร HTML/XHTML ให้มีหน้าตา สีสัน ระยะห่าง พื้นหลัง เส้นขอบและอื่น ๆ ตามที่ต้องการ CSS ย่อมาจาก Cascading Style Sheets มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกกำหนดมาตรฐานโดย W3C เป็นภาษาหนึ่งในการตกแต่งเว็บไซต์ ได้รับความนิยมอย่างแพร่หลาย

2.5.1 ความสามารถของ ซีเอสอส

2.5.1.1 ช่วยให้เนื้อหาภายในเอกสาร HTML มีความเข้าใจได้ง่ายขึ้นและในการแก้ไขเอกสารก็สามารถทำได้ง่ายกว่าเดิม เพราะการใช้ CSS จะช่วยลดการใช้ภาษา HTML ลงได้ในระดับหนึ่ง และแยกระหว่างเนื้อหากับรูปแบบในการแสดงผลได้อย่างชัดเจน

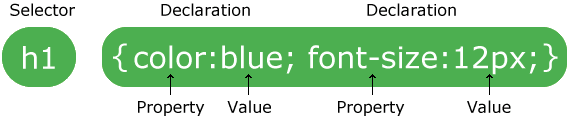
2.5.1.2 ทำให้สามารถดาวน์โหลดไฟล์ได้เร็ว เนื่องจาก code ในเอกสาร HTML ลดลง จึงทำให้ไฟล์มีขนาดเล็กลง

2.5.1.3 สามารถกำหนดรูปแบบการแสดผลจากคำสั่ง style sheet ชุดเดียวกัน ให้มีการแสดงผลในเอกสารแบบเดียวทั้งหน้าหรือในทุก ๆ หน้าได้ ช่วยลดเวลาในการปรับปรุงและทำให้การสร้างเอกสารบนเว็บมีความรวดเร็วยิ่งขึ้น นอกจากนี้ยังสามารถควบคุมการแสดงผล ให้คล้ายหรือเหมือนกันได้ในหลาย Web Browser

2.5.1.4 ช่วยในการกำหนดการแสดงผลในรูปแบบที่มีความเหมาะกับสื่อต่าง ๆ ได้เป็นอย่างดี

2.5.1.5 ทำให้เว็บไซต์มีความเป็นมาตรฐานมากขึ้นและมีความทันสมัย สามารถรองรับการใช้งานในอนาคตได้ดี [2]

**ตัวอย่าง**



**รูปที่ 2.1** รูปตัวอย่างรูปแบบการเขียน css

body {

background-color: lightblue;

}

h1 {

color: white;

text-align: center;

}

p {

font-family: verdana;

font-size: 20px;

}

**2.6 จาวาสคริปต์**

จาวาสคริปต์ คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง จาวา จาวาสคริปต์ เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช่ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปทีละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียลเต็ด (Object Oriented Programming) ที่มีเป้าหมายในการ ออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ต สำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)

2.6.1 ความสามารถของ จาวาสคริปต์

2.6.1.1 จาวาสคริปต์ ทำให้สามารถใช้เขียนโปรแกรมแบบง่ายๆได้ โดยไม่ต้องพึ่งภาษาอื่น

2.6.1.2 จาวาสคริปต์ มีคำสั่งที่ตอบสนองกับผู้ใช้งาน เช่นเมื่อผู้ใช้คลิกที่ปุ่ม หรือ Checkbox ก็สามารถสั่งให้เปิดหน้าใหม่ได้ ทำให้เว็บไซต์ของเรามีปฏิสัมพันธ์กับผู้ใช้งานมากขึ้น นี่คือข้อดีของ จาวาสคริปต์ เลยก็ว่าได้ที่ทำให้เว็บไซต์ดังๆทั้งหลายเช่น Google Map ต่างหันมาใช้

2.6.1.3 จาวาสคริปต์ สามารถเขียนหรือเปลี่ยนแปลง HTML Element ได้ นั่นคือสามารถเปลี่ยนแปลงรูปแบบการแสดงผลของเว็บไซต์ได้ หรือหน้าแสดงเนื้อหาสามารถซ่อนหรือแสดงเนื้อหาได้แบบง่ายๆนั่นเอง

2.6.1.4 จาวาสคริปต์ สามารถใช้ตรวจสอบข้อมูลได้ สังเกตว่าเมื่อเรากรอกข้อมูลบางเว็บไซต์ เช่น Email เมื่อเรากรอกข้อมูลผิดจะมีหน้าต่างฟ้องขึ้นมาว่าเรากรอกผิด หรือลืมกรอกอะไรบางอย่าง เป็นต้น

2.6.1.5 จาวาสคริปต์ สามารถใช้ในการตรวจสอบผู้ใช้ได้เช่น ตรวจสอบว่าผู้ใช้ ใช้ web browser อะไร

2.6.1.6 จาวาสคริปต์สร้าง คุกกี้ (เก็บข้อมูลของผู้ใช้ในคอมพิวเตอร์ของผู้ใช้เอง) ได้

[2]

**ตัวอย่าง**

x = findMax(1, 123, 500, 115, 44, 88);

function findMax() {

var i;

var max = -Infinity;

for (i = 0; i < arguments.length; i++) {

if (arguments[i] > max) {

max = arguments[i];

}

}

return max;

}

**2.7 เจเควียรี**

เจเควียรี (jQuery) คือไลบรารีของโค้ดจาวาสคริปต์ ที่ใช้เป็นส่วนหนึ่งในการสร้างเว็บไซต์ช่วยในการเรียกใช้งานจาวาสคริปต์ให้ง่ายขึ้น เปิดตัวครั้งแรกในงานบาร์แคมป์นิวยอร์ก โดย จอห์น เรซิก (John Resig) เมื่อวันที่ 14 มกราคม พ.ศ. 2549 ตัวโค้ดของเจเควียรีมีลิขสิทธิ์และสัญญาอนุญาตแบบโอเพนซอร์ซ โดยใช้สัญญาอนุญาตของ GFDL และ MIT License

2.7.1 ความสามารถของ เจเควียรี

2.7.1.1 HTML/DOM manipulation

2.7.1.2 CSS manipulation

2.7.1.3 HTML event methods

2.7.1.4 Effects and animations

2.7.1.5 AJAX

2.7.1.6 Utilities [2]

**ตัวอย่าง**

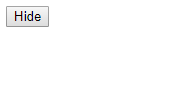
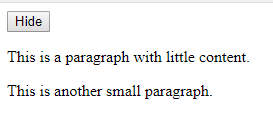
$(document).ready(function(){

$("button").click(function(){

$("p").hide(1000);

});

});



รูปที่ 2.2 เมื่อกดปุ่มจะทำการซ้อนข้อความ และดีเล 1000 ml/sec[3]

$(document).ready(function(){

$("button").click(function(){

$.getJSON("demo\_ajax\_json.js", function(result){

$.each(result, function(i, field){

$("div").append(field + " ");

});

});

});

});

</script>



**รูปที่ 2.3** เมื่อคลิกที่ปุ่ม function จะแสดงคำว่า “John Doe 25” หนึ่งครั้ง[4]

**2.8 ลาลาเวล**

ลาลาเวล(Laravel) คือ PHP Framework ที่เต็มเปลี่ยมไปด้วยพลังที่ทำให้คุณสามารถเขียนโค๊ดที่ดูสะอาดตาและสามารถอ่านและทำความเข้าใจได้โดยง่าย ยังสามารถดาวน์โหลดมาใช้งานได้ฟรี ออกแบบมาเพื่อพัฒนาเว็บแอพพลิเคชั่นในรูปแบบ MVC พัฒนาโดยมีผู้นำทีมคือนาย Taylor Otwell ภายใต้ลิขสิทธิ์ของ MIT และ source code ได้ถูกเก็บไว้บน host ของ GitHub

ลักษณะเด่นของ Laravel

2.8.1 Bundle (สิ่งที่บรรจุมาด้วยกัน) Laravel ได้มาพร้อมกับ แพคเก็จของระบบ ที่ทำให้เราสามารถนำมาใช้กับ เว็บแอพลิเคชั่นของเราได้เลย จึงทำให้เราประหยัดเวลาในการเขียนโค๊ด และ ลดจำนวนการเขียนโค๊ดลงอย่างมาก

2.8.2 Class Autoloading (โหลด Class อัตโนมัติ) ระบบจะทำการโหลด Class ของ PHP มาใช้งานอัตโนมัติ โดยไม่ต้องกำหนดค่าการโหลดใช้งานเอง ในการโหลดระบบจะป้องกันการโหลดในส่วนประกอบ (component) ที่ไม่ใช้งาน และ จะเลือกโหลดเฉพาะส่วนประกอบที่นำมาใช้งานเท่านั้น

2.8.3 View Composer (ส่วนของ View) ส่วนนี้จะเป็นส่วนของโค๊ด(HTML) ที่นำมาเรียงติดต่อกัน และจะทำงาน (run) หลังจากประกอบกันเสร็จเรียบร้อยแล้ว เช่นเราแบ่งส่วน header,container, sidebar, footer เป็นต้น

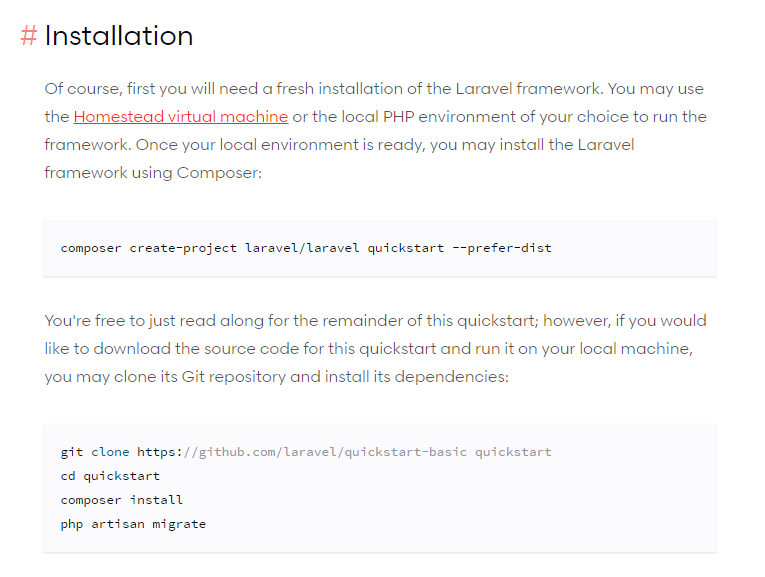
2.8.4 Unit testing (หน่วยทดสอบ) Laravel ยินยอมให้ผู้ใช้งานสามารถสร้าง unit test ขึ้นมาเพื่อทดสอบงานของตัวเองได้โดยผ่าน Artisan utility.

2.8.5 The Eloquent ORM (ชุดคำสั่งในการ Query) เป็นเครื่องมือที่ช่วยในการ Query ข้อมูลต่าง ๆ ในฐานข้อมูล

2.8.6 Reverse Routing (เราท์ติ้งค์) ในส่วนนี้จะทำให้คุณสามารถกำหนดชื่อของ URL เพื่อทีจะชี้ไปยังส่วนต่าง ๆ ตามต้องการ

2.8.7 Restful Controller (กรองชนิดตามการส่งคำขอ) ช่วยให้เราสามารถกรองชนิดการส่งคำร้องขอจากฟอร์มทั้งแบบ Post และ Get

2.8.8 The IoC container (Inversion of Control) เป็นส่วนในการจัดเก็บ Library ภายนอกที่เราจะนำเข้ามาใช้ [2]

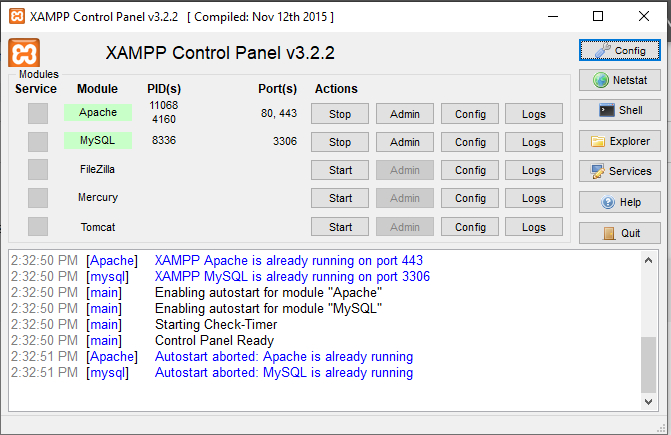


**รูปที่ 2.4** คำสั่งในการใช้ติดตั้ง ลาลาเวลแฟรมเวิร์ค

**2.9 แซมป์**

Xampp คืออะไร เป็นโปรแกรม Apache web server ไว้จำลอง web server เพื่อไว้ทดสอบ สคริปหรือเว็บไซต์ในเครื่องของเรา โดยที่ไม่ต้องเชื่อมต่ออินเตอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใด ๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนาเว็บแอพลิเคชั่นที่เป็นที่นิยม , MySQL ฐานข้อมูล, Apache จะทำหน้าที่เป็นเว็บ เซิร์ฟเวอร์, Perl อีกทั้งยังมาพร้อมกับ OpenSSL , phpMyAdmin (ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล สนับสนุนฐานข้อมูล MySQL และ SQLite โปรแกรม Xampp จะอยู่ในรูปแบบของไฟล์ Zip, tar, 7z หรือ exe โปรแกรม Xampp อยู่ภายใต้ใบอนุญาตของ GNU General Public License แต่บางครั้งอาจจะมีการเปลี่ยนแปลงเรื่องของลิขสิทธิ์ในการใช้งาน จึงควรติดตามและตรวจสอบโปรแกรมด้วย [2]

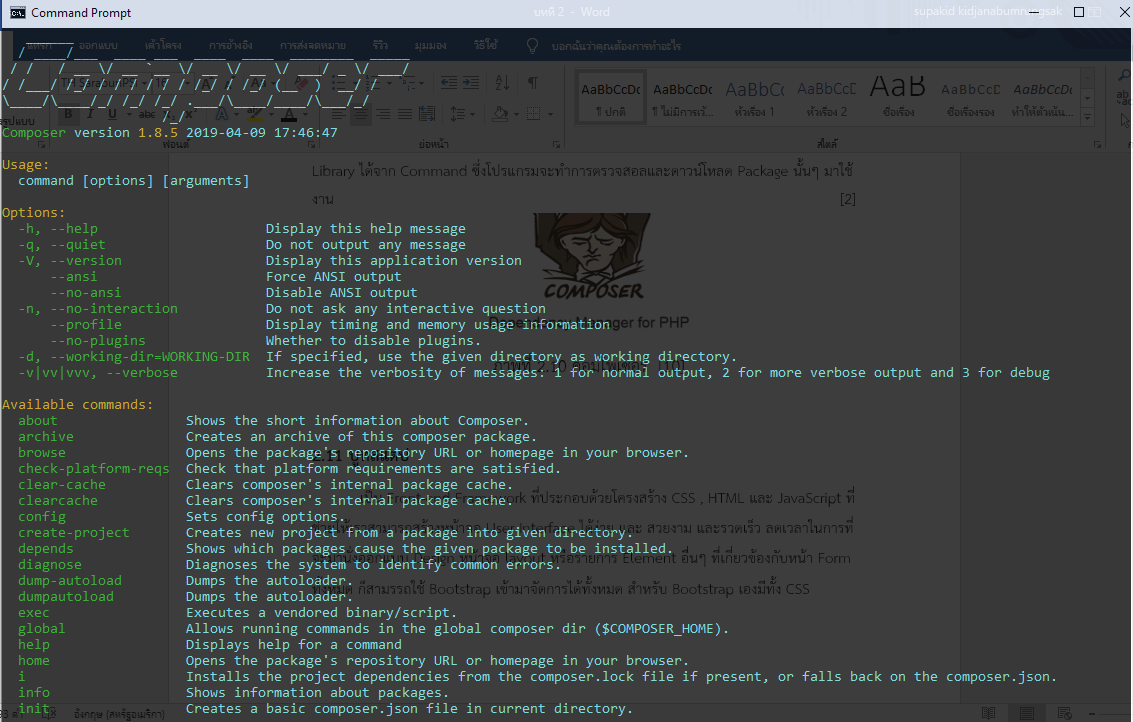
**ตัวอย่าง**



**รูปที่ 2.5 รูปหน้าต่างใช้งาน xampp**

**2.10 คอมโพเซอร์**

Composer คืออะไร การติดตั้งบน PHP Composer เช่น XAMPP , Appserv และอื่น ๆ การพัฒนาโปรแกรม php เรากำลังอยู่ในยุคของการจัดการกับ Package และ Library ต่าง ๆ ที่เป็น Open Source ที่มีอยู่มากมายตามแหล่งต่าง ๆ ให้มีความเป็นระบบเรียบร้อย มีความปลอดภัยในการดาวน์โหลดและ นำมาใช้ เพราะก่อนหน้านี้เราจะเห็นว่าในกรณีที่จะต้องใช้ Library ต่าง ๆ นั้น จะต้องใช้วิธีการดาวน์โหลดไฟล์จากแหล่งนั้นมาใช้งาน ซึ่งจะพบกับความเสี่ยงต่าง ๆ จากการดาวน์โหลด ไม่ว่าจะเป็นความปลอดภัย ไวรัส มัลแวร์ ความน่าเชื่อถือ รวมทั้งไฟล์นั้นใช้งานตรงกับเวอร์ชั่นของ php ที่จะสามารถใช้งานได้หรือไม่ ซึ่งปัญหาเหล่านี้ได้ถูกพัฒนาจนมาถึงปัจจุบัน โดย Composer แนวคิดคือจะรวบรวมและจัดระเบียบ package และ library ต่าง ๆ ให้มาอยู่ในแหล่งเดียวกัน สามารถติดตั้ง Library ได้จาก Command ซึ่งโปรแกรมจะทำการตรวจสอลและดาวน์โหลด Package นั้น ๆ มาใช้งาน [2]



รูปที่ 2.6 รูปหน้าต่างใช้งาน composer

**2.11 บูทสแตป**

บูทสแตป(Bootstrap)เป็น Front-end Framework ที่ประกอบด้วยโครงสร้าง CSS , HTML และ JavaScript ที่ช่วยให้เราสามารถสร้างหน้าจอ User Interface ได้ง่าย และ สวยงาม และรวดเร็ว ลดเวลาในการที่จะมานั่งออกแบบ Design หน้าจอ layout หรือรายการ Element อื่น ๆ ที่เกี่ยวข้องกับหน้า Form ทั้งหมด ก็สามรรถใช้ Bootstrap เข้ามาจัดการได้ทั้งหมด สำหรับ Bootstrap เองมีทั้ง CSS Component และ JavaScript Plugin ที่ทำงานร่วมกับ jQuery ที่สามารถเรียกใช้งานได้มากมาย และที่สำคัญคือ Bootstrap มีการแสดงผลในรูปแบบของ Responsive ซึ่งจะแสดงสภาพแวดล้อมที่เหมาะสมกับอุปกรณ์ที่ทำการเปิดอยู่ในขณะนั้น เช่น PC Desktop , Tablets , Mobile หรืออุปกรณ์อื่น ๆ ที่เกี่ยวข้อง ซึ่งทำให้เรานั้นออกแบบเว็บและเขียนคำสั่งสั่งต่าง ๆ ก็สามารถที่จะรองรับอปุกรณ์ทั้งหมดได้เลย [2]

**ตัวอย่าง**

<div class="jumbotron text-center">

<h1>My First Bootstrap Page</h1>

<p>Resize this responsive page to see the effect!</p>

</div>

<div class="container">

<div class="row">

<div class="col-sm-4">

<h3>Column 1</h3>

<p>Lorem ipsum dolor..</p>

</div>

<div class="col-sm-4">

<h3>Column 2</h3>

<p>Lorem ipsum dolor..</p>

</div>

<div class="col-sm-4">

<h3>Column 3</h3>

<p>Lorem ipsum dolor..</p>

</div>

</div>

</div>

**2.12 เอ็มพีดีเอฟ**

เอ็มพีดีเอฟ(mPDF) คือ หนึ่งใน library PHP โดยประโยชน์ของการใช้ library คือมันเป็นโค้ดที่เขียนสำเร็จรูปมาแล้ว ถูกรวบรวมไว้อย่างเป็นระเบียบ พร้อมใช้งาน และนำมาใช้งานซ้ำได้ โดยส่วนใหญ่จะหมายถึงโค้ดที่ไม่ได้เป็นระบบใหญ่ แต่เป็นโค้ส่วนย่อย ๆ ที่มีหน้าที่ ทำงานในส่วนย่อย ๆ เท่านั้นโดยตัว mPDF นี้เป็น library ที่ใช้สำหรับการช่วยสร้างไฟล์ PDF โดยมีข้อดีคือ การอ่านค่าของ css ได้ค่อนข้างดี ซึ่งมีความคล้ายคลึงกับ FPDF TCPDF [2]

**ตัวอย่าง**

<?php

// Require composer autoload

require\_once \_\_DIR\_\_ . '/vendor/autoload.php';

// Create an instance of the class:

$mpdf = new \Mpdf\Mpdf();

// Write some HTML code:

$mpdf->WriteHTML('Hello World');

// Output a PDF file directly to the browser

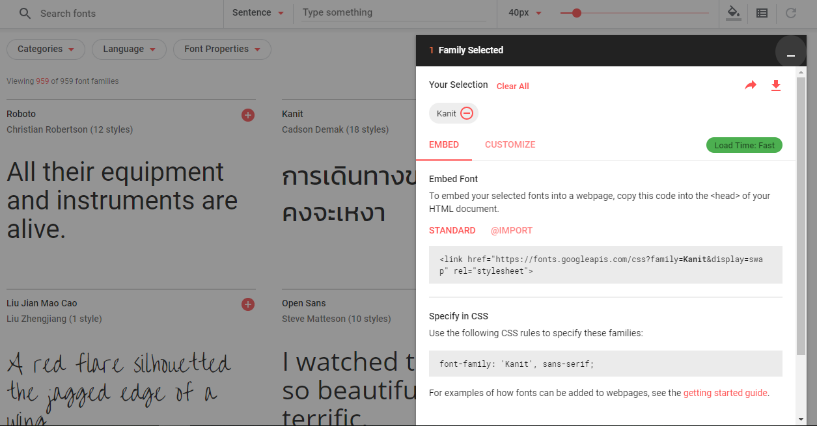
$mpdf->Output();

//command install

$ composer require mpdf/mpdf

**2.13 กูเกิล ฟอนต์**

Google Fonts คือ บริการของ Google ที่มีฟ้อนต์หรือตัวหนังสือแบบต่าง ๆ ให้เราสามารถที่จะเลือกใช้ฟรีได้ แต่ส่วนใหญ่ก็จะเป็นฟ้อนที่สนับสนุนภาษาอังกฤษ คือ เมื่อพิมพ์เป็นภาษาอังกฤษ ตัวหนังสือก็จะสวยตามแบบที่เลือก แต่เมื่อพิมพ์เป็นภาษาไทยก็จะไม่มีการเปลี่ยนแปลงใด ๆ แต่ในระยะหลังก็เริ่มมีฟ้อนต์ไทยเพิ่มมากขึ้น บางธีมก็จะมี Google Fonts มาให้แล้ว แต่สำหรับธีมที่ไม่มี เราก็สามารถที่จะ Import มาใช้ได้ [2]



**รูปที่ 2.7** หน้าเว็ป google font

**2.14 โหนดเจเอส**

เทคโนโลยีฝั่ง Server Side ที่ถูกพัฒนาด้วยภาษา JavaScript

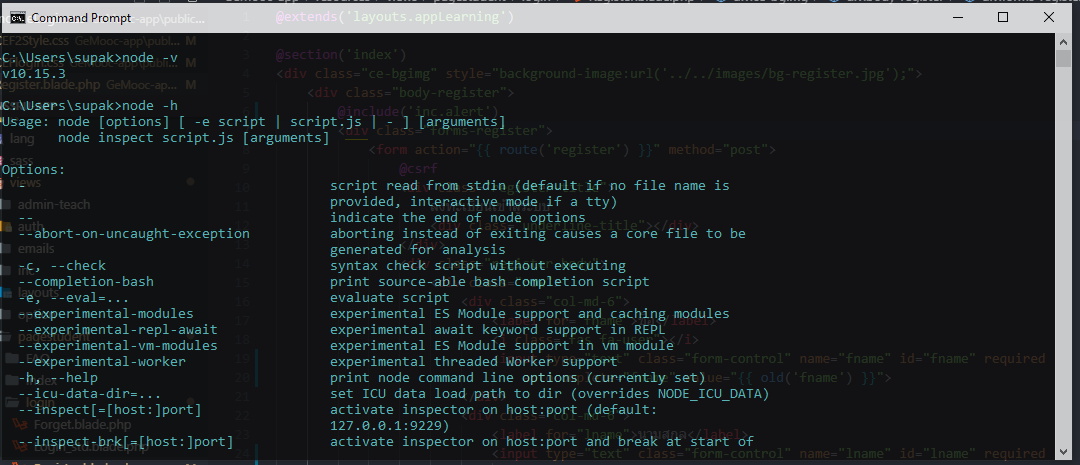
เดิมทีภาษา JavaScript ทำงานฝั่ง Client เป็นหลัก แต่จริง ๆ แล้ว NodeJS เป็น Client หรือ Server ก็ขึ้นอยู่กับจุดประสงค์ของแอพนั้น แต่จุดตั้งต้นเริ่มมาจาก Server Side เป็นหลัก

ผู้สร้าง คือ คนนี้แหละครับ Ryan Dahl เขาว่าคือพ่อมด แฮรี่ ที่มาร่ายเวทย์ด้วยภาษา JavaScript

NodeJS คือ JavaScript มีการ Complied เป็น Byte Code ด้วย V8 Engine ของ Google

Debug ได้ ต่างจาก JavaScript ในยุคแรกsaๆ ทำให้แก้ปัญหาได้ง่ายขึ้นรันได้บนทุกระบบปฏิบัติการยอดนิยมมาพร้อมกับเทคโนโลยีที่เรียกว่า Non - Blocking I/O ปัจจุบันเติบโตเร็วมาก มี Library แล้ว 235,745 Package ข้อมูล ณ วันที่ 2/2/2016 ปัจจุบัน NodeJS ถูกนำมาทำเป็น Web Server , Mobile Hybrid , IOT , Web kit , TVOS ,OS อื่นๆอีกมาก เรียกได้ว่าเข้าถึงได้หลากหลายเทคโนโลยี

**ตัวอย่าง**



**รูปที่ 2.8 ตัวอย่างการใช้งาน node.js**

**บทที่ 3**

**การออกแบบและวิธีการทดสอบ**

การออกแบบระบบเว็บแอพพลิเคชั่นแบบคำร้องสำหรับนักศึกษา จะประกอบด้วยหลายส่วนมาประกอบกันทั้งฝั่งของ client และ server ซึ่งส่วนสำคัญก็คือข้อมูลที่จะนำไปแสดงผลการกระทำต่าง ๆ ของข้อมูลต้องถูกต้อง และไม่ซ้ำซ้อนเพื่อประสิทธิภาพที่ดีของตัวระบบ และนอกจากการออกแบบข้อมูลแล้วยังต้องออกแบบหน้าต่างแสดงผล (User Interface) เพื่อให้สอดคล้องกับการทำงาน รวมไปถึงการทดสอบ และปรับปรุงพัฒนาชิ้นงานจนเหมาะสมและสามารถใช้งานได้จริง ซึ่งการออกแบบ และทดสอบจะมีขั้นตอนดังนี้

1. การออกแบบ
   1. รวบรวมข้อมูล และวิเคราะห์รูปแบบ
      1. วิเคราะห์ข้อมูลจากเอกสารใบคำร้องทั้งหมด
      2. วิเคราะห์รูปแบบการส่งใบคำร้อง
   2. ออกแบบระบบ
      1. Use Case Diagram
      2. Entity Relationship Diagrams
      3. Data Dictionary
   3. ออกแบบส่วนติดต่อผู้ใช้งาน
      1. หน้าต่างแสดงผลผู้ใช้งาน (User Interface)
         1. Login
         2. Client (Student)
         3. Client (Author)
         4. Server (Admin)
2. วิธีการทดสอบ
3. ออกแบบชุดข้อมูลทดสอบ
   1. การออกแบบ
   2. รวบรวมข้อมูล และวิเคราะห์รูปแบบ
   3. วิเคราะห์ข้อมูลจากเอกสารใบคำร้องทั้งหมด

สมาชิกผู้จัดทำโครงงานได้รวบรวมเอกสารข้อมูลของแบบเอกสารคำร้องต่าง ๆ ทั้งหมดของ สำนักวิชาการศึกษาทั่วไปและนวัตกรรมการเรียนรู้อิเล็กทรอนิกส์ มหาวิทยาลัยราชภัฎสวนสุนันทา ออกมาได้ทั้งหมด 7 แบบด้วยกัน มีดังนี้ 1. แบบคำร้องขอตรวจสอบผลการเรียน 2. แบบฟอร์มแจ้งสาเหตุการขาดสอบรายวิชาศึกษาทั่วไป 3. แบบคำร้องแก้ไขผลการเรียน 4. แบบคำร้องขอสอบภายหลัง 5. ใบลาป่วย / ลากิจ 6. ใบคำร้องขอรหัสผ่านเข้าระบบ 7. แบบคำร้องทั่วไป

จากการรวบรวมข้อมูลใบคำร้องทั้งหมดแล้วนำมาวิเคราะห์นั้น ทางสมาชิกผู้จัดทำเห็นได้ว่ารูปแบบของใบคำร้องทั้งหมดจะมีจุดเฉพาะที่เหมือนๆกันในแต่ละเอกสาร ยกตัวอย่างส่วนที่สำคัญที่จะต้องมีในเอกสารได้แก่ ตรามหาลัยทางด้านบนของเอกสาร วันที่กำกับที่ต้องใส่ในทุกแผ่น คำขึ้นต้น และ ปิดท้ายด้วยช่องลายเซ็น เป็นต้น ทางสมาชิกจึงนำข้อมูลที่ซ้ำกันและข้อมูลที่มีในแต่ละใบเอกสารออกจากกันเพื่อหารูปแบบของข้อมูลที่สำคัญในแต่ละเอกสารออกมาเป็นข้อมูลได้ดังนี้

3.1.1.1.1 แบบคำร้องขอตรวจสอบผลการเรียน

3.1.1.1.1.1 วิชา

3.1.1.1.1.2 กลุ่มเรียน

3.1.1.1.2 แบบฟอร์มแจ้งสาเหตุการขาดสอบรายวิชาศึกษาทั่วไป

3.1.1.1.2.1 วิชา

3.1.1.1.2.2 กลุ่มเรียน

3.1.1.1.2.3 ประเภทการสอบ

3.1.1.1.2.4 ประเภทการสอบ

3.1.1.1.2.5 สาเหตุ

3.1.1.1.2.6 เอกสารสำเนาบัตรนักศึกษา

3.1.1.1.3 แบบคำร้องแก้ไขผลการเรียน

3.1.1.1.3.1 วิชา

3.1.1.1.3.2 กลุ่มเรียน

3.1.1.1.3.3 ปีการศึกษา

3.1.1.1.3.4 สาเหตุ

3.1.1.1.3.5 เอกสารสำเนาบัตรนักศึกษา

3.1.1.1.4 แบบคำร้องขอสอบภายหลัง

3.1.1.1.4.1 วิชา

3.1.1.1.4.2 กลุ่มเรียน

3.1.1.1.4.3 ประเภทเว็บไซต์

3.1.1.1.4.4 สาเหตุ

3.1.1.1.5 ใบลาป่วย / ลากิจ

3.1.1.1.5.1 วิชา

3.1.1.1.5.2 กลุ่มเรียน

3.1.1.1.5.3 ประเภทการลา

3.1.1.1.5.4 สาเหตุ

3.1.1.1.5.5 วันที่ลา - วันที่กลับ

3.1.1.1.6 ใบคำร้องขอรหัสผ่านเข้าระบบ

3.1.1.1.6.1 วิชา

3.1.1.1.6.2 กลุ่มเรียน

3.1.1.1.6.3 ประเภทเว็บไซต์

3.1.1.1.6.4 สาเหตุ

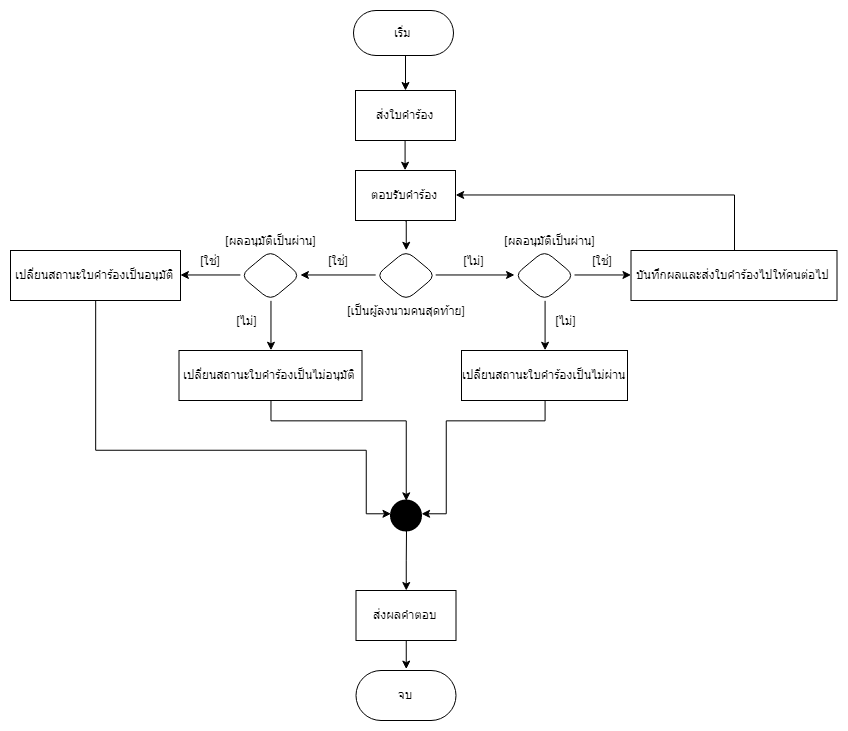
3.1.1.1.7 แบบคำร้องทั่วไป

3.1.1.1.7.1 กลุ่มเรียน

3.1.1.1.7.2 ความประสงค์

* 1. วิเคราะห์รูปแบบการส่งข้อมูลใบคำร้อง

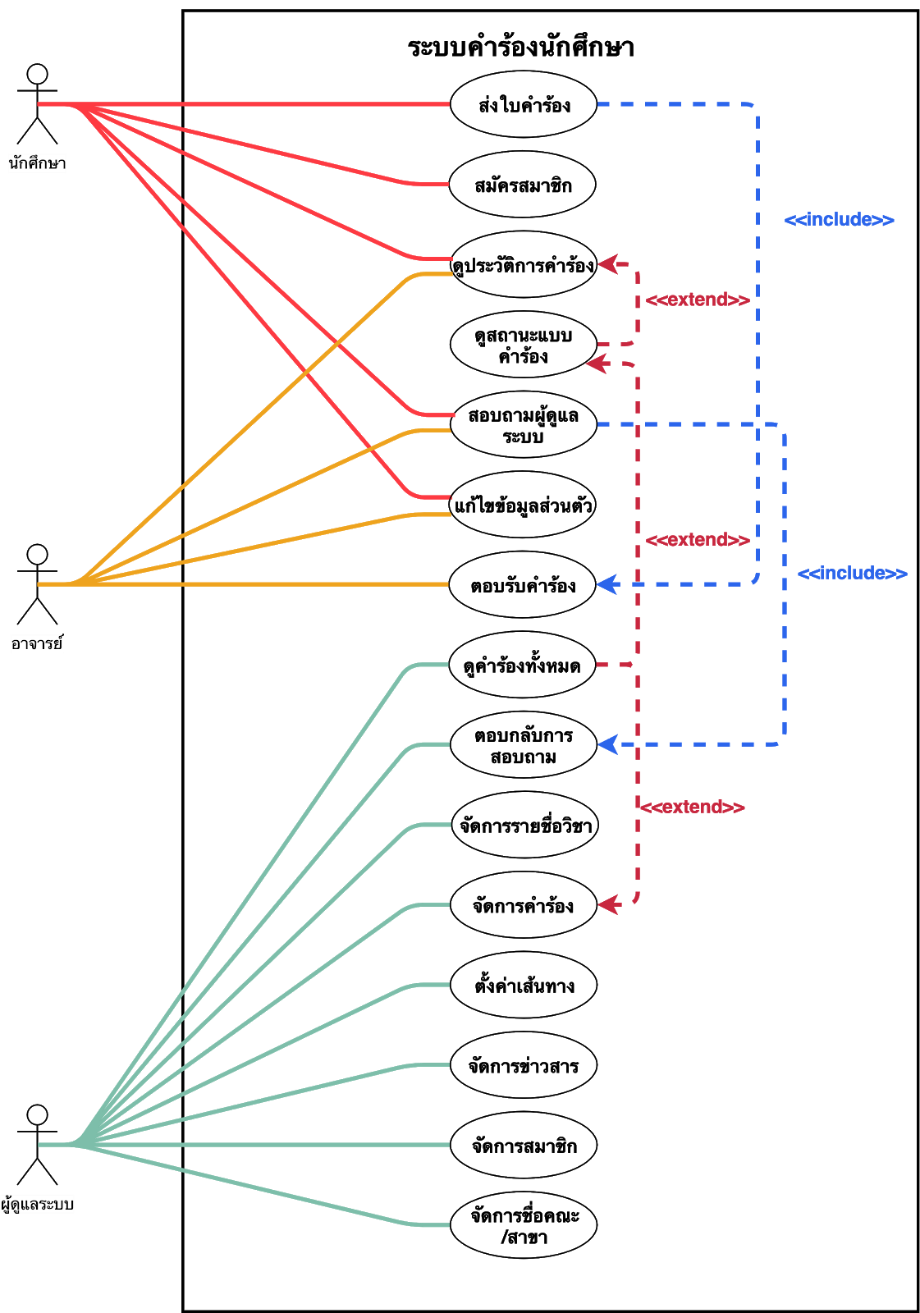
เริ่มต้นจากนักศึกษาไปขอใบคำร้องที่ต้องการมากรอกข้อมูลลงไปที่เอกสาร พอกรอกข้อมูลครบถ้วนก็ส่งเอกสาร ฝ่ายเอกสารก็จะรวบรวมเอกสารและแยกชุดเอกสารออกมาเป็นกลุ่มๆ และส่งเอกสารไปที่ผู้รับผิดชอบด้านนั้น ๆ การตรวจเอกสารจะเป็นการตรวจแบบไล่ไปทีละกลุ่ม ก็คือเอกสารจะผ่านกลุ่มผู้รับผิดชอบในแต่ละกลุ่มนั้น ๆ พอตรวจเสร็จลงนามและผ่านก็จะส่งไปที่กลุ่มผู้อนุมัติกลุ่มต่อไปจนไปถึงผู้อนุมัติและให้เอกสารผ่านก็จะครบกระบวนการ และส่งเอกสารคืนกลับมา ฝ่ายเอกสารก็ติดต่อกลับมาที่ตัวนักศึกษาและมารับเอกสารถือเป็นการจบขั้นตอนการทำงานหนึ่งขั้นตอน แต่ถ้ามีการยกเลิกหรือไม่อนุมัติเอกสารก็จะจบกระบวนการตรวจเช่นกัน ดังรูปตัวอย่างแผนภาพระบบการทำงาน (System Flow Diagram) ต่อไปนี้



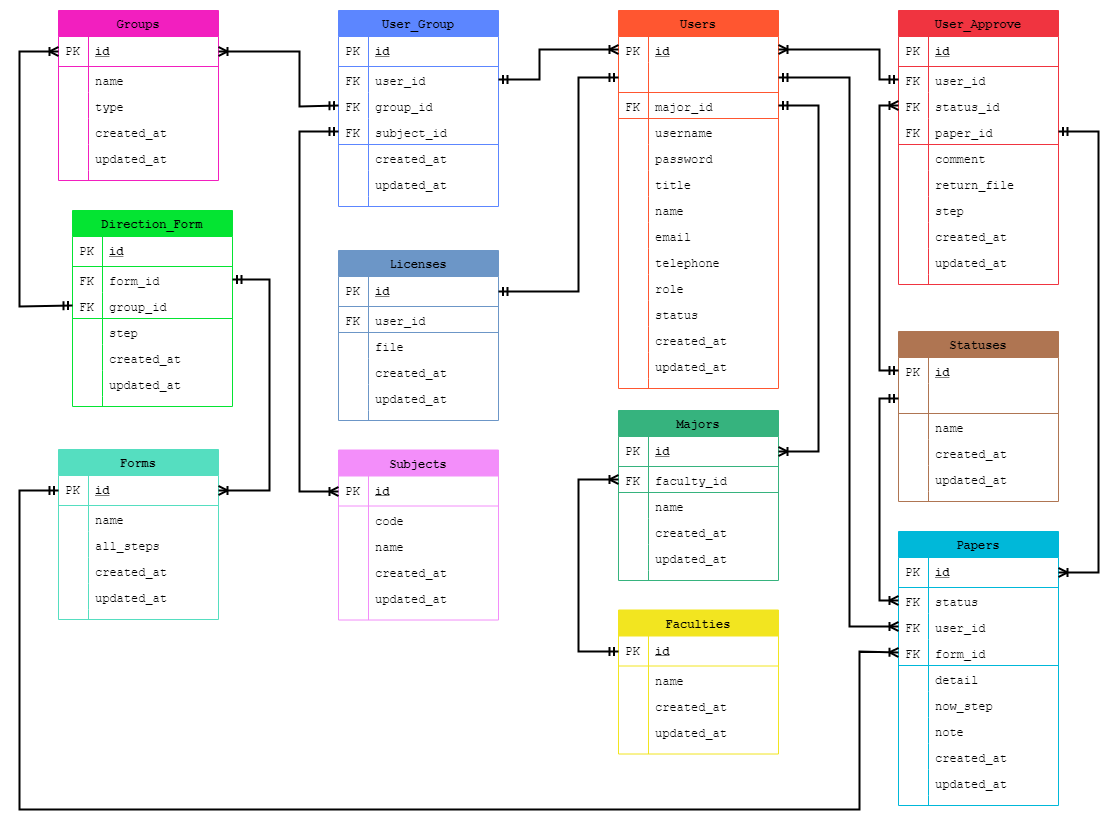
**รูปที่ 3.1** System Flow Diagram รูปแบบการทำงานของระบบส่งเอกสาร [1]

1. ออกแบบระบบ
   1. Use Case Diagram

เมื่อพิจารณารูปแบบการทำงานของตัวระบบแล้วจึงแบ่งฟังก์ชันการทำงานออกได้ 15 ฟังก์ชันดังภาพด้านล่าง แสดงผลออกมาแบบแผนภาพ Use Case Diagram ซึ่งจากแผนภาพจะเห็นว่าจะมีฟังก์ชันบางฟังก์ชันที่มีความสัมพันธ์ต่อกัน เช่น ดูประวัติคำร้องจะ extend ฟังก์ชันการดูสถานะแบบคำร้องออกมา และการดูสถานะแบบคำร้องก็ extend ฟังก์ชันการดูคำร้องทั้งหมดออกมา หรือฟังก์ชันที่ include กันเช่น ฟังก์ชันตอบรับใบคำร้องต้อง include ฟังก์ชันส่งใบคำร้องมาก่อน



**รูปที่ 3.2** แผนภาพ Use Case Diagram ของระบบคำร้องนักศึกษา [2]

* 1. Entity Relationship Diagram

**รูปที่ 3.3** แผนภาพอีอาร์ เบื่อต้น [3]

* 1. Data Dictionary

**ตารางที่ 3.1.1** Data Dictionary ผู้ใช้งาน (User)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Users | id | รหัสผู้ใช้งาน | Bigint(20) | PK |  |
|  | major\_id | รหัสสาขา | Int(5) | FK | Majors |
|  | username | รหัสเข้าสู่ระบบ | Varcher(20) |  |  |
|  | password | รหัสผ่านเข้าสู่ระบบ | Varcher(20) |  |  |
|  | title | คำนำหน้าชื่อ | Varcher(20) |  |  |
|  | name | ชื่อ | Varcher(50) |  |  |
|  | email | อีเมล | Varcher(50) |  |  |
|  | telephone | เบอร์โทรศัพท์ | Varcher(10) |  |  |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
|  | role | สิทธิ์ | Int(5) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.2** Data Dictionary สาขาทั้งหมดที่มี (Major)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Majors | id | รหัสสาขาวิชา | Int(5) | PK |  |
|  | faculty\_id | รหัสคณะ | Int(5) | FK | Faculties |
|  | name | ชื่อสาขาวิชา | Varcher(50) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.3** Data Dictionary คณะทั้งหมดที่มี (Faculties)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Faculties | id | รหัสคณะ | Int(5) | PK |  |
|  | name | ชื่อคณะ | Varcher(50) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.4** Data Dictionary เอกสาร (Papers)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Papers | id | รหัสเอกสาร | Bigint(20) | PK |  |
|  | user\_id | รหัสผู้ใช้งาน | Bigint(20) | FK | Users |
|  | form\_id | รหัสฟอร์ม | Int(5) | FK | Forms |
|  | status | สถานะเอกสาร | Int(5) | FK | Statuses |
|  | detail | รายระเอียดเอกสาร | Text |  |  |
|  | now\_step | ขั้นตอนขณะนี้ | Int(2) |  |  |
|  | note | หมายเหตุ | Text |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.5** Data Dictionary ฟอร์ม (Forms)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Forms | id | รหัสฟอร์ม | Int(5) | PK |  |
|  | name | ชื่อฟอร์ม | Varcher(50) |  |  |
|  | all\_steps | ขั้นตอนทั้งหมด | Int(5) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.6** Data Dictionary เส้นทางเอกสาร (Direction\_Forms)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Direction\_Forms | id | รหัสเส้นทาง | Bigint(20) | PK |  |
|  | form\_id | รหัสฟอร์ม | Int(5) | FK | Forms |
|  | group\_id | รหัสกลุ่ม | Bigint(20) | FK | Groups |
|  | step | ลำดับขั้น | Int(5) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.7** Data Dictionary กลุ่มผู้ตรวจ (Groups)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Groups | id | รหัสกลุ่ม | Bigint(20) | PK |  |
|  | name | ชื่อกลุ่ม | Varcher(50) |  |  |
|  | type | ประเภทกลุ่ม | Int(5) |  |  |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.8** Data Dictionary ผู้ใช้ของกลุ่ม (User\_Group)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| User\_Group | id | รหัสผู้ใช้ของกลุ่ม | Bigint(20) | PK |  |
|  | user\_id | รหัสผู้ใช้งาน | Bigint(20) | FK | Users |
|  | group\_id | รหัสกลุ่ม | Bigint(20) | FK | Groups |
|  | subject\_id | รหัสวิชา | Bigint(20) | FK | Subjects |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.9** Data Dictionary วิชา (Subjects)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Subjects | id | รหัสวิชา | Bigint(20) | PK |  |
|  | code | รหัสประจำวิชา | Varcher(10) |  |  |
|  | name | ชื่อวิชา | Varcher(50) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.10** Data Dictionary การอนุมัติ (User\_Approves)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| User\_Approves | id | รหัสการอนุมัติ | Bigint(20) | PK |  |
|  | user\_id | รหัสผู้ใช้งาน | Bigint(20) | FK | Users |
|  | status\_id | รหัสสถานะ | Int(5) | FK | Statuses |
|  | paper\_id | รหัสเอกสาร | Bigint(20) | FK | Papers |
|  | comment | ความคิดเห็น | Text |  |  |
|  | return\_file | ไฟล์แนบ | Text |  |  |
|  | step | ลำดับขั้น | Int(5) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

**ตารางที่ 3.1.11** Data Dictionary สถานะ (Statuses)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Statuses | id | รหัสสถานะ | Int(5) | PK |  |
|  | name | ชื่อสถานะ | Varcher(50) |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

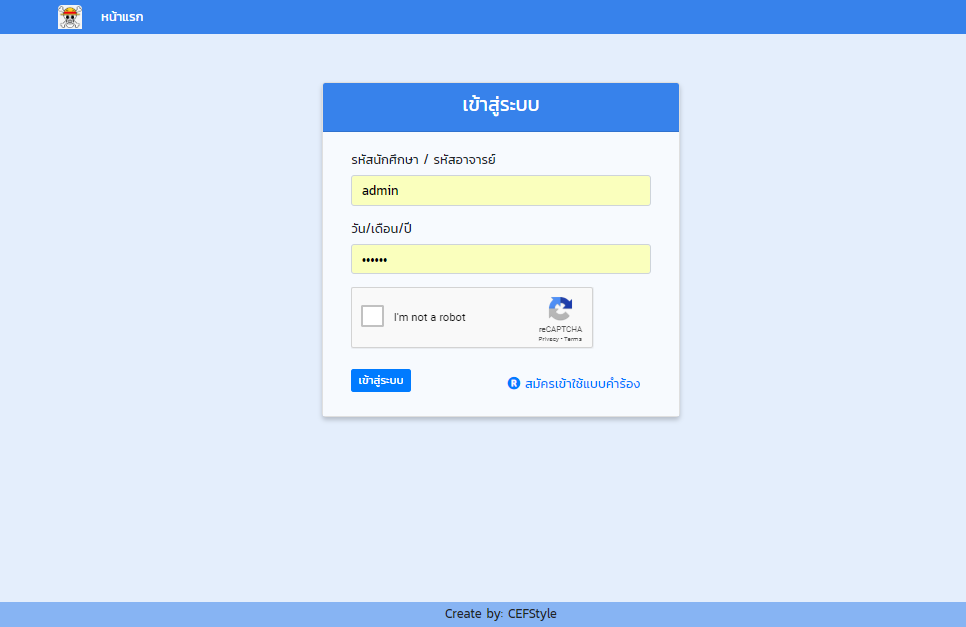
**ตารางที่ 3.1.12** Data Dictionary ลายเซ็น (Licenses)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table Name** | **Attribute Name** | **Description** | **Type(size)** | **Key Type** | **Reference Table** |
| Licenses | id | รหัสลายเซ็น | Bigint(20) | PK |  |
|  | user\_id | รหัสผู้ใช้งาน | Bigint(20) | FK | Users |
|  | file | ตำแหน่งที่เก็บไฟล์ | Text |  |  |
|  | created\_at | วัน-เวลาที่สร้างข่อมูล | Timestamp |  |  |
|  | updated\_at | วัน-เวลาทีแก้ไขข่อมูล | Timestamp |  |  |

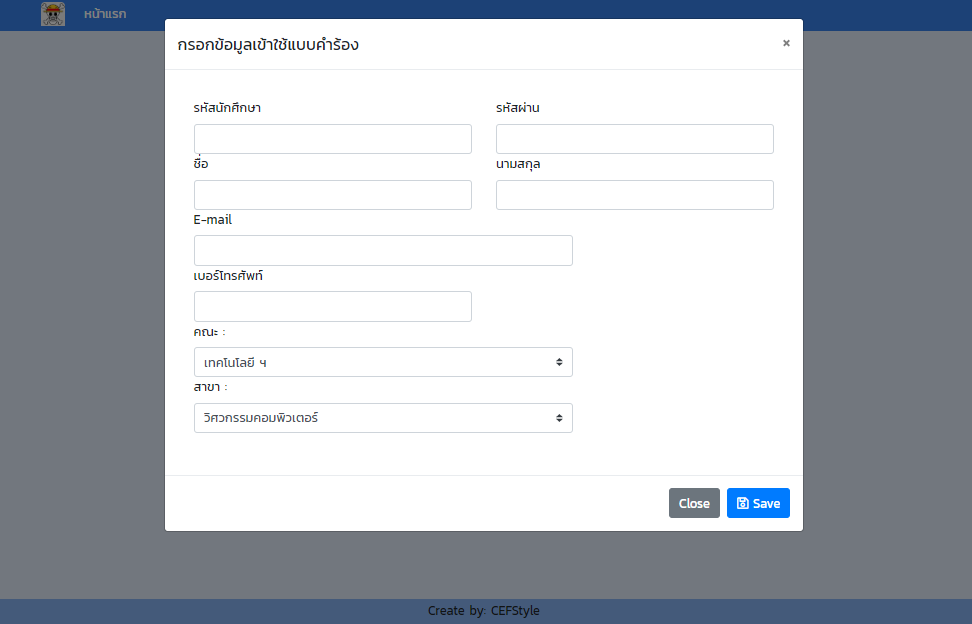
1. ออกแบบส่วนติดต่อผู้ใช้งาน

หลังจากออกแบบฐานข้อมูลเสร็จเรียบร้อยแล้วก็มาถึงส่วนที่เป็นการออกแบบส่วนการแสดงผลออกมาทางเว็บไซส์

* 1. หน้าต่างแสดงผลของผู้ใช้งาน
     1. ออกแบบหน้าจอ Login



**รูปที่ 3.4** ตัวอย่างหน้าจอเข้าสู่ระบบ (Login) [4]



**รูปที่ 3.5** ตัวอย่างหน้าสมัครสมาชิก [5]

* + 1. Client (Student)
       1. ส่วนขอนการแสดงข่าวสาร



**รูปที่ 3.6** ตัวอย่างส่วนของการแสดงข่าวสาร [6]

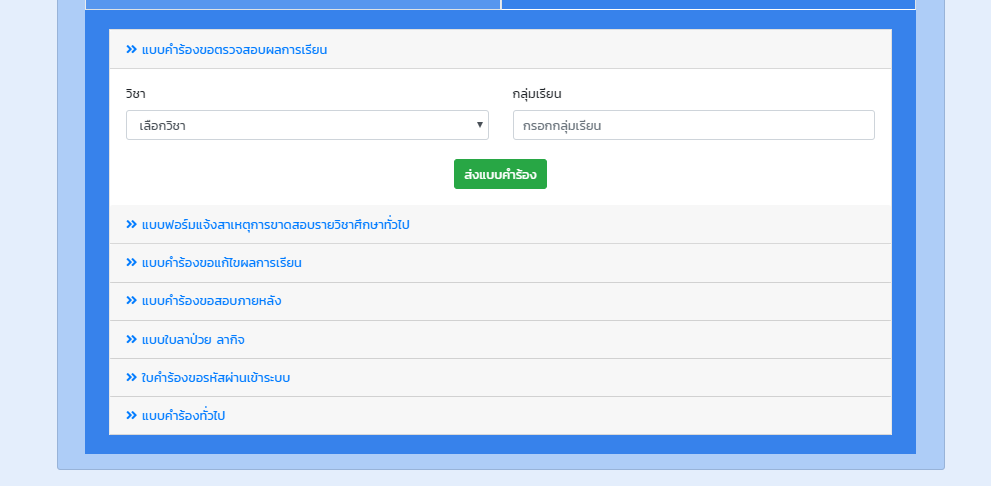
* + - 1. ส่วนประวัติคำร้อง / แบบฟอร์มคำร้อง



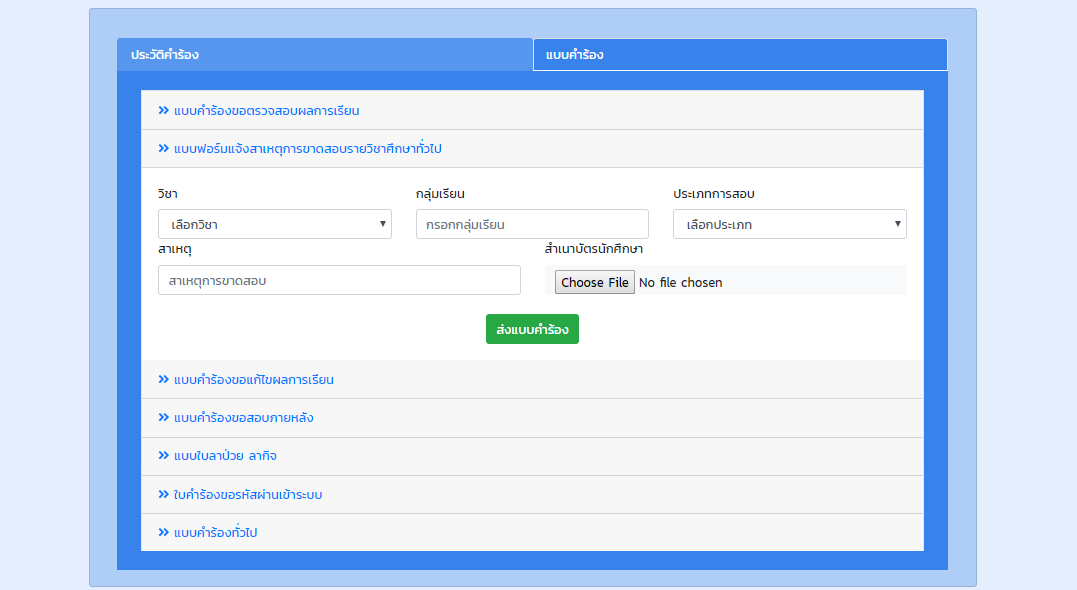
**รูปที่ 3.7** ตัวอย่างส่วนของการแสดงข่าวสาร [7]



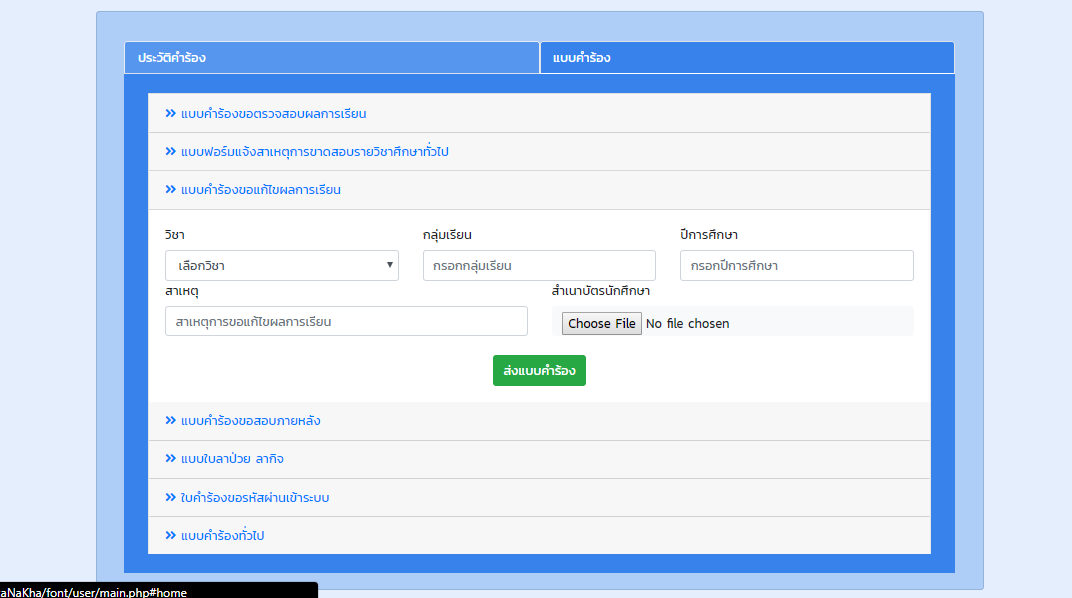
**รูปที่ 3.8** ตัวอย่างส่วนชุดแบบฟอร์มคำร้อง [8]

* + - 1.  ออกแบบช่องอินพุต (Input) ข้อมูลในแต่ละเอกสาร

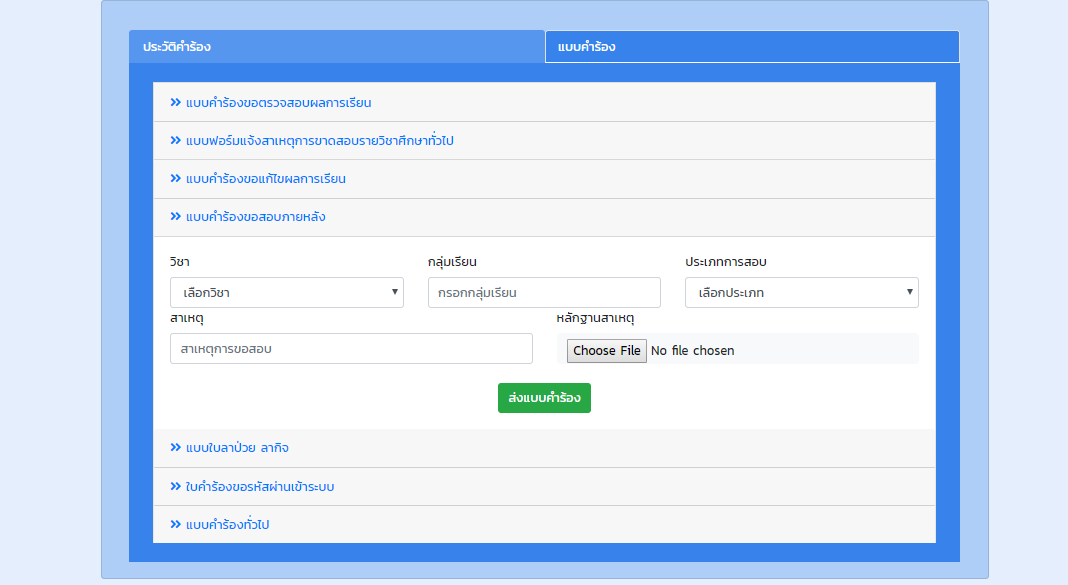
**รูปที่ 3.9** ตัวอย่างช่องกรอกข้อมูล แบบคำร้องขอตรวจสอบผลการเรียน [9]



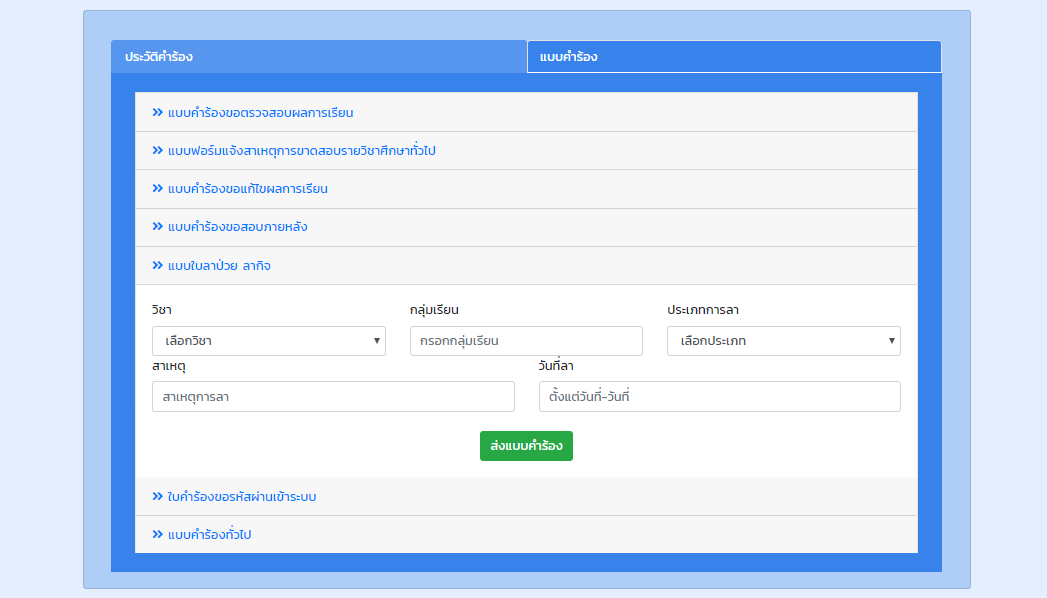
**รูปที่ 3.10** ตัวอย่างช่องกรอกข้อมูล แบบฟอร์มแจ้งสาเหตุการขาดสอบรายวิชาศึกษาทั่วไป [10]



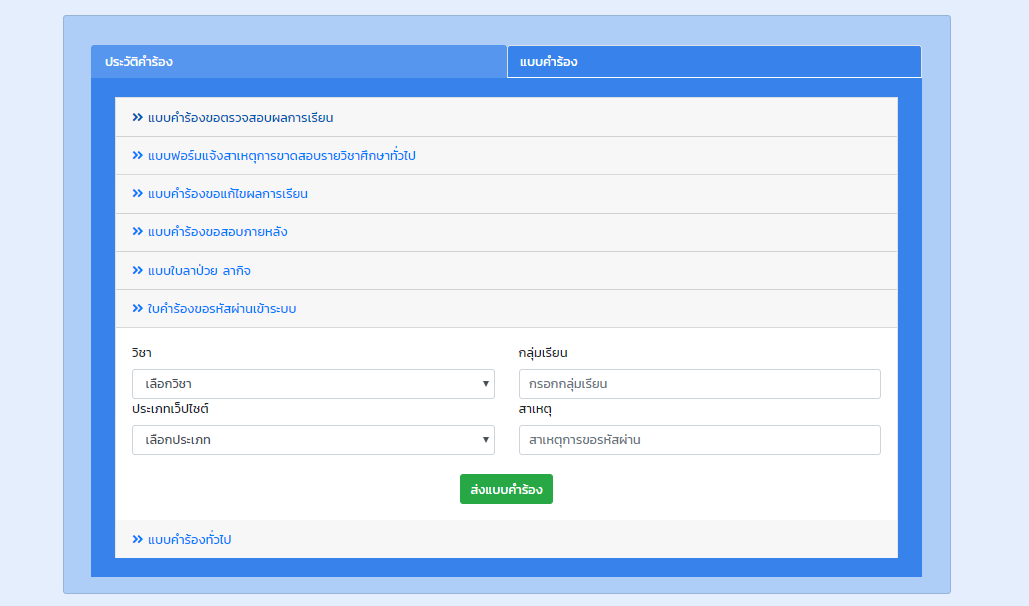
**รูปที่ 3.11** ตัวอย่างช่องกรอกข้อมูล แบบคำร้องแก้ไขผลการเรียน [11]



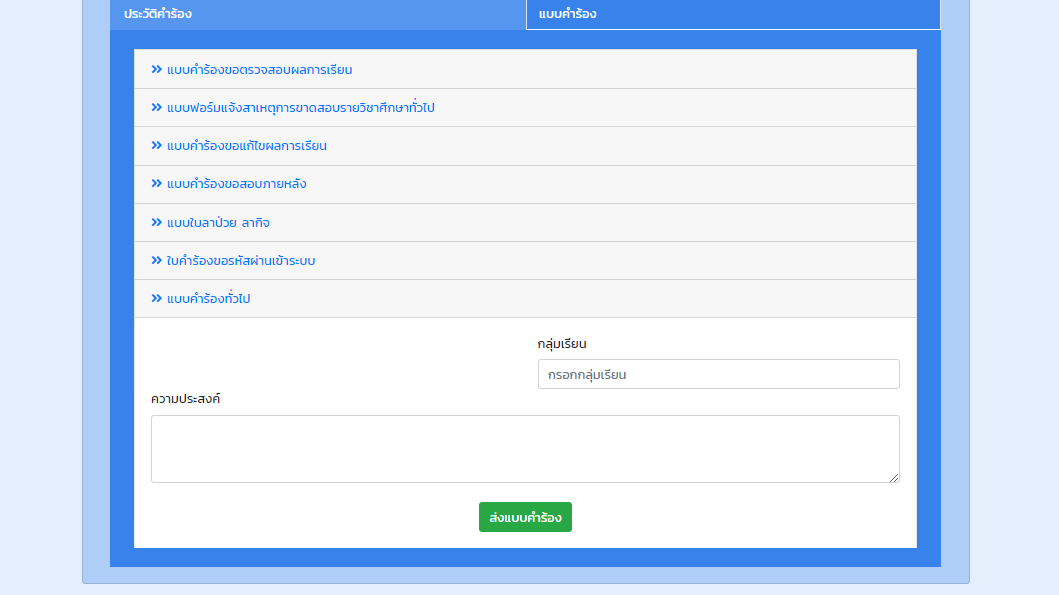
**รูปที่ 3.12** ตัวอย่างช่องกรอกข้อมูล แบบคำร้องขอสอบภายหลัง [12]



**รูปที่ 3.13** ตัวอย่างช่องกรอกข้อมูล ใบลาป่วย / ลากิจ [13]



**รูปที่ 3.14** ตัวอย่างช่องกรอกข้อมูล ใบคำร้องขอรหัสผ่านเข้าสู่ระบบ [14]

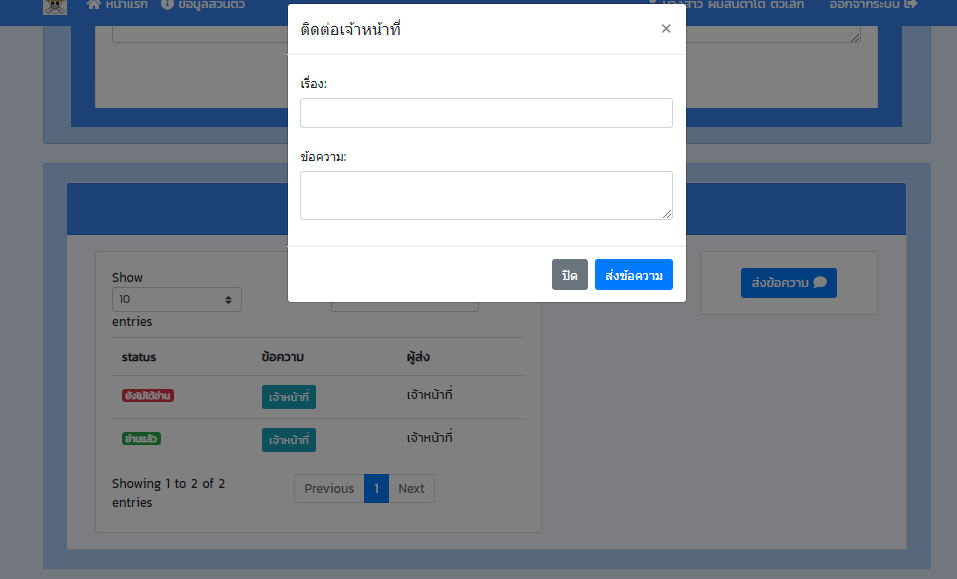


**รูปที่ 3.15** ตัวอย่างช่องกรอกข้อมูล แบบคำร้องทั่วไป [15]

* + - 1. ออกแบบส่วนติดต่อเจ้าหน้าที่

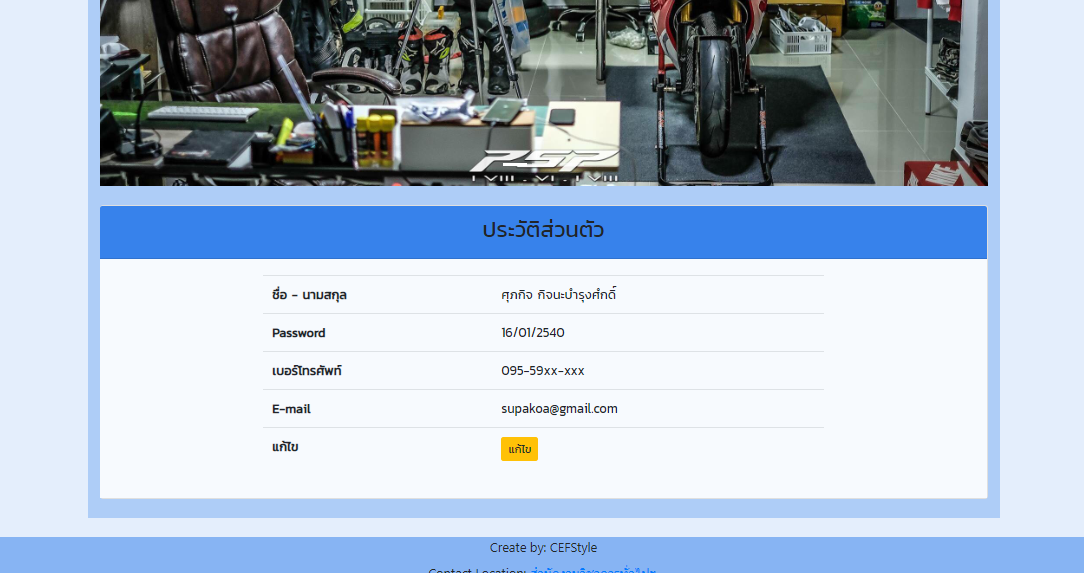


**รูปที่ 3.16** ตัวอย่างช่องติดต่อเจ้าหน้าที่ [16]



**รูปที 3.17** ตัวอย่างช่องติดต่อเจ้าหน้าที่ (เมื่อกดส่งข้อความ) [17]

* + - 1. ออกแบบส่วนของการตั้งค่าแก้ไขข้อมูลส่วนตัว



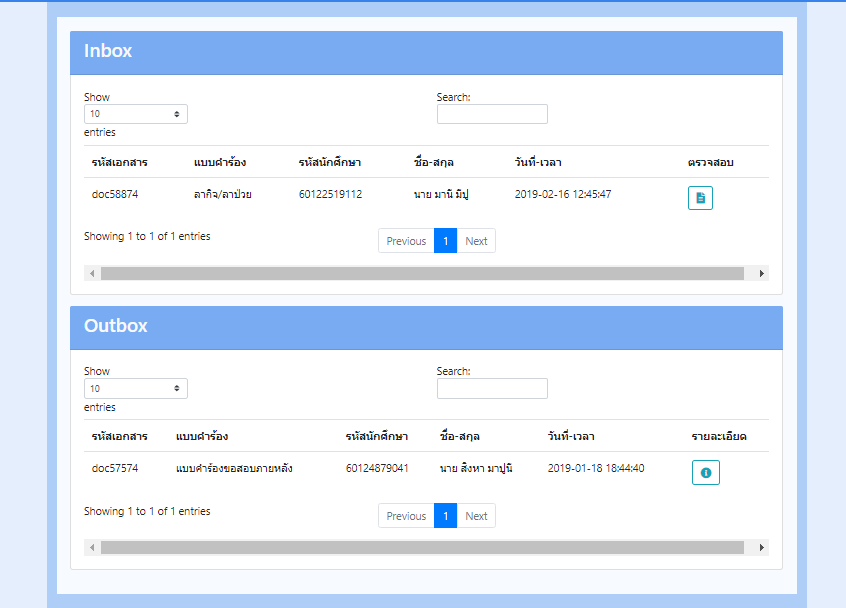
**รูปที 3.18** ตัวอย่างส่วนแสดงประวัติส่วนตัวผู้ใช้งาน [18]



**ภาพที3.19** ตัวอย่างส่วนแสดงประวัติส่วนตัวผู้ใช้งาน [19]

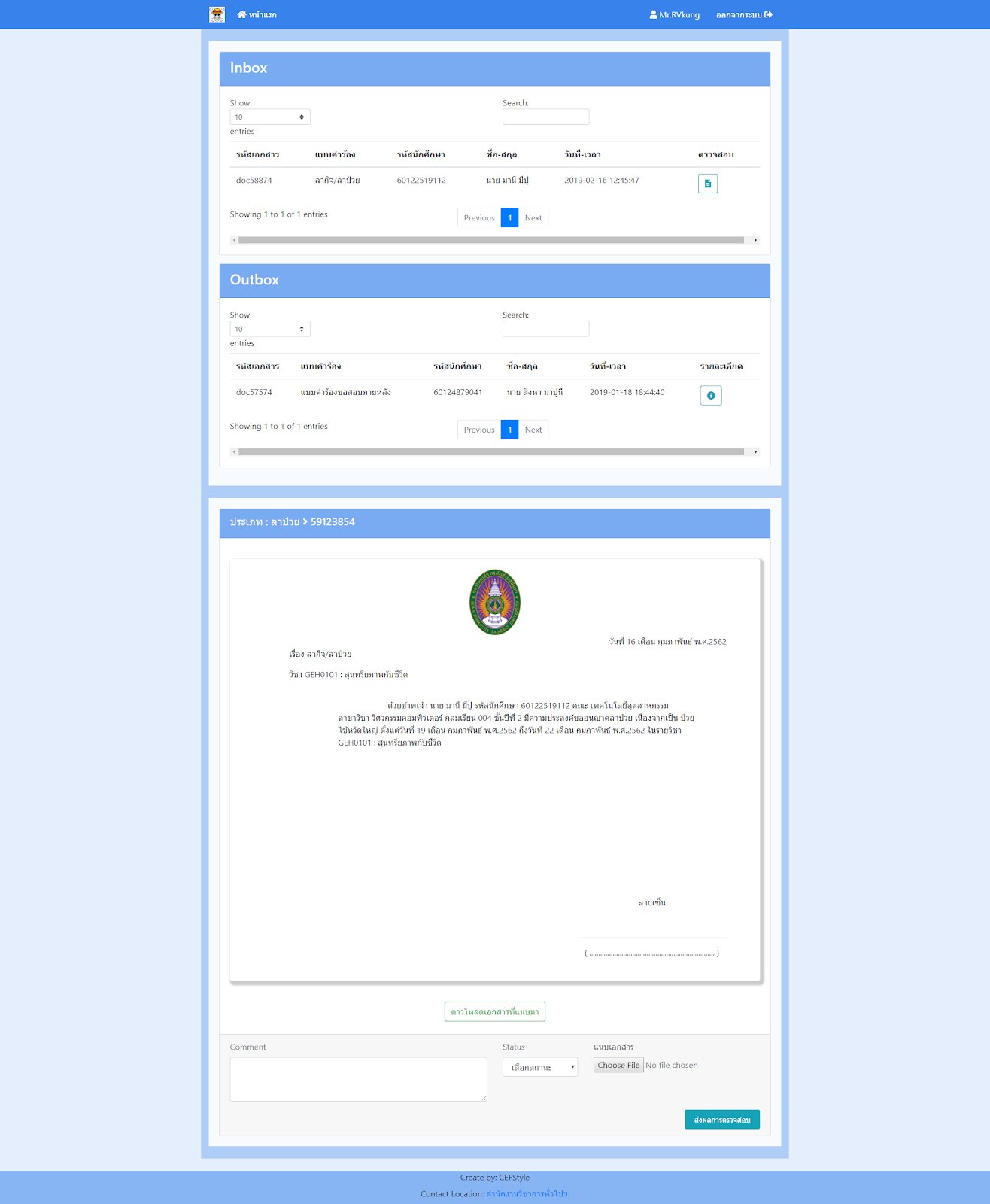
* + 1. Client (Author)

1. ออกแบบส่วนแสดงเอกสารที่ยังไม่ได้ตรวจ / ออกแบบส่วนแสดงเอกสารที่ตรวจแล้ว



**ภาพที3.20** ตัวอย่างส่วนแสดงเอกสารที่ยังไม่ได้ตรวจ / แสดงเอกสารที่ตรวจแล้ว [20]

1. ออกแบบหน้าจอยืนยันผลการตรวจเอกสาร



**ภาพที3.21** ตัวอย่างหน้าจอยืนยันผลการตรวจเอกสาร [21]

* + 1. Server (Admin)
       1. ออกแบบหน้าตอบข้อความ



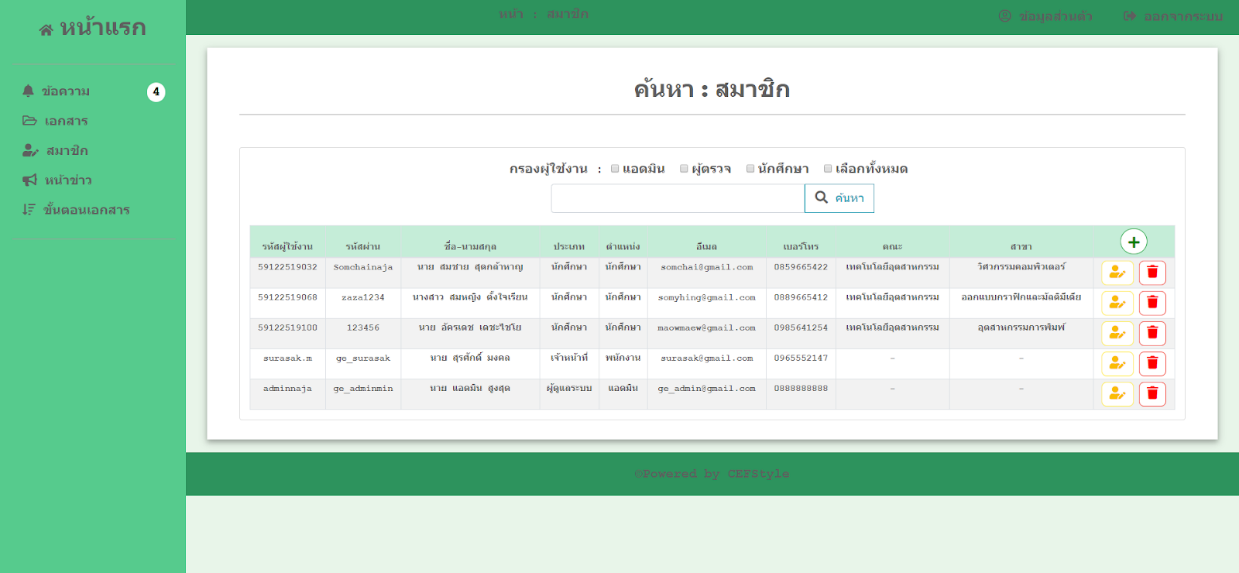
**รูปที่ 3.22** ตัวอย่างหน้าจอข้อความถึงผู้ดูแลระบบ [22]

* + - 1. ออกแบบหน้าตารางข้อมูลเอกสาร



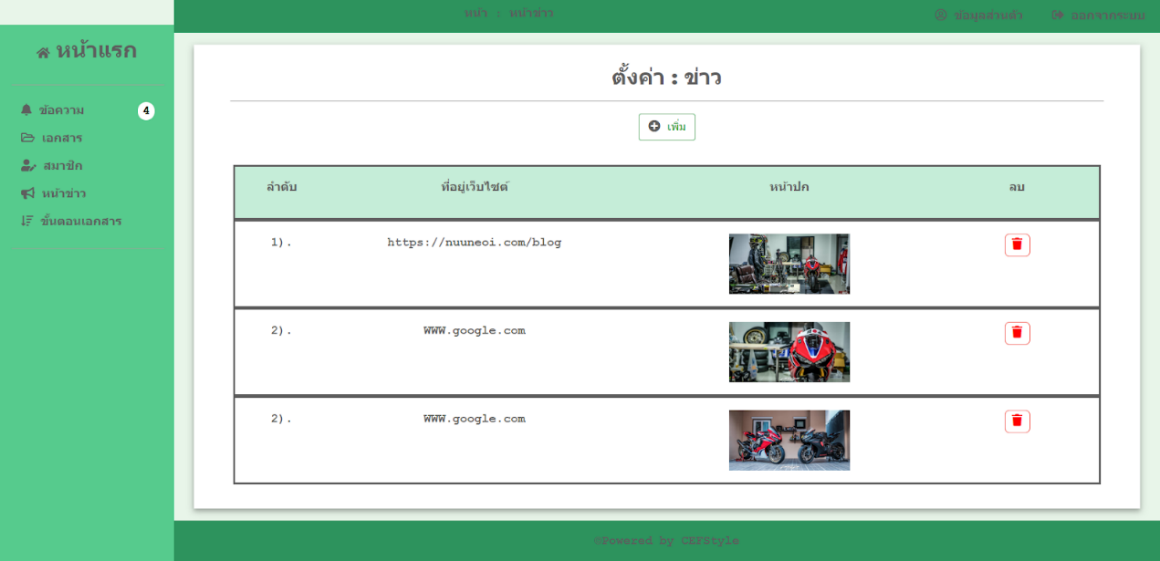
**รูปที่ 3.23** ตัวอย่างหน้าข้อมูลเอกสาร [23]

* + - 1. ออกแบบหน้าจัดการสมาชิก

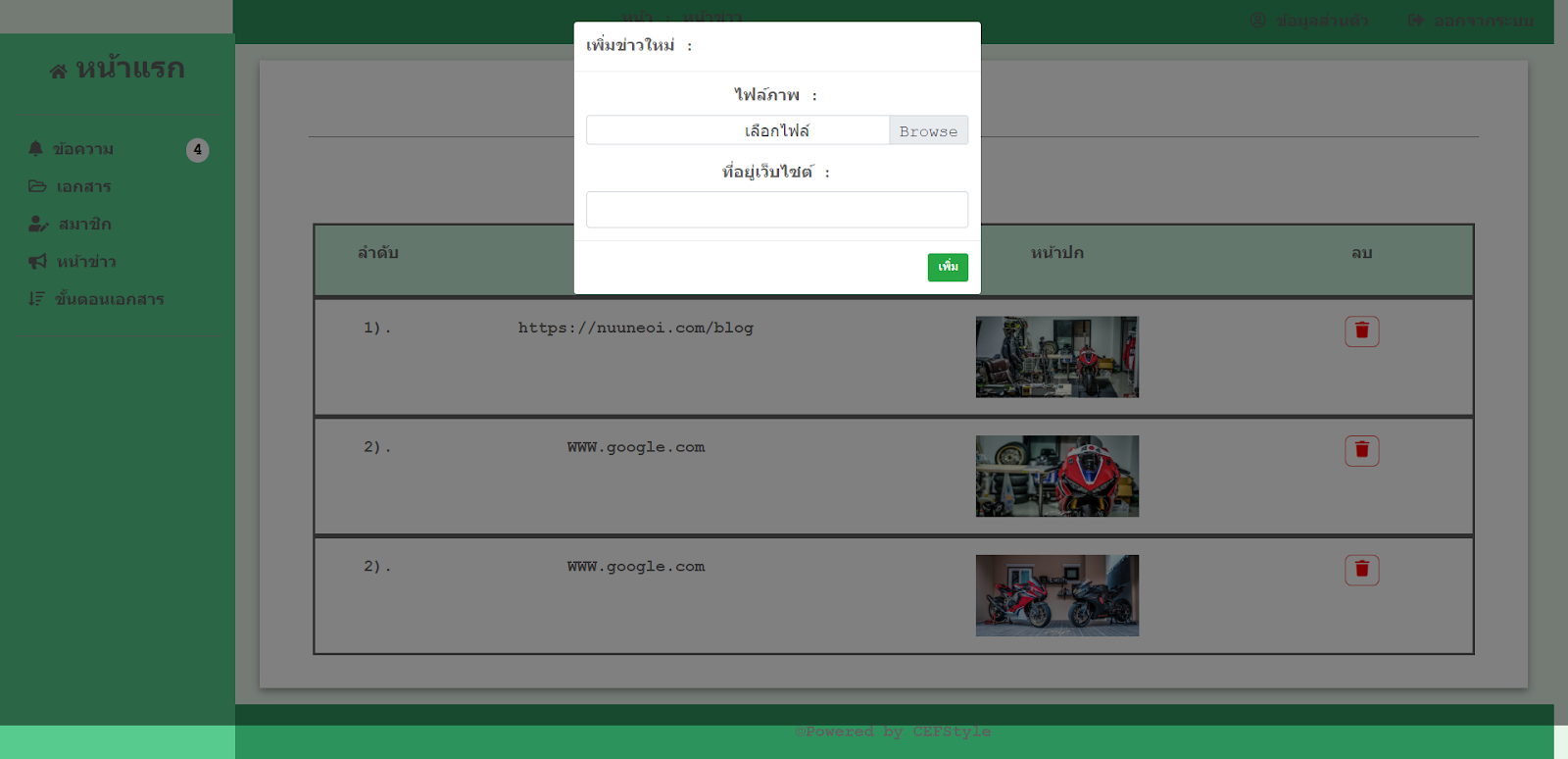


**รูปที่ 3.24** ตัวอย่างหน้าจัดการสมาชิก [24]

* + - 1. ออกแบบหน้าจัดการข้อมูลข่าวสาร



**รูปที่ 3.25** ตัวอย่างหน้าจัดการข้อมูลข่าวสาร [25]

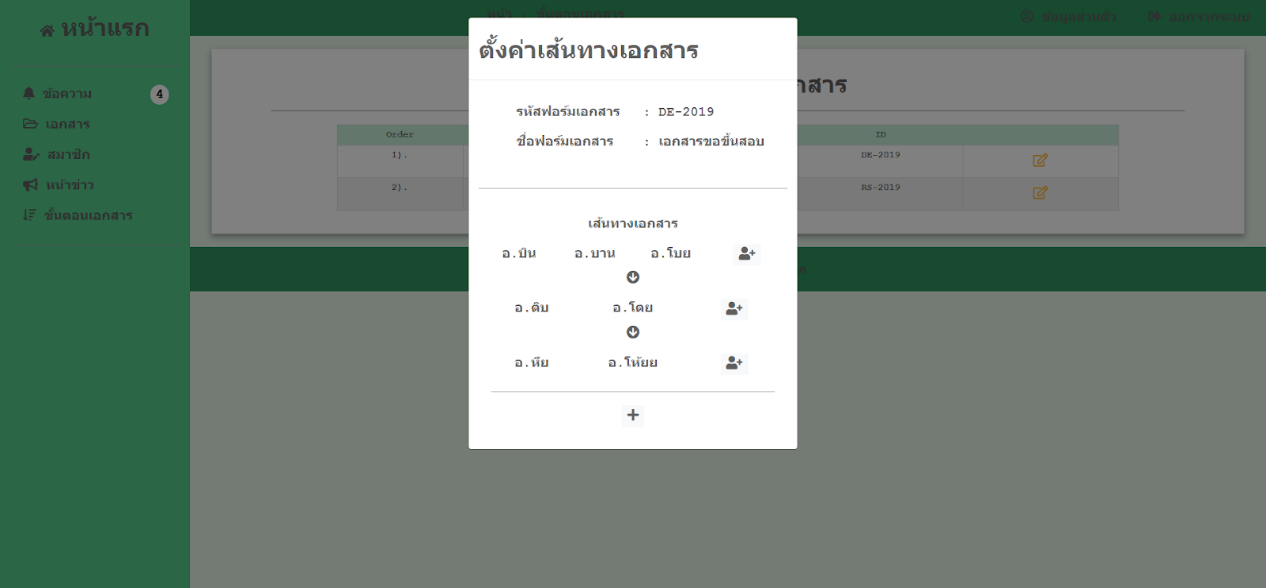


**รูปที่ 3.26** ตัวอย่างส่วนเพิ่มข้อมูล [26]

* + - 1. ออกแบบส่วนกำหนดเส้นทางการตรวจของเอกสาร



**รูปที่ 3.27** ตัวอย่างส่วนกำหนดเส้นทางการตรวจของเอกสาร [27]



**รูปที่ 3.28** ตัวอย่างส่วนกำหนดเส้นทางการตรวจของเอกสาร [28]

1. **วิธีการทดสอบ**
   1. ออกแบบชุดข้อมูลการทดสอบ

**ตารางที่ 3.2.1** การสมัครสมาชิกของผู้ใช้ภายนอก

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **คำอธิบาย** | **Input** | **Output** | **ผลลัพธ์** | **หมายเหตุ** |
| รหัสนักศึกษา | 59122519010 | 59122519010 | รหัสนักศึกษาถูกต้อง | จะต้องเป็นเลขรหัสนักศึกษาเท่านั้น และจะไม่ซ้ำฐานข้อมูล |
| เบอร์โทรศัพท์ | 0930028577 | 0930028577 | เบอร์โทรศัพท์ถูกต้อง | ต้อเป็นเลขเบอร์โทรศัพท์เท่านั้น |
| รหัสผ่าน | drowssap | $2y$10$702gSxHffBtldJETWWxgl.HrLYfEH1Nt72xI8ZvEpqWVZNeWVjlbm | รหัสผ่านถูกต้อง | รหัสผ่านจะถูกเข้ารหัส hash เพื่อไม่ให้เห็นข้อมูลจริง |
| คำนำหน้าชื่อ | นาย | นาย | คำนำหน้าถูกต้อง | จะต้องเป็นภาษาไทย หรือ อังกฤษเท่านั้น |
| ชื่อจริง | ธีระยุทธ์ | ธีระยุทธ์ | ชื่อจริงถูกต้อง | จะต้องเป็นภาษาไทย หรือ อังกฤษเท่านั้น |
| นามสกุล | เติมแต้ม | เติมแต้ม | นามสกุลถูกต้อง | จะต้องเป็นภาษาไทย หรือ อังกฤษเท่านั้น |
| คณะ | เทคโนโลยีอุตสาหกรรม | เทคโนโลยีอุตสาหกรรม | คณะถูกต้อง |  |
| สาขา | วิศวกรรมคอมพิวเตอร์ | วิศวกรรมคอมพิวเตอร์ | สาขาถูกต้อง |  |
| อีเมล์ |  | s59122519010@ssru.ac.th | อีเมล์ถูกต้อง | ข้อมูลอีเมล์จะถูกสร้างมาจากเลขรหัสนักศึกษา |
| ประเภทผู้ใช้งาน |  | นักศึกษา |  | ผู้ที่สมัครสมาชิกจากส่วนนี้มีประเภทผู้ใช้จะเป็นนักศึกษา |

**ตารางที่ 3.2.1** การลงชื่อเข้าสู่ระบบ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **คำอธิบาย** | **Input** | **Output** | **ผลลัพธ์** | **หมายเหตุ** |
| ชื่อผู้ใช้งาน | 59122519010 | 59122519010 | เข้าสู่ระบบไม่ได้ |  |
| รหัสผ่าน | drowssap | $2y$10$702  gSxHffBtld  JETWWxgl.H  rLYfEH1Nt7  2xI8ZvEpq  WVZNeWVjlbm | เข้าสู่ระบบไม่ได้ |  |
| Recaptcha | ไม่กด | ไม่ได้ยืนยันตัวตน | เข้าสู่ระบบไม่ได้ | ไม่มีการยืนยันตัวตนใน Recapcha |

**ตารางที่ 3.2.1** ผลการส่งเอกสาร

|  |  |  |  |
| --- | --- | --- | --- |
| **คำอธิบาย** | **Input** | **ผลลัพธ์** | **หมายเหตุ** |
| ถ้าเอกสารถูกอนุมัติ | เอกสารใบคำร้อง | ใส่สถานะเอกสารว่าผ่านแล้วส่งต่อไปที่ผู้ตรวจต่อไป | ต้องอนุมัติครบทุกกลุ่มก่อนถึงจะส่งเอกสารกลับไปได้ |
| ถ้าเอกสารไม่ถูกอนุมัติ | เอกสารใบคำร้อง | ใส่สถานะเอกสารว่าเป็นผ่านแล้วส่งกลับ | ถ้าเกิดมีการไม่อนุมัติขึ้นในกลุ่มใดๆ เอกสารจะถูกส่งกลับมาที่เจ้าของโดยทันที |