



คู่มือการปฏิบัติงาน (Work Manual)

กระบวนการ การพัฒนาระบบสารสนเทศ
ของ ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร
สำนักงานปลัดกระทรวงเกษตรและสหกรณ์

สารบัญ

๑. วัตถุประสงค์ของการจัดทำคู่มือปฏิบัติงาน	๓
๒. ขอบเขต	๓
๓. คำจำกัดความ.....	๔
๔. หน้าที่ความรับผิดชอบ.....	๕
๕. Work Flow กระบวนการ	๖
๖. ขั้นตอนการปฏิบัติงาน.....	๘
๗. มาตรฐานคุณภาพงาน	๑๔
๘. ระบบติดตามประเมินผล	๑๔
๙. เอกสารอ้างอิง	๑๔
๑๐. แบบฟอร์มที่ใช้.....	๑๔
ภาคผนวก.....	๑๕
ภาคผนวก ๑.....	๑๖
การพัฒนาระบบสารสนเทศ (Web Application) ด้วย Laravel Framework.....	๑๖
ภาคผนวก ๒.....	๔๐
รายชื่อผู้จัดทำ.....	๔๐

คู่มือการปฏิบัติงาน
กระบวนการ พัฒนาระบบสารสนเทศ
ของ ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร

๑. วัตถุประสงค์ของการจัดทำคู่มือปฏิบัติงาน

๑.๑ เพื่อให้ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร มีคู่มือปฏิบัติงานที่ชัดเจนอย่างเป็นลายลักษณ์อักษร ที่แสดงถึงรายละเอียด ขั้นตอน การปฏิบัติงาน การพัฒนาระบบสารสนเทศ และสร้างมาตรฐานการปฏิบัติงานที่มุ่งไปสู่การบริหารคุณภาพทั่วทั้งองค์กรอย่างมีประสิทธิภาพ เกิดผลงานที่ได้มาตรฐานเป็นไปตามเป้าหมาย ได้ผลผลิตที่มีคุณภาพ และบรรลุข้อกำหนดสำคัญที่ของกระบวนการ

๑.๒ เพื่อเป็นหลักฐานแสดงวิธีการพัฒนาระบบสารสนเทศ ที่สามารถถ่ายทอดให้กับผู้เข้ามาปฏิบัติงานใหม่ พัฒนาให้การทำงานเป็นมืออาชีพ และใช้ประกอบการประเมินผลการปฏิบัติงานของบุคลากร รวมทั้งแสดงหรือเผยแพร่ให้กับบุคคลภายนอก ผู้ใช้บริการสามารถเข้าใจ และใช้ประโยชน์จากกระบวนการที่มีอยู่ เพื่อขอรับบริการที่ตรงกับความต้องการ

๑.๓ เพื่อใช้เป็นเครื่องมือในการสร้างมาตรฐานการพัฒนาระบบสารสนเทศของศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร

๒. ขอบเขต

คู่มือกระบวนการพัฒนาระบบสารสนเทศ สำหรับผู้มีหน้าที่รับผิดชอบในการพัฒนาระบบสารสนเทศของศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร ประกอบด้วยขั้นตอน ดังนี้

- ๒.๑ รับเรื่องขอความอนุเคราะห์ในการพัฒนาระบบ
- ๒.๒ ผู้บังคับบัญชาพิจารณา
- ๒.๓ ศึกษาปัญหาขั้นตอนการทำงานของระบบปัจจุบัน และความต้องการระบบใหม่ของหน่วยงานผู้ขอรับบริการ
- ๒.๔ วิเคราะห์และสรุปความต้องการของระบบใหม่
- ๒.๕ ทวนสอบความต้องการระบบใหม่กับผู้ขอรับบริการ เพื่อยืนยันความถูกต้อง
- ๒.๖ ออกแบบระบบใหม่ที่สอดคล้องกับความต้องการของผู้ขอรับบริการ
- ๒.๗ ทวนสอบการออกแบบระบบใหม่กับผู้ขอรับบริการ เพื่อยืนยันความถูกต้อง
- ๒.๘ พัฒนาระบบ
- ๒.๙ ทดสอบโปรแกรม พร้อมปรับแก้
- ๒.๑๐ จัดทำคู่มือการใช้งาน
- ๒.๑๑ ติดตั้งระบบ
- ๒.๑๒ ฝึกอบรมการใช้งานระบบสารสนเทศ
- ๒.๑๓ ติดตาม ประเมินผลการใช้งาน (ศทส.จะดำเนินการเฉพาะระบบสารสนเทศที่มีการใช้งานในภาพรวมของสำนักงานปลัดกระทรวงเกษตรและสหกรณ์)
- ๒.๑๔ สรุปผลการดำเนินงานและประมวลปัญหา อุปสรรค การใช้งาน

๓. คำจำกัดความ

๓.๑ ระบบสารสนเทศ หมายถึง ระบบที่ประกอบด้วยส่วนต่าง ๆ ได้แก่ ระบบคอมพิวเตอร์ ทั้งฮาร์ดแวร์ ซอฟต์แวร์ ระบบเครือข่าย ฐานข้อมูล ผู้พัฒนาระบบ ผู้ใช้ระบบ พนักงานที่เกี่ยวข้อง และผู้เชี่ยวชาญในสาขา ทุกองค์ประกอบนี้ทำงานร่วมกันเพื่อกำหนด รวบรวม จัดเก็บข้อมูล ประมวลผล ข้อมูลเพื่อสร้างสารสนเทศ และส่งผลลัพธ์หรือสารสนเทศที่ได้ให้ผู้ใช้เพื่อช่วยสนับสนุนการทำงาน การตัดสินใจ การวางแผน การบริหาร การควบคุม การวิเคราะห์และติดตามผลการดำเนินงานขององค์กร

๓.๒ ฮาร์ดแวร์ (Hardware) หมายถึง ส่วนที่ประกอบเป็นเครื่องคอมพิวเตอร์ รวมอุปกรณ์ต่อพ่วงต่าง ๆ ที่เกี่ยวข้องกับคอมพิวเตอร์ที่เราสามารถมองเห็นและสัมผัสได้ เช่น ตัวเครื่อง จอภาพ คีย์บอร์ด และเมาส์ เป็นต้น

๓.๓ ซอฟต์แวร์ (Software) หมายถึง ชุดคำสั่งหรือโปรแกรมที่ใช้ควบคุมการทำงานของเครื่องคอมพิวเตอร์ที่เขียนขึ้นโดยภาษาคอมพิวเตอร์จากนักเขียนโปรแกรม (Programmer) เนื่องจากคอมพิวเตอร์นั้นมีการทำงานตามขั้นลำดับ รับภาษาที่ใช้ในการเขียนโปรแกรมนั้นจะมีลักษณะรูปแบบโดยเฉพาะที่จะสามารถทำให้เครื่องคอมพิวเตอร์สามารถเข้าใจได้ เช่น ภาษาเบสิก ภาษาโคบอล ภาษาปาสคาล เป็นต้น

๓.๔ ระบบฐานข้อมูล (Database) หมายถึง ระบบที่รวบรวมข้อมูลต่าง ๆ ที่เกี่ยวข้องกันเข้าไว้ด้วยกันอย่างมีระบบมีความสัมพันธ์ระหว่างข้อมูลต่าง ๆ ที่ชัดเจน ในระบบฐานข้อมูลจะประกอบด้วยแฟ้มข้อมูลหลายแฟ้มที่มีข้อมูล เกี่ยวข้องสัมพันธ์กันเข้าไว้ด้วยกันอย่างเป็นระบบและเปิดโอกาสให้ผู้ใช้สามารถใช้งานและดูแลรักษาป้องกันข้อมูลเหล่านี้ ได้อย่างมีประสิทธิภาพ โดยมีซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูล เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (data base management system) มีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ได้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล

๓.๕ วิเคราะห์ความต้องการ/วิเคราะห์ปัญหา หมายถึง การศึกษาวิถีทางหรือแนวทางการดำเนินงานเพื่อนำไปใช้ในการ ออกแบบและสร้างระบบงานสารสนเทศด้วยเครื่องคอมพิวเตอร์

๓.๖ นักวิเคราะห์ระบบ คือผู้ทำหน้าที่ วิเคราะห์ ออกแบบ และบางครั้งเขียนโปรแกรม นักวิเคราะห์ระบบควรมีความรู้ด้าน เทคนิค มีทักษะด้านการวิเคราะห์ ด้านการจัดการ และการสื่อสารระหว่างบุคคล

๓.๗ การออกแบบระบบ หมายถึง การออกแบบรายละเอียดหรือการออกแบบเชิงกายภาพ (physical design) ประกอบด้วย การออกแบบผลลัพธ์ การออกแบบวิธีการนำข้อมูลเข้า การออกแบบส่วนต่อประสานกับผู้ใช้ การออกแบบแฟ้มข้อมูลและฐานข้อมูล การพิจารณาด้านฮาร์ดแวร์ ซอฟต์แวร์ และอุปกรณ์การสื่อสารที่ต้องใช้ในระบบ

๓.๘ การพัฒนาระบบสารสนเทศ หมายถึง การสร้างระบบงานใหม่หรือปรับเปลี่ยนระบบงานเดิมที่มีอยู่แล้วให้สามารถทำงานเพื่อแก้ปัญหา โดยการกำหนดความต้องการด้านซอฟต์แวร์ หมายถึงนำระบบที่ได้ออกแบบไว้มาทบทวนเพื่อกำหนดการจัดทำซอฟต์แวร์ การออกแบบซอฟต์แวร์ การเขียนโปรแกรม และการทดสอบโปรแกรม

๓.๙ เครื่องมือในการพัฒนาระบบ (Tools) หมายถึง ซอฟต์แวร์ที่ช่วยสร้างหรือวาดแบบจำลองชนิดต่าง ๆ ตรวจสอบความถูกต้องของแบบจำลอง ช่วยสร้างรายงานและแบบฟอร์ม รวมทั้งช่วยสร้างโค้ดโปรแกรมให้อัตโนมัติ

๓.๑๐ วงจรการพัฒนาารระบบ (System Development Life Cycle : SDLC) หมายถึง กระบวนการทางความคิด (Logical Process) ในการพัฒนาระบบสารสนเทศเพื่อแก้ปัญหาและตอบสนอง ความต้องการของผู้ใช้งาน โดยภายในวงจรมันจะแบ่งกระบวนการพัฒนาออกเป็นระยะ (Phase) ได้แก่ ระยะวางแผน (Planning Phase) ระยะการวิเคราะห์ (Analysis Phase) ระยะการออกแบบ (Design Phase) และระยะการสร้างและการพัฒนา (Implementation Phase) โดยแต่ละระยะจะประกอบไปด้วย ขั้นตอน (Steps) ต่าง ๆ ซึ่งแต่ละโครงการพัฒนาระบบจะมีการแบ่งระยะและขั้นตอนในแต่ละระยะแตกต่างกัน โดยในปัจจุบันมีรูปแบบวงจรการพัฒนาารระบบแตกต่างกันออกไปมากมาย อาทิเช่น Waterfall Model, Evolutionary Model, Incremental Model, Incremental Model หรือ Spiral Model เป็นต้น

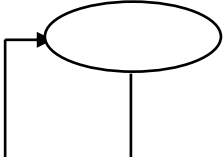
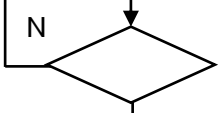
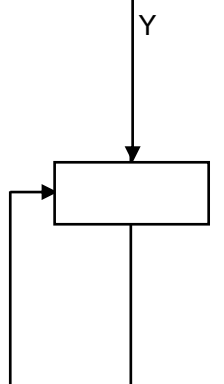
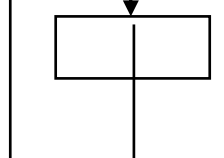
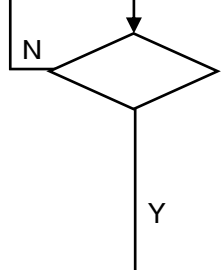
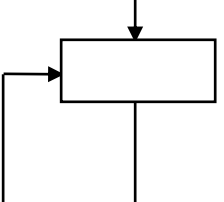
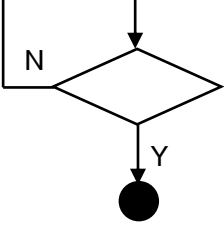
๓.๑๑ เว็บแอปพลิเคชัน (Web Application หรือ Web Based Application) หมายถึง โปรแกรมที่เขียนมาเพื่อทำงานบนเว็บไซต์โดยเรียกใช้งานผ่านทางออนไลน์ทั้งระบบอินเทอร์เน็ต หรืออินทราเน็ต โดยผู้ใช้สามารถใช้งานเว็บแอปพลิเคชันได้ง่ายผ่านเบราว์เซอร์ เช่น Internet Explorer, Firefox, Google Chrome บนอุปกรณ์ต่าง ๆ เช่น คอมพิวเตอร์ แท็บเล็ต สมาร์ทโฟน เป็นต้น ข้อดีของ เว็บแอปพลิเคชัน คือ สามารถใช้งานได้โดยไม่ต้องติดตั้งซอฟต์แวร์บนอุปกรณ์ของผู้ใช้ รวมถึงสามารถเข้ามา ใช้งานโปรแกรมได้ตลอดเวลา อีกทั้งยังสามารถอัปเดตแอปพลิเคชันและข้อมูลต่าง ๆ ได้อย่างสะดวก รวดเร็ว เหมาะอย่างยิ่งกับระบบที่ให้บริการแบบ Real Time หรือระบบที่ต้องใช้ข้อมูลออนไลน์

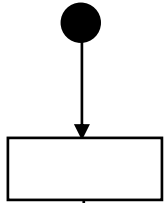
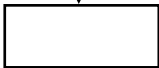
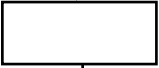
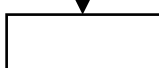
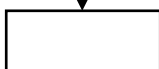
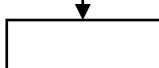
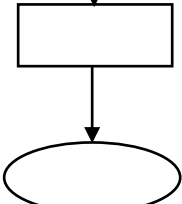
๓.๑๒ โมบายแอปพลิเคชัน (Mobile Application) หมายถึงแอปพลิเคชันที่ช่วยการทำงานของ ผู้ใช้บนอุปกรณ์สื่อสารแบบพกพา เช่น โทรศัพท์มือถือ ซึ่งแอปพลิเคชันเหล่านี้จะทำงานบน ระบบปฏิบัติการ (OS) ที่แตกต่างกันไป ตัวอย่างของระบบปฏิบัติการบนอุปกรณ์เคลื่อนที่ ได้แก่ Windows Mobile ของ Microsoft, iOS ของ Apple, Android OS ของ Google เป็นต้น

๔. หน้าที่ความรับผิดชอบ

ผู้เกี่ยวข้อง	ความรับผิดชอบ
ผู้อำนวยการศูนย์เทคโนโลยีสารสนเทศ และการสื่อสาร	กำกับ ดูแล ให้นโยบาย ข้อเสนอแนะ และสนับสนุน การดำเนินงาน
หัวหน้ากลุ่มระบบงานสารสนเทศและ ภูมิสารสนเทศ	ตรวจสอบ ให้ข้อเสนอแนะในการพัฒนาระบบสารสนเทศ ในภาพรวม
กลุ่มระบบคอมพิวเตอร์ และเครือข่าย	ให้คำแนะนำ ดูแล ให้บริการ จัดสรรพื้นที่ (Server) สำหรับ ติดตั้งระบบสารสนเทศ
กลุ่มระบบงานสารสนเทศและภูมิ สารสนเทศ (ผู้พัฒนาระบบ)	วิเคราะห์ความต้องการ วิเคราะห์ปัญหา ให้สอดคล้องกับ ความต้องการของผู้ใช้งาน และดำเนินการออกแบบ พัฒนา ระบบ จัดทำคู่มือและอบรมผู้ใช้งาน
ผู้ขอรับบริการ (หน่วยงานที่ต้องการ พัฒนาระบบ)	แจ้งความประสงค์ขอรับบริการ โดยให้ข้อมูล เสนอแนะ ระบบงานที่ต้องการแก่ผู้พัฒนาระบบ

๕. Work Flow กระบวนการ

กิจกรรม	ขั้นตอน	ระยะเวลา	หน่วยงานที่เกี่ยวข้อง	จุดควบคุม/ความเสี่ยง
๑. รับเรื่องขอความอนุเคราะห์ในการพัฒนาระบบ		๑ วัน	ผู้ขอรับบริการ, ผอ.ศทส., กรส.	
๒. ผู้บังคับบัญชาพิจารณา		๑-๒ วัน	ผอ.ศทส., กรส.	
๓. ศึกษาปัญหา ขั้นตอนการทำงาน ของระบบปัจจุบัน และความต้องการ ระบบใหม่ของ หน่วยงานผู้ขอรับบริการ		๑๐ วัน	ผู้ขอรับบริการ, กรส.	ความเสี่ยง: ได้รับความต้องการของผู้ขอใช้บริการ และข้อกำหนดนโยบายของหน่วยงานไม่ครบถ้วน
๔. วิเคราะห์และสรุปความต้องการของระบบใหม่		๕ วัน	ผู้ขอรับบริการ, กรส.	
๕. ทวนสอบความต้องการระบบใหม่ กับผู้ขอรับบริการ เพื่อยืนยันความถูกต้อง		๑ วัน	ผู้ขอรับบริการ, กรส.	
๖. ออกแบบระบบใหม่ที่สอดคล้องกับความต้องการของผู้ขอรับบริการ		๑๐ วัน	กรส., กรค.	
๗. ทวนสอบการออกแบบระบบใหม่ กับผู้ขอรับบริการ เพื่อยืนยันความถูกต้อง		๑ วัน	ผู้ขอรับบริการ, กรส.	

กิจกรรม	ขั้นตอน	ระยะเวลา	หน่วยงานที่เกี่ยวข้อง	จุดควบคุม/ความเสี่ยง
๘. พัฒนาระบบ		*	กรส.	
๙. ทดสอบโปรแกรมพร้อมปรับแก้		๑๐ วัน	กรส.	
๑๐. จัดทำคู่มือการใช้งาน		๕ วัน	กรส.	
๑๑. ติดตั้งระบบ		๒ วัน	กรส., กรค., ผู้ขอรับบริการ	
๑๒. ฝึกอบรมการใช้งานระบบสารสนเทศ		๑-๒ วัน/ครั้ง	ผู้ขอรับบริการ, กรส.	
๑๓. ติดตาม ประเมินผลการใช้งาน **		ตามระยะเวลาที่กำหนด (หลังจากมีการใช้ระบบ)	ผู้ขอรับบริการ, กรส.	
๑๔. สรุปผลการดำเนินงานและประมวลปัญหาอุปสรรค การใช้งาน		๕ วัน	กรส.	

หมายเหตุ * ระยะเวลาขึ้นอยู่กับระบบยาก-ง่าย ยุ่งยาก ซับซ้อนหรือไม่ มีรูปแบบข้อมูล โครงสร้างของระบบที่แตกต่างกัน

** ในขั้นตอนที่ ๑๓ ศทส. จะดำเนินการเฉพาะระบบสารสนเทศที่มีการใช้งานในภาพรวมของสำนักงาน ปลัดกระทรวงเกษตรและสหกรณ์

- ผอ.ศทส. หมายถึง ผู้อำนวยการศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร
- กรส. หมายถึง กลุ่มระบบงานสารสนเทศและภูมิสารสนเทศ
- กรค. หมายถึง กลุ่มระบบคอมพิวเตอร์และเครือข่าย

๖. ขั้นตอนการปฏิบัติงาน

รายละเอียดงาน	ขั้นตอนการปฏิบัติงาน	เอกสาร/ระเบียบ/ แบบฟอร์มที่ใช้	ผู้รับผิดชอบ	เงื่อนไขในการปฏิบัติงาน
๑. รับเรื่องขอความอนุเคราะห์ ในการพัฒนาระบบ	๑. ผู้ขอรับบริการส่งบันทึกแจ้ง ความต้องการพัฒนาระบบ หรือ ปรึกษาปัญหาการปฏิบัติงานหรือ ระบบที่ใช้งาน ๒. ผู้อำนวยการศูนย์เทคโนโลยี สารสนเทศและการสื่อสาร และ/ หรือผู้อำนวยการส่วนระบบ สารสนเทศและภูมิสารสนเทศ พิจารณาผู้รับผิดชอบ และ มอบหมายให้ดำเนินการ	๑. บันทึกข้อความขอความ อนุเคราะห์ในการพัฒนาระบบ	๑. ผู้อำนวยการศูนย์เทคโนโลยี สารสนเทศและการสื่อสาร / หัวหน้ากลุ่มระบบงานสารสนเทศ และภูมิสารสนเทศ ๒. สำนัก/กอง ผู้ขอรับบริการ	
๒. ผู้บังคับบัญชาพิจารณา	๑. ผู้บังคับบัญชาพิจารณาให้ ความเห็นในการพัฒนาระบบ ๒. กำหนดผู้รับผิดชอบ และ มอบหมายให้ดำเนินการ		๑. ผู้อำนวยการศูนย์เทคโนโลยี สารสนเทศและการสื่อสาร ๒. กลุ่มระบบงานสารสนเทศ และภูมิสารสนเทศ	
๓. ศึกษาปัญหาขั้นตอนการ ทำงานของระบบปัจจุบัน และความต้องการระบบใหม่ ของหน่วยงานผู้ขอรับบริการ	จัดประชุมเพื่อรวบรวมข้อมูล ความต้องการของผู้ใช้งาน และ ศึกษา วิเคราะห์ สถานภาพของ		๑. กลุ่มระบบงานสารสนเทศ และภูมิสารสนเทศ ๒. สำนัก/กอง ผู้ขอรับบริการ	ผู้ขอรับบริการให้ข้อมูลความ ต้องการได้อย่างถูกต้อง ครบถ้วน

รายละเอียดงาน	ขั้นตอนการปฏิบัติงาน	เอกสาร/ระเบียบ/ แบบฟอร์มที่ใช้	ผู้รับผิดชอบ	เงื่อนไขในการปฏิบัติงาน
	ระบบการทำงาน หรือข้อมูลที่มีอยู่ในปัจจุบัน			
๔. วิเคราะห์และสรุปความต้องการของระบบใหม่	๑. ศึกษาความเป็นไปได้ในการพัฒนาระบบ ๒. สรุปความต้องการของระบบใหม่ ๓. กำหนดกรอบแนวทางหรือขอบเขตในการพัฒนาระบบสารสนเทศตามความต้องการของผู้ใช้งาน		๑. กลุ่มระบบงานสารสนเทศและภูมิสารสนเทศ ๒. สำนัก/กอง ผู้ให้บริการ	
๕. ทวนสอบความต้องการระบบใหม่กับผู้ให้บริการเพื่อยืนยันความถูกต้อง	ติดต่อประสานงานกับผู้ให้บริการ เพื่อประชุมหารือเพื่อสรุปความต้องการของระบบใหม่		๑. กลุ่มระบบงานสารสนเทศและภูมิสารสนเทศ ๒. สำนัก/กอง ผู้ให้บริการ	
๖. ออกแบบระบบใหม่ที่สอดคล้องกับความต้องการของผู้ให้บริการ โดยออกแบบขั้นตอนความสามารถในการทำงานของโปรแกรมหรือระบบที่จะพัฒนา (Input / Output	๑. รวบรวมความต้องการในระบบใหม่จากผู้ใช้นำมาวิเคราะห์โดยใช้เครื่องมือชนิดต่าง ๆ ตามความเหมาะสม เช่น Data Dictionary, Data Flow Diagram, Process Specification, Data Model,		๑. กลุ่มระบบงานสารสนเทศและภูมิสารสนเทศ ๒. กลุ่มระบบคอมพิวเตอร์และเครือข่าย	รายละเอียด ขั้นตอนต่าง ๆ เป็นไปตามหลักการออกแบบและพัฒนาระบบสารสนเทศ

รายละเอียดงาน	ขั้นตอนการปฏิบัติงาน	เอกสาร/ระเบียบ/ แบบฟอร์มที่ใช้	ผู้รับผิดชอบ	เงื่อนไขในการปฏิบัติงาน
<p>Process) โครงสร้างระบบฐานข้อมูล รวมทั้งฮาร์ดแวร์และซอฟต์แวร์ การแสดงผลการนำเสนอข้อมูล (Data Display)</p>	<p>System Model, Prototype, System Flowcharts, Entity Relationship Diagram: E-R Diagram</p> <p>๒. การออกแบบเชิงกายภาพ (Physical Design) หรือเชิงเทคนิค โดยระบุถึงคุณลักษณะของอุปกรณ์ที่จะนำมาใช้ เทคโนโลยี โปรแกรมภาษาที่จะนำมาทำการเขียนโปรแกรมฐานข้อมูล และเครือข่ายที่เหมาะสมกับระบบซึ่งมีระดับการออกแบบ ดังนี้</p> <p>๒.๑ ออกแบบฐานข้อมูลในระดับ Physical</p> <p>๒.๒ ออกแบบ Application</p> <p>๒.๓ ออกแบบเชิงตรรกะ (Logical Design) โดยการออกแบบลักษณะการทำงานของระบบกำหนดถึงลักษณะของรูปแบบ รายงานที่เกิดจากการทำงานของระบบ ลักษณะของ</p>			

รายละเอียดงาน	ขั้นตอนการปฏิบัติงาน	เอกสาร/ระเบียบ/ แบบฟอร์มที่ใช้	ผู้รับผิดชอบ	เงื่อนไขในการปฏิบัติงาน
	<p>การนำข้อมูลเข้าสู่ระบบ และผลลัพธ์ที่ได้จากระบบ ซึ่งจะเลือกใช้การนำเสนอรูปแบบของรายงาน และลักษณะของจอภาพ</p> <p>๓. การกำหนดสิทธิ์การเข้าใช้งานระบบ</p> <p>๔. จัดทำเอกสาร System Design Specification เพื่อให้ผู้พัฒนาสามารถใช้เป็นแบบในการเขียนโปรแกรม</p>			
๗. ทวนสอบการออกแบบระบบใหม่กับผู้ให้บริการเพื่อยืนยันความถูกต้อง	ติดต่อประสานงานกับผู้ให้บริการ เพื่อประชุมหารือสรุปแนวทางการพัฒนาระบบ		<p>๑. กลุ่มระบบงานสารสนเทศ และภูมิสารสนเทศ</p> <p>๒. สำนัก/กอง ผู้ให้บริการ</p>	
๘. พัฒนาระบบ (เขียนโปรแกรมเพื่อให้เป็นไปตามคุณลักษณะและรูปแบบที่ได้ออกแบบไว้)	<p>๑.เขียนชุดคำสั่ง (Coding) / พัฒนาระบบงานให้ได้ตรงตามการออกแบบไว้</p> <p>๒.เชื่อมโยงระบบฐานข้อมูล</p> <p>๓.จัดทำเอกสารวิเคราะห์และพัฒนาควบคู่ไปด้วยเพื่อใช้ในการตรวจสอบและปรับปรุงภายหลัง</p>		กลุ่มระบบงานสารสนเทศและภูมิสารสนเทศ (ผู้พัฒนาระบบ)	การพัฒนาระบบเป็นไปตามที่ออกแบบไว้

รายละเอียดงาน	ขั้นตอนการปฏิบัติงาน	เอกสาร/ระเบียบ/ แบบฟอร์มที่ใช้	ผู้รับผิดชอบ	เงื่อนไขในการปฏิบัติงาน
๙. การทดสอบระบบ เพื่อตรวจสอบหาข้อผิดพลาดของระบบ และดำเนินการแก้ไข	๑. ทำการทดสอบโปรแกรมตรวจสอบหาข้อผิดพลาดของโปรแกรมที่พัฒนา ๒. หากพบข้อบกพร่องไม่เป็นไปตามที่ได้ออกแบบไว้ให้ดำเนินการปรับปรุงแก้ไขและทดสอบซ้ำเพื่อให้ได้ระบบงานและการใช้งานที่ถูกต้องสมบูรณ์		กลุ่มระบบงานสารสนเทศและ ภูมิสารสนเทศ (ผู้พัฒนาระบบ)	ชุดคำสั่งมีความถูกต้อง และการพัฒนาระบบเป็นไปตามที่ออกแบบไว้
๑๐. จัดทำคู่มือการใช้งาน	๑.จัดทำคู่มือสำหรับผู้ดูแลระบบ (System/Programmer manual) ๒.จัดทำคู่มือการใช้งานระบบ(User manual)		กลุ่มระบบงานสารสนเทศและ ภูมิสารสนเทศ (ผู้พัฒนาระบบ)	ผู้ใช้งาน (User) สามารถใช้คู่มือประกอบการปฏิบัติงานได้
๑๑. ติดตั้งระบบ และส่งมอบระบบให้ผู้ขอรับบริการ	๑.ติดตั้งระบบ พร้อมอุปกรณ์ประกอบ (ถ้ามี) ให้กับผู้ใช้งาน ๒.ทำบันทึกส่งมอบระบบให้สำนัก/กอง ผู้ขอรับบริการ	บันทึกข้อความการส่งมอบระบบ	๑. กลุ่มระบบงานสารสนเทศและภูมิสารสนเทศ ๒. กลุ่มระบบคอมพิวเตอร์และเครือข่าย ๓.สำนัก/กอง ผู้ขอรับบริการ	ระบบและอุปกรณ์ต่าง ๆ สามารถใช้งานได้

รายละเอียดงาน	ขั้นตอนการปฏิบัติงาน	เอกสาร/ระเบียบ/ แบบฟอร์มที่ใช้	ผู้รับผิดชอบ	เงื่อนไขในการปฏิบัติงาน
๑๒. ฝึกอบรมการใช้งานระบบสารสนเทศ	๑.ประสานงานกับสำนัก/กองผู้ ขอรับบริการ เพื่อจัดฝึกอบรม ๒.จัดฝึกอบรมให้ผู้ใช้งาน และ ผู้ดูแลระบบ (ตามที่ตกลงกับ สำนัก/กองผู้ขอรับบริการ)	๑.เอกสาร/ระเบียบการจัด ฝึกอบรม ๒.แบบประเมินผลการฝึกอบรม	๑. กลุ่มระบบงานสารสนเทศ และภูมิสารสนเทศ ๒. สำนัก/กอง ผู้ขอรับบริการ	เป็นไปตามแนวทางการจัด ฝึกอบรมที่หน่วยงานกำหนด
๑๓. ติดตาม ประเมินผลการใช้ งาน	ติดตามประเมินผลการใช้งาน ระบบ ตามรอบระยะเวลาที่ กำหนด		๑. กลุ่มระบบงานสารสนเทศ และภูมิสารสนเทศ ๒. สำนัก/กอง ผู้ขอรับบริการ	ตัวชี้วัดผลสำเร็จของงาน
๑๔. สรุปผลการดำเนินงานและ ประมวลปัญหา อุปสรรค การใช้งาน	สรุปผลการดำเนินการของการ พัฒนาระบบต่อผู้บังคับบัญชา		กลุ่มระบบงานสารสนเทศและ ภูมิสารสนเทศ	

๗. มาตรฐานคุณภาพงาน

ระบบสารสนเทศที่พัฒนามีความครบถ้วน สมบูรณ์ตรงกับความต้องการของผู้ใช้งานหรือสำนัก/กอง ผู้ขอรับบริการ

๘. ระบบติดตามประเมินผล

มีการทดสอบและประเมินผลการใช้งานระบบสารสนเทศอย่างต่อเนื่องทุก ๖ เดือน เพื่อติดตามผล การใช้งานของผู้ใช้งานหรือสำนัก/กองผู้ขอรับบริการ

๙. เอกสารอ้างอิง

- ไม่มี -

๑๐. แบบฟอร์มที่ใช้

- ไม่มี -

ภาคผนวก

ภาคผนวก ๑

การพัฒนาระบบสารสนเทศ

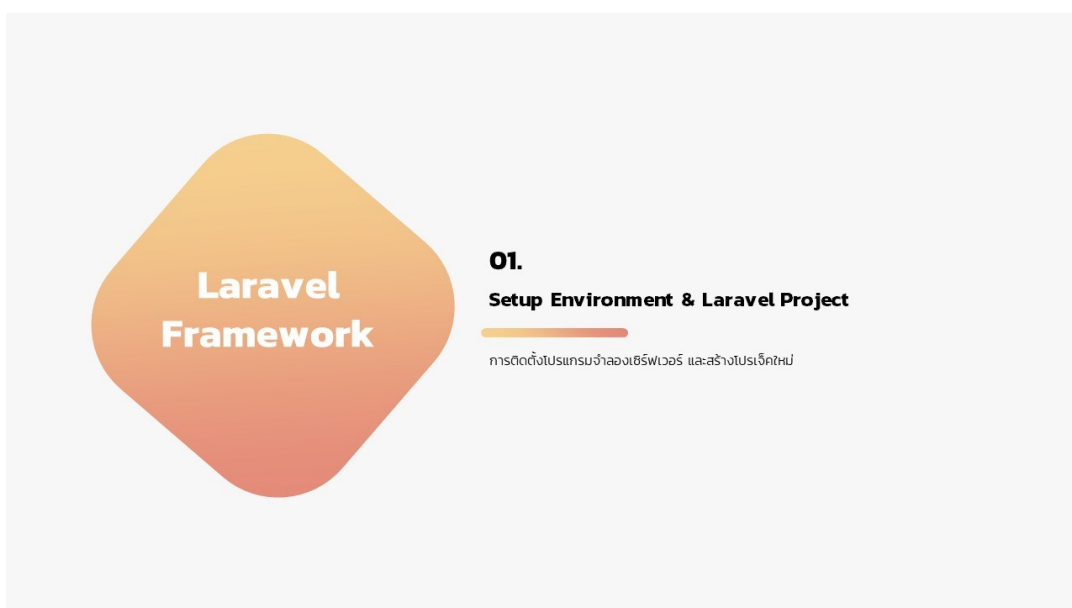
(Web Application)

ด้วย Laravel Framework



CONTENTS

- 01. Setup Environment & Laravel Project**
การติดตั้งโปรแกรมจำลองเซิร์ฟเวอร์ และสร้างโปรเจกต์ใหม่
- 02. Introduction to Laravel Framework**
ทำความเข้าใจกับ Laravel Framework และ MVC Model
- 03. Workshop (Webboard Web Application)**
การสร้างเว็บบอร์ดด้วย Laravel Framework.
- 04. Deploy Laravel Project**
การอัปโหลดโปรเจกต์ไปสู่เซิร์ฟเวอร์เพื่อใช้งานจริง





Xampp เป็นโปรแกรม Apache web server จำลอง web server เพื่อทดสอบ สคริปหรือเว็บไซต์ในเครื่องของเรา โดยที่ไม่ต้องเชื่อมต่ออินเทอร์เน็ตและไม่ต้องมีค่าใช้จ่ายใดๆ ง่ายต่อการติดตั้งและใช้งานโปรแกรม Xampp จะมาพร้อมกับ PHP ภาษาสำหรับพัฒนาเว็บแอปพลิเคชันที่เป็นที่นิยม MySQL ฐานข้อมูล, Apache จะทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ อีกทั้งยังมาพร้อมกับ OpenSSL , phpMyadmin (ระบบบริหารฐานข้อมูลที่พัฒนาโดย PHP เพื่อใช้เชื่อมต่อไปยังฐานข้อมูล) สนับสนุนฐานข้อมูล MySQL และ SQLite

<https://www.apachefriends.org/index.html>



Sublime Text 3 เป็นโปรแกรมเขียนโค้ดซึ่งสนับสนุนภาษาที่หลากหลาย เช่น C, C++, C#, CSS, D, Erlang, HTML, Groovy, Haskell, HTML, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, Matlab, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile และ XML

<https://www.sublimetext.com/3>



Composer เป็นเครื่องมือของ PHP ใช้จัดการ library ที่ต้องการใช้ในโปรเจกต์ ลักษณะการใช้งานคือ ให้เราระบุ library ที่โปรเจกต์ของเราต้องการไว้ในไฟล์ composer.json จากนั้น composer จะทำการติดตั้งหรืออัปเดต library ที่เราต้องการให้เลย ช่วยให้เรารจัดการกับ library ได้ง่ายขึ้น

<https://getcomposer.org/>



Laravel

Laravel คือ PHP Framework รูปแบบ MVC และเป็นที่ยอมรับใช้มากของนักพัฒนา ระบบ หรือเว็บแอปพลิเคชันในปัจจุบันเพราะมีความสามารถที่ช่วยในการทำงานให้ง่าย และเป็นระเบียบมากขึ้น

ข้อดี : Laravel เป็น framework ที่สามารถดึง class php สำเร็จรูปที่ดีที่มีคนเผยแพร่ใน อินเทอร์เน็ตไว้ให้โหลดฟรีมาใช้งานได้จึงทำให้นักพัฒนาระบบหรือเว็บแอปพลิเคชัน ไม่ต้องเขียนโค้ดเองทั้งหมดโดยทำงานร่วมกับ composer ซึ่งเป็นโปรแกรมที่ให้เรา ติดตั้งบนเครื่องที่เราใช้งานประโยชน์ของมันคือช่วยให้เราดาวน์โหลดปลั๊กอินมาติดตั้ง ในโปรเจกต์

ข้อเสีย : ถ้ามีการอัปเดตเวอร์ชันบางที่อาจทำให้มีบางไฟล์เปลี่ยนไปทำให้ไม่ตรงกับเวอร์ชันเดิม จึงอาจทำให้เกิดความวุ่นวายในการอัปเดตพอสมควร



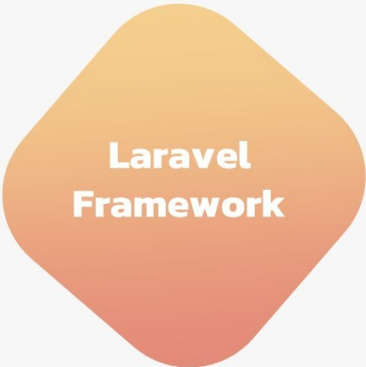
Laravel

Setup Laravel Project

```
composer create-project --prefer-dist laravel/laravel blog "5.4.*"
```

Run Laravel Project

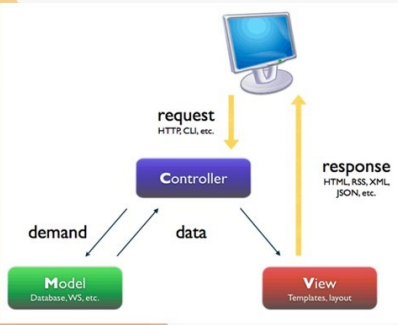
```
php artisan serve
```



Laravel Framework

02.
Introduction to Laravel framework

ทำความเข้าใจกับ Laravel Framework ใ้: MVC Model

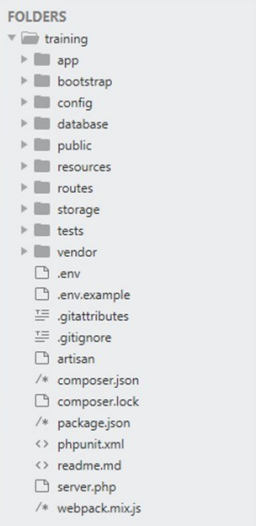


The diagram illustrates the MVC Model. At the top, a computer icon represents the user. A yellow arrow labeled 'request' (with 'HTTP, CLI, etc.' below it) points from the user to a blue box labeled 'Controller'. A yellow arrow labeled 'response' (with 'HTML, RSS, XML, JSON, etc.' below it) points from the Controller back to the user. Below the Controller, a green box labeled 'Model' (with 'Database, WS, etc.' below it) and a red box labeled 'View' (with 'Templates, layout' below it) are connected to the Controller. A double-headed arrow labeled 'demand' connects the Model to the Controller, and another double-headed arrow labeled 'data' connects the Controller to the View.

MVC Model

MVC เป็นสถาปัตยกรรมซอฟต์แวร์รูปแบบหนึ่ง ที่มีโครงสร้างแบ่งออกมาเป็น 3 ส่วนหลัก ตามด้วยย่อของชื่อ ได้แก่ Model View และ Controller

- 1. Controller**
 - เป็นส่วนที่ควบคุมการทำงานของโปรแกรม (Logic)
 - เป็นส่วนที่ติดต่อการทำงานระหว่างผู้ใช้และ โปรแกรม
 - มีการติดต่อกับ Database(ฐานข้อมูล) ด้วย Model และ แสดงผลข้อมูลผ่านทาง View
 - เป็นส่วนที่มีการประมวลผลหลัก ของ โปรแกรม
- 2. Model**
 - เป็นส่วนที่ติดต่อสื่อสารระหว่าง Object กับ Database (ฐานข้อมูล)
 - เชื่อมความสัมพันธ์ของข้อมูล (Relationship)
 - ตรวจสอบความถูกต้องของข้อมูล
- 3. View**
 - เป็นส่วนที่แสดงผลผ่าน Web browser ที่ให้ผู้ใช้งานมองเห็น
 - โดยจะใช้ HTML แทนด้วย script PHP
 - การทำงานสัมพันธ์อยู่กับ Controller



A tree view of the Laravel project structure. The root folder is 'training', which contains several sub-folders: 'app', 'bootstrap', 'config', 'database', 'public', 'resources', 'routes', 'storage', 'tests', 'vendor', and files: '.env', '.env.example', '.gitattributes', '.gitignore', 'artisan', 'composer.json', 'composer.lock', 'package.json', 'phpunit.xml', 'readme.md', 'server.php', and 'webpack.mix.js'.

Laravel Project Structure

app – เก็บส่วนหลักๆ ของโปรเจกต์อย่าง Controller และ Model ที่ใช้ในการประมวลผลและติดต่อกับฐานข้อมูล

config – เก็บส่วนการตั้งค่าต่างๆ ของโปรเจกต์

database – เก็บส่วนที่จัดการกับ database ต่างๆ อย่าง migration และ seed data

public – เก็บส่วนที่ให้ user เห็นได้ เช่น ไฟล์ css, js และอื่นๆ และ ยังเก็บไฟล์ index.php ที่เป็นไฟล์เริ่มต้นของโปรเจกต์

resources – เก็บส่วนของ View ทั้งหมด

routes – เก็บส่วนของไฟล์ที่ใช้กำหนดเส้นทางของ URL ที่ User เรียกเข้ามา

storage – เก็บส่วนของไฟล์ cache, session ต่างๆ

vendor – เก็บ Package ต่างๆ ที่ใช้ Composer ติดตั้งเข้ามา

.env – ไฟล์ที่ใช้ในการตั้งค่าส่วนต่างๆ ของโปรเจกต์

Routing

การกำหนดเส้นทางของ URL และตรวจสอบสิทธิ์การเข้าถึงข้อมูล และกำหนดว่า Request ที่เข้ามานั้นจะทำงานต่ออย่างไร

```
Route::get('test', function() {
    return "ทดสอบ";
});
```

```
Route::post('test', function() {
    return "ทดสอบ";
});
```

```
Route::get('view', function() {
    return view('welcome');
});
```

```
Route::get('view', function() {
    return view('welcome');
})->name('view');
```

```
Route::get('/exportExcel', 'FormController@exportExcel');
```

Routing Prefix

สามารถจัดกลุ่มของ Routing โดยอ้างอิงชื่อ Prefix โดยไม่ต้องสร้างชื่อเดิมซ้ำๆ

```
Route::prefix('product')->group(function() {  
  
    Route::get('get', function() {  
        return "Get Product";  
    })  
  
    Route::post('create', function() {  
        return "Create Product";  
    })  
  
});
```

```
Route::get('product/get', function() {  
    return "Get Product";  
})
```

```
Route::post('product/create', function() {  
    return "Create Product";  
})
```

Routing Parameter

ในการทำงาน หลายครั้งจะมีการส่งค่า parameter ต่างๆ มาทาง URL ด้วย เช่น www.moac.go.th/product/1234

```
Route::get('test/{id}', function($id) {  
    return $id;  
})
```

```
Route::get('test/{id?}', function($id=null) {  
    return $id;  
})
```

```
Route::get('test/{id}/{name}', function() {  
    return $id;  
})->where(['id' => '[0-9]+', 'name' => '[a-z]+']);
```

Controller

ส่วนของการจัดการ การทำงานต่างๆ ของระบบ (Logic) ซึ่งอยู่ระหว่างกลางของ Model และ View

การสร้าง Controller

```
php artisan make::controller TestController
```

controller ที่สร้างจะอยู่ที่ไฟล์ชื่อ `App/Http/Controllers`

```
public function test() {  
    return "ทดสอบ";  
}  
  
public function testView() {  
    return view("welcome");  
}  
  
public function testRedirect() {  
    return redirect()->to('/')->with('status', 'success');  
}
```

Routing Parameter To Controller

การส่งค่า Parameter ต่างๆ ที่มาจาก URL ไปที่ Controller

```
Route::get('/test/{id}/{name}', 'TestController@test');
```

```
public function test($id, $name) {  
  
    return $id . $name;  
}  
  
public function test(Request $request) {  
  
    return $request->id . $request->name;  
}
```

View

ส่วนที่ใช้ในการแสดงผลข้อมูลผ่านเว็บเบราว์เซอร์ ที่ User Request เข้ามาผ่านทาง URL

view จะอยู่ที่ resources/views

ทดลองสร้างไฟล์ php ในโฟลเดอร์ view โดยใช้นามสกุล .blade.php จากนั้นสร้างเส้นทางสำหรับเรียก view ที่สร้างขึ้นมา

```
<html>  
<body>  
    <h1> ทดสอบ View </h1>  
</body>  
</html>
```

```
<html>  
<body>  
  
    @if(...)  
        ...  
    @endif  
  
    @foreach(...)  
        ...  
    @endforeach  
  
</body>  
</html>
```

Passing data to view

การส่งข้อมูลจาก Controller มาแสดงผลในส่วนของ view

```
Route::get('/view/{id}/{name}', 'TestController@view');
```

```
public function view($request $request) {  
    $data = ['id' => $request->id, 'name' => $request->name];  
    return view('test')->with($data);  
}
```

```
<html>  
<body>  
    <h1> ID = {{ $id }} </h1>  
    <h1> Name = {{ $name }} </h1>  
</body>  
</html>
```

Blade Layout

สร้างไฟล์ `display.blade.php` เป็นไฟล์แสดงผลของ Layout

```
@extends('layout/master')

@section('content')

    <h1>แสดงส่วนของเนื้อหา</h1>

@endsection
```

`@extends` ใช้ในการเรียก Layout จาก `master.blade.php` มาใช้งาน

`@section` ใช้ในการแสดงผลส่วนของ `@yield` ที่สร้างไว้ใน `master.blade.php` โดยต้องทำการปิดด้วย `@endsection`

`@include` ใช้ในการดึงส่วนของ Layout นั้นๆ มาใช้งาน

Blade Layout

ในส่วนของ View จะมีการนำเอา Blade Template มาใช้งาน ซึ่งเป็น Template Engine ที่สามารถทำ Layout ต่างๆ แยกออกจากกัน แล้วนำมาประกอบกัน เพื่อสร้างเป็นหน้าเว็บเพจได้ โดยไม่ต้องเขียนโค้ดเหมือนเดิมซ้ำๆ

สร้างไฟล์ชื่อ `layout` และสร้างไฟล์ `master.blade.php` เป็นไฟล์ Master ของ Layout

```
<html>
  <body>
    <h1> Header </h1>
    <h1> Content </h1>
    @yield('content')
    <h1> Footer </h1>
  </body>
</html>
```

`@yield` ใช้ในการสร้างตัวแปร สำหรับที่จะนำมาแสดงผลเนื้อหาใน Layout อื่นๆ

Model

ส่วนของแบบจำลองแต่ละตารางในฐานข้อมูล ที่จะช่วยให้เราสามารถเชื่อมต่อและทำงานกับฐานข้อมูลได้ง่ายขึ้น

สร้างฐานข้อมูลใหม่โดยใช้ `phpMyAdmin`

```
Database name = training
Collation = utf8_general_ci
```

เชื่อมต่อกับฐานข้อมูล โดยตั้งค่าในไฟล์ `.env`

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=training
DB_USERNAME=root
DB_PASSWORD=
```


Migration

เรียกใช้ฟังก์ชัน up จากไฟล์ migration ด้วยคำสั่ง

```
php artisan migrate
```

เรียกใช้ฟังก์ชัน down จากไฟล์ migration ด้วยคำสั่ง

```
php artisan migrate:rollback
```

Migration

ส่วนของการจัดการตารางในฐานข้อมูล เช่น การเพิ่ม ลบ หรือแก้ไข ช่วยให้สามารถจัดการโครงสร้างในฐานข้อมูลได้อย่างเป็นระบบ

สร้างไฟล์ migration โดยใช้คำสั่ง

```
php artisan make:migration create_training_table
```

migrations ที่สร้างจะอยู่ที่โฟลเดอร์ database/migrations

```
public function up() {  
    Schema::create('training', function (Blueprint $table) {  
        $table->increments('id')->index();  
        $table->string('name')->default("OHM");  
        $table->string('surname', 100)->nullable();  
        $table->integer('phone');  
        $table->timestamp();  
    });  
}
```

```
public function down()  
{  
    Schema::dropIfExists('training');  
}
```

Model Relationships

การสร้างความสัมพันธ์ของโมเดลแต่ละตารางในฐานข้อมูล ช่วยให้การเรียกใช้ข้อมูลจากฐานข้อมูลทำได้ง่ายขึ้น

แก้ไขตาราง Training เพื่อเพิ่มความสัมพันธ์กับตาราง Users

```
php artisan make:migration update_training_table
```

```
public function up() {  
    Schema::table('training', function (Blueprint $table) {  
        $table->integer('user_id')->after('id')->unsigned();  
        $table->foreign('user_id')->references('id')->on('users');  
    });  
}
```

```
public function down()  
{  
    Schema::table('training', function (Blueprint $table) {  
        $table->dropForeign(['user_id']);  
        $table->dropColumn('user_id');  
    });  
}
```


Model Relationships

สร้างความสัมพันธ์ระหว่างตาราง Users และ Training

User Model

สร้างความสัมพันธ์แบบ **one-to-one**

```
public function training()
{
    return $this->hasOne(Training::class, 'user_id');
}
```

สร้างความสัมพันธ์แบบ **one-to-many**

```
public function training()
{
    return $this->hasMany(Training::class, 'user_id');
}
```

Model Relationships

สร้าง Model ด้วยคำสั่ง

```
php artisan make:model Training
```

Model ที่สร้างจะอยู่ที่โฟลเดอร์ **app**

กำหนดชื่อตารางในฐานข้อมูลให้กับโมเดล

```
protected $table = "training";
```

Model Relationships

สร้างความสัมพันธ์ระหว่างตาราง Users และ Training

Training Model

สร้างความสัมพันธ์กลับไป User Model แบบ **one-to-one** หรือ **one-to-many**

```
public function user()
{
    return $this->belongsTo(User::class);
}
```

สร้างความสัมพันธ์กลับไป User Model แบบ **many-to-many**

```
public function user()
{
    return $this->belongsToMany(User::class);
}
```

Eloquent

Insert Data

```
$user = new User;
$user->name = "OHM";
$user->email = "nattapat_pr@opsmoac.go.th";
$user->password = bcrypt(123456);
$user->save();
```

```
$training = new Training;
$training->user_id = $user->id;
$training->name = "Nattapat";
$training->surname = "Prachayatiwat";
$training->save();
```

Eloquent

สำหรับ Model ใน Laravel จะมีการใช้ Eloquent ORM ที่จะช่วยให้การ Query ข้อมูลจากฐานข้อมูลได้อย่างง่ายดาย และรวดเร็ว

สร้าง Routing ใหม่

```
Route::get('/database/get', 'TestController@get');
Route::get('/database/insert', 'TestController@insert');
Route::get('/database/update', 'TestController@update');
Route::get('/database/delete', 'TestController@delete');
```

การเรียกใช้งาน Model ใน Controller

```
use App\User;
use App\Training;
```

Eloquent

Get Data

```
$user = User::all();
```

```
$user = User::where('id', '=', 1)->get();
```

```
$user = User::find(1);
```

```
$user = User::with('training')->get();
```

```
$user = User::whereHas('training', function($q) {
    $q->where('id', 1);
})->get();
```

```
$user->name;
```

```
$user->training->name;
```

Eloquent

Update Data

```
$training = Training::where('id', 1)->update(['name' => "N", "surname" => "P"]);
```

```
$training = Training::find(1);  
$training->name = "N";  
$training->surname = "P";  
$training->save();
```

Delete Data

```
Training::find(1)->delete();
```

```
$training = Training::find(1);  
$training->delete();
```

Laravel Framework

03.

Workshop (Webboard Web Application)

การสร้างเว็บบอร์ดด้วย Laravel Framework

Workshop

Requirement

1. เว็บบอร์ดสามารถเขียนโพสต์ได้เฉพาะสมาชิกเท่านั้น
2. ผู้ที่ไม่ได้เป็นสมาชิกสามารถเข้าสู่โพสต์ได้ แต่ไม่สามารถเขียนโพสต์ใหม่ และไม่สามารถแสดงความคิดเห็นได้
3. ทุกโพสต์ต้องผ่านการตรวจสอบจากผู้ดูแลระบบก่อน
4. เมื่อมีการแก้ไขโพสต์ต้องผ่านการตรวจสอบจากผู้ดูแลระบบอีกครั้ง
5. การลบโพสต์ สามารถทำได้โดยผู้ดูแลระบบเท่านั้น
6. การเขียนโพสต์ และการเขียนแสดงความคิดเห็น ต้องสามารถปรับรูปแบบข้อความได้ แสดงรูปภาพได้

Workshop

Setup New Project

```
composer create-project --prefer-dist laravel/laravel blog "5.4.*"
```

Run Laravel Project

```
php artisan serve
```

Workshop

Database Design

สร้างตารางฐานข้อมูลโดยใช้ Migration จาก Diagram ด้านล่างนี้ จากนั้นสร้าง Model พร้อมทั้งกำหนดความสัมพันธ์

users	
id	int (PK)
name	varchar
email	varchar
password	varchar
remember_token	varchar
created_at	timestamp
update_at	timestamp

topic	
id	int (PK)
user_id	int (FK)
approve_status	char
subject	varchar
message	test
created_at	timestamp
updated_at	timestamp

comment	
id	int (PK)
topic_id	int (FK)
user_id	int (FK)
message	text
created_at	timestamp
updated_at	timestamp

Workshop

Authentication

สร้างระบบการตรวจสอบสิทธิ์การเข้าถึงข้อมูล ก่อนการเข้าใช้งานระบบ

สามารถสร้างระบบการตรวจสอบสิทธิ์ด้วยคำสั่ง

```
php artisan make:auth
```

Class Auth สามารถใช้ดึงข้อมูลของผู้ใช้งานที่อยู่ในระบบออกมาได้ทันที

```
use Auth;
Auth::check();
Auth::user()->id;
Auth::user()->name;
```

สร้าง Controller ใหม่

```
php artisan make:controller TopicController
```

Workshop

Authentication

เพิ่มการตรวจสอบสิทธิ์ของ controller ในส่วนของฟังก์ชัน construct

```
public function __construct() {  
    $this->middleware('auth');  
}
```

เพิ่มฟังก์ชัน index เพื่อใช้ในการแสดงผลหน้าแรกของระบบ

```
public function index() {  
    return view('home');  
}
```

เพิ่มการยกเว้นการตรวจสอบสิทธิ์ให้กับฟังก์ชัน index เพื่อให้ผู้ใช้งานทั่วไปเข้าถึงได้

```
public function __construct() {  
    $this->middleware('auth', ['except' => ['index']]);  
}
```

Workshop

Authentication

ปรับเส้นการแสดงผลหน้าแรกไปที่ TopicController@index

```
Route::get('/', 'TopicController@index');
```

Workshop

Config

ตั้งค่าระบบเบื้องต้น

ที่ไฟล์ .env

```
APP_NAME=Training-Workshop  
APP_ENV=local  
APP_DEBUG=true  
APP_URL=http://127.0.0.1:8000
```

ที่ไฟล์ config/app.php

```
'timezone' => 'Asia/Bangkok',
```

Workshop

สร้างปุ่มสำหรับการเพิ่มโพสต์ใหม่ ที่ไฟล์ `resources/view/home.blade.php`

```
href="{{ url('post/create') }}"
```

สร้างเส้นให้กับการเพิ่มโพสต์ใหม่

```
Route::get('/post/create', 'TopicController@create');
```

สร้างฟังก์ชัน `create` ใน `TopicController`

```
public function create()
{
    return view('post/create');
}
```

Workshop

สร้างไฟล์คอนโทรลเลอร์ `post` ที่ `resources/view` และสร้างไฟล์ `create.blade.php` ในโฟลเดอร์ `post`

สร้างฟอร์มสำหรับการโพสต์ใหม่ด้วย Blade Template จากนั้น Post ข้อมูลในฟอร์ม ไปที่ TopicController พร้อมทั้ง Insert ข้อมูลเข้าไปที่ Topic Model

Workshop

Form Validation

Controller สามารถตรวจสอบข้อมูลที่มาจากฟอร์มได้ว่าถูกต้องครบถ้วนหรือไม่

```
use Validator;

$this->validate($request, [
    'subject' => 'required|max:255|min:20',
    'message' => 'required|min:100',
]);
```

Workshop

Install CKEditor Package

Package ที่ช่วยในการสร้าง Text Editor Form

```
composer require unisharp/laravel-ckeditor
```

ที่ไฟล์ `config/app.php` เพิ่ม `package service` ใน array ของ `providers`

```
Unisharp\CKeditor\ServiceProvider::class,
```

Publish the resources เพื่อให้ง่ายต่อการจัดการกับ package

```
php artisan vendor:publish --tag=ckeditor --force
```

Workshop

Install CKEditor Package

ที่ไฟล์ `Create Post` ในไฟล์เตอร์ `resources/view/post` เพิ่ม ID ให้กับ `Textarea`

```
id="ckeditor"
```

เพิ่ม script ส่วนของ Footer

```
<script src="vendor/unisharp/laravel-ckeditor/ckeditor.js"></script>
```

```
<script> CKEDITOR.replace('ckeditor') </script>
```

Workshop

Install CKEditor Package

ที่ไฟล์ `public/vendor/unisharp/config.js` ปรับให้สามารถจัดตำแหน่งข้อความได้

```
config.extraPlugins = 'justify';
```

Workshop

Install File Manager Package

Package ที่ช่วยให้สามารถจัดการกับไฟล์จากเครื่องไคลเอนต์ได้ง่ายขึ้น สามารถใช้ร่วมกับ CKEditor ได้

```
composer require unisharp/laravel-filemanager
```

ที่ไฟล์ config/app.php เพิ่ม package service ใน array ของ providers

```
UniSharp\LaravelFilemanager\LaravelFilemanagerServiceProvider::class,
```

```
Intervention\Image\ImageServiceProvider::class,
```

เพิ่ม package service ใน array ของ aliases

```
'Image' => Intervention\Image\Facades\Image::class,
```

Workshop

Install File Manager Package

Publish the resources เพื่อให้ง่ายต่อการจัดการกับ package

```
php artisan vendor:publish --tag=lfm_config—force
```

```
php artisan vendor:publish --tag=lfm_public—force
```

```
php artisan vendor:publish --tag=lfm_view —force
```

Link โฟลเดอร์ storage เพื่อใช้ในการเก็บข้อมูลรูปภาพที่อัปโหลด

```
php artisan storage:link
```

Workshop

Install File Manager Package

สร้างเส้นทางสำหรับ Package File Manager ใน Routing

```
Route::group(['prefix' => 'laravel-filemanager', 'middleware' => ['web', 'auth']], function () {  
    \UniSharp\LaravelFilemanager\Lfm::routes();  
});
```

ที่ไฟล์ vendor/unisharp/laravel-filemanager/Lfm.php แก้ไข Path ของ Routing

```
\UniSharp\LaravelFilemanager\Controllers\
```


Workshop

Install File Manager Package

ที่ไฟล์ `Create Post` ในโฟลเดอร์ `resources/view/post` เพิ่ม Script ส่วนของ Footer

```
var options = {
    filebrowserImageBrowseUri: '/laravel-filemanager?type=Images',
    filebrowserImageUploadUri: '/laravel-filemanager/upload?type=Images&_token=' + '{{ csrf_token() }}',
    filebrowserBrowseUri: '/laravel-filemanager?type=Files',
    filebrowserUploadUri: '/laravel-filemanager/upload?type=Files&_token=' + '{{ csrf_token() }}'
};
```

ปรับส่วนของการเรียกฟังก์ชัน `CKEDITOR` โดยเพิ่มการ `options`

```
CKEDITOR.replace('ckeditor', options);
```

Workshop

Middleware

สร้างระบบการตรวจสอบสิทธิ์การเข้าถึงข้อมูล ก่อนการเข้าใช้งานระบบ แบบหลายระดับชั้น

```
php artisan make:middleware AdminAuth
```

middleware ที่สร้างจะอยู่ที่โฟลเดอร์ `app/Http/Middleware`

```
use Auth;

public function handle($request, Closure $next)
{
    if (Auth::user()->role == 1) {
        return $next($request);
    } else {
        return redirect('/');
    }
}
```

Workshop

Middleware

แก้ไขตาราง `Users` เพิ่มคอลัมน์ `role` สำหรับใช้ในการกำหนดสิทธิ์ของแต่ละผู้ใช้งาน

```
php artisan make:migration update_users_table
```

```
public function up()
{
    Schema::table('users', function (Blueprint $table) {
        $table->char('role')->default('0')->after('id');
    });
}
```

```
public function down()
{
    Schema::table('users', function (Blueprint $table) {
        $table->dropColumn('role');
    });
}
```

Workshop

Middleware

เพิ่มส่วนของ middleware ที่ได้สร้างใหม่ เข้าไปในการทำงานของระบบ

ที่ไฟล์ `app/Http/Kernel.php` เพิ่ม middleware ใน array ของ `$routeMiddleware`

```
'adminAuth' => \App\Http\Middleware\AdminAuth::class,
```

สร้างปุ่มสำหรับการเข้าไปที่หน้าอนุมัติโพสต์ของผู้ใช้งาน โดยปุ่มนี้ admin สามารถมองเห็นได้เท่านั้น

ที่ไฟล์ `resources/view/home.blade.php`

```
@if(Auth::user()->role == 1)
```

```
@endif
```

Workshop

Data Seeder

เพิ่มข้อมูลเข้าฐานข้อมูลโดยตรง ส่วนมากจะใช้เพื่อเพิ่มข้อมูลที่มีค่าคงที่ เช่น ข้อมูลจังหวัด และใช้กับการเพิ่มข้อมูล Dummy เพื่อทดสอบ

```
php artisan make:seeder AdminSeeder
```

Seeder ที่สร้าง จะอยู่ในไฟล์ชื่อ `database/seeds`

Insert Data ในฟังก์ชัน `run`

```
DB::table('users')->insert([
    'role' => 1,
    'name' => 'admin',
    'email' => 'admin@moac.go.th',
    'password' => bcrypt(123456),
]);
```

```
php artisan db:seed --class=AdminSeeder
```

Workshop

Middleware

สร้างเส้นทางให้กับการอนุมัติโพสต์ของผู้ใช้งาน พร้อมทั้งกำหนดสิทธิ์การเข้าถึงเฉพาะ admin เท่านั้น

```
Route::group(['middleware' => 'adminAuth'], function()
{
    Route::get('/admin/approve/list', 'TopicController@approveList');
});
```

```
Route::group(['prefix' => 'admin', 'middleware' => 'adminAuth'], function()
{
    Route::get('/approve/list', 'TopicController@approveList');
});
```

Workshop

Middleware

สร้างฟังก์ชัน approveList ใน TopicController พร้อมทั้งดึงข้อมูลโพสต์ที่ยังไม่ได้อนุมัติจาก Model ไปแสดงที่ View

```
use App\Topic;

public function approveList() {
    $topic = Topic::where('approve_status', 'F')->get();
    return view('post/approve/list')->with('topic', $topic);
}
```

สร้างไฟล์เตอร์ approve ที่ resources/view/post และสร้างไฟล์ list.blade.php ในไฟล์เตอร์ approve

```
@foreach($topic as $value) {
    {{ $topic->subject }}
    {{ $topic->user->name }}
}
```

Workshop

สร้างหน้า View สำหรับดูรายละเอียดของโพสต์ และสร้างปุ่มสำหรับการ Approve โพสต์จากนั้นทำการ POST ข้อมูลไปที่ Controller พร้อมทั้ง Update ข้อมูลใน Model

Workshop

แสดงข้อมูลของโพสต์ที่ได้รับการอนุมัติแล้วที่หน้า Home โดยดึงข้อมูลจาก Model และผู้ใช้งานทั่วไปสามารถมองเห็นได้

Workshop

Pagination

การแบ่งหน้าข้อมูลในกรณีที่มีข้อมูลมีจำนวนมากๆ

```
public function approveList() {  
    $topic = Topic::where('approve_status', 'T')->paginate(10);  
    return view('/', >with($topic);  
}
```

```
@foreach($topic as $value) {  
    $topic->subject;  
    $topic->message;  
}
```

```
{!! $topic->render() !!}
```

Workshop

สร้างหน้า View สำหรับดูรายละเอียดของโพสต์ โดยผู้ใช้งานทั่วไปสามารถมองเห็นได้และสร้างฟอร์มสำหรับการแสดงความคิดเห็นสำหรับสมาชิกเท่านั้น จากนั้นให้ทำการ POST ข้อมูลความคิดเห็นไปที่ Controller เพื่อ Insert ข้อมูลเข้าสู่ Model พร้อมทั้งให้แสดงข้อมูลความคิดเห็นในหน้ารายละเอียดโพสต์นี้ด้วย

Workshop

สร้างหน้าโพสต์ของฉันและทำการดึงข้อมูลโพสต์ที่ผู้ใช้งานนั้นได้ทำการสร้างไว้มาแสดงผล พร้อมทั้งสร้างปุ่มสำหรับแก้ไขโพสต์ จากนั้นสร้างเส้นทางสำหรับการแก้ไขและโพสต์ไปที่ Controller เพื่อ Update ข้อมูลใน Model โดยต้องให้ผู้ดูแลระบบทำการตรวจสอบอีกครั้ง

Workshop

สร้างปุ่มสำหรับการ ลบโพสต์ ที่สามารถมองเห็นได้เฉพาะ admin เท่านั้น ในหน้า Home จากนั้นสร้างเส้นทางสำหรับการลบโพสต์ ไปที่ Controller เพื่อ Delete ข้อมูลใน Model

Laravel Framework

04.

Deploy Laravel Project

การอัปเดตประเจ็ดขึ้นสู่เซิร์ฟเวอร์เพื่อใช้งานจริง

Deploy Laravel Project

Config

ตั้งค่าระบบก่อนการอัปเดตขึ้นเซิร์ฟเวอร์

ที่ไฟล์ .env

```
APP_NAME=Training-Workshop
APP_ENV=production
APP_DEBUG=false
APP_URL=Domain...
```

Deploy Laravel Project

สร้าง `PublicPathServiceProvider` เพื่อให้ระบบสามารถค้นหาไฟล์เดอร์ `public` ได้ หลังจากเปลี่ยน environment

```
php artisan make:provider PublicPathServiceProvider
```

provider ที่สร้างจะอยู่ที่ไฟล์เดอร์ `app/http/Providers` แก้ไขในฟังก์ชัน `register()`

```
if ($this->app->environment() === 'local') {
    $this->app['path.public'] = public_path();
} else if ($this->app->environment() === 'production') {
    $this->app['path.public'] = base_path().'../.env('PUBLIC_PATH');
}
```

Deploy Laravel Project

เพิ่มส่วนของ provider ที่ได้สร้างใหม่ เข้าไปในการทำงานของระบบ

ที่ไฟล์ `app/config.php` เพิ่ม provider ใน array ของ `providers`

```
App\Providers\PublicPathServiceProvider::class,
```

Deploy Laravel Project

แก้ไขไฟล์ `index.php` เพื่อเปลี่ยน path การเข้าถึง root folder ของระบบ

ที่ไฟล์ `public/index.php`

```
require __DIR__.'/../training/bootstrap/autoload.php';
```

```
function public_path($path = "")
{
    return realpath(__DIR__).( $path ? DIRECTORY_SEPARATOR.$path : $path);
}
```


```
$app = require_once __DIR__.'/../training/bootstrap/app.php';
```

Workshop

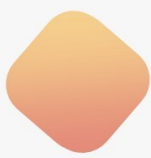
Upload Files

อัปโหลดไฟล์เดอร์ public ไปไว้ที่ไฟล์เดอร์ htdocs

อัปโหลดไฟล์เดอร์อื่นๆ ที่เหลือทั้งหมด ไปไว้ที่ไฟล์เดอร์ xampp (ระดับเดียวกับ htdocs)



Thank You



ภาคผนวก ๒
รายชื่อผู้จัดทำ

รายชื่อผู้จัดทำ

- | | |
|-------------------------------|--|
| ๑. นายสัญญา รัศมีจิรวีไล | ผู้อำนวยการศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร |
| ๒. นางสาวยุพาพร พัชราพินิจชัย | นักวิชาการคอมพิวเตอร์ชำนาญการพิเศษ |
| ๓. นางบุรินทร์พรรณ โพธิ์ทอง | นักวิชาการคอมพิวเตอร์ชำนาญการ |
| ๔. นายณัฐภัทร ปรัชญาธิวัฒน์ | นักวิชาการคอมพิวเตอร์ปฏิบัติการ |

กลุ่มระบบสารสนเทศและภูมิสารสนเทศ ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร
สำนักงานปลัดกระทรวงเกษตรและสหกรณ์