



อีดูเควส

EduQuest

นาย พชรพล เนตรสุวรรณ 664230022

66/45

โครงงานนี้เป็นส่วนหนึ่งของการศึกษารายวิชา 7204903

โครงงานด้านเทคโนโลยีสารสนเทศ 2

สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม

ภาคเรียนที่ 1 ปีการศึกษา 2568

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบัน การติดตามความก้าวหน้าทางการเรียนรู้และการพัฒนาทักษะส่วนบุคคลมักอาศัยวิธีการแบบดั้งเดิม เช่น การจดบันทึก การใช้สเปรดชีต หรือรายการสิ่งที่ต้องทำ ซึ่งเครื่องมือเหล่านี้มักมีลักษณะหยุดนิ่ง ขาดแรงจูงใจ และไม่สามารถให้ภาพรวมของความสามารถที่เปลี่ยนแปลงไปได้อย่างเป็นรูปธรรม โดยเฉพาะอย่างยิ่งในบริบทของการเรียนการสอนหรือการฝึกอบรม กระบวนการส่งมอบงานเพื่อรับการประเมินยังคงกระจุกตัวอยู่ตามช่องทางการสื่อสารต่างๆ ทำให้การรวบรวมหลักฐาน การให้คะแนน และการติดตามผลย้อนหลังทำได้ไม่สะดวก เกิดความล่าช้า และขาดระบบการจัดเก็บที่เป็นหนึ่งเดียว

จากปัญหาดังกล่าว โครงการนี้จึงมุ่งพัฒนาระบบเว็บแอปพลิเคชัน Edu Quest สำหรับติดตามและแสดงผลความก้าวหน้าทางการเรียนรู้ในรูปแบบเกม (Gamification) โดยผู้ใช้งานสามารถส่งเควส (Quest) หรืองานที่ทำสำเร็จพร้อมหลักฐานเพื่อรับการตรวจสอบจากผู้ดูแลระบบ เมื่อเควสได้รับการอนุมัติ ผู้ใช้จะได้รับคะแนนเพื่อไปเพิ่ม สเตตัส (Status) หรือค่าสถานะความสามารถในด้านต่างๆ ซึ่งจะถูกแสดงผลเป็นกราฟเรดาร์ที่เข้าใจง่าย

#### 1.2 แนวคิดในการแก้ไขปัญห

แนวคิดหลักของระบบคือการประยุกต์ใช้กลไกแบบเกม (Gamification) เพื่อให้การติดตามและประเมินความก้าวหน้าเป็นรูปธรรม มีแรงจูงใจ และสามารถตรวจสอบย้อนหลังได้ โดยได้พัฒนาเว็บแอปพลิเคชันขึ้นเพื่อทำหน้าที่เป็นศูนย์กลางในการส่งมอบและประเมินผลงาน ผู้ใช้จะส่ง "เควส" หรือภารกิจที่ตนเองทำสำเร็จพร้อมหลักฐานเข้ามาในระบบ จากนั้นข้อมูลจะถูกตรวจสอบโดยผู้ดูแล ซึ่งเมื่อได้รับการอนุมัติ ระบบจะ คำนวณและอัปเดตคะแนน ให้กับ สเตตัส ของผู้ใช้โดยอัตโนมัติ ข้อมูลเควสและสถานะทั้งหมดจะถูกจัดเก็บในรูปแบบไฟล์เจสัน (JSON) บนเซิร์ฟเวอร์เพื่อให้เรียกใช้และประมวลผลได้อย่างรวดเร็ว

ผลลัพธ์จะถูกนำเสนอผ่าน กราฟเรดาร์ ที่แสดงภาพรวมความสามารถของผู้ใช้โดยแยกตามหมวดหมู่อย่างชัดเจน ผู้ใช้สามารถ กดดูสถานะของตนเอง เพื่อประเมินจุดแข็งและจุดที่ต้องพัฒนาได้ตลอดเวลา ช่วยให้ทั้งผู้เรียนและผู้ดูแลใช้ ข้อมูลเชิงประจักษ์ที่ตรวจสอบได้ จากเควสที่ผ่านการอนุมัติ

แล้ว มาใช้ประกอบการวางแผนการเรียนรู้ ลดภาระในการรวบรวมและติดตามความคืบหน้าด้วยตนเอง และสร้างแพลตฟอร์มกลางที่ทำให้กระบวนการประเมินผลเป็นไปอย่างต่อเนื่องและมีประสิทธิภาพ

### 1.3 วัตถุประสงค์ของระบบ

โครงการนี้มีวัตถุประสงค์หลักในการพัฒนาระบบเว็บแอปพลิเคชันสำหรับติดตามความก้าวหน้าทางการเรียนรู้ด้วยกลไกแบบเกม (Gamification) โดยเป็นระบบศูนย์กลางที่ผู้ใช้สามารถส่งภารกิจ (เควส) และให้ผู้ดูแลตรวจสอบอนุมัติได้ในที่เดียว เพื่อสร้างแรงจูงใจในการเรียนรู้ และทำให้การประเมินผลเป็นไปอย่างมีระบบและชัดเจน โดยมีวัตถุประสงค์ย่อยดังนี้

1.3.1 เพื่อให้ผู้ดูแลระบบสามารถตรวจสอบและอนุมัติภารกิจ (เควส) ที่ผู้ใช้งานเข้ามา พร้อมทั้งให้คะแนนและข้อเสนอแนะได้อย่างเป็นระบบในที่เดียว

1.3.2 เพื่อให้ผู้ใช้สามารถติดตามความก้าวหน้าและประวัติการทำภารกิจของตนเองได้ผ่านการแสดงผลข้อมูลที่เข้าใจง่าย

1.3.3 เพื่อพัฒนาระบบการแสดงผลสถานะ (สเตตัส) ของผู้ใช้ในรูปแบบกราฟเรดาร์ที่ชัดเจน ทำให้สามารถวิเคราะห์จุดแข็งและจุดที่ควรพัฒนาของตนเองได้

### 1.4 ขอบเขตการศึกษา

การศึกษาในโครงการนี้จะเน้นไปที่การพัฒนาระบบติดตามความก้าวหน้าทางการเรียนรู้ด้วยกลไกแบบเกม (Quest & Status) โดยมีขอบเขตการศึกษาและการพัฒนาในหลายด้านที่เกี่ยวข้อง ดังนี้

#### 1.4.1 ขอบเขตของระบบ

##### 1.4.1.1 ผู้ดูแลระบบ

- ก) ตรวจสอบภารกิจ (เควส) ที่ผู้ใช้งานเข้ามาพร้อมหลักฐาน
- ข) อนุมัติหรือปฏิเสธเควส พร้อมให้คะแนนและบันทึกข้อเสนอแนะได้
- ค) ดูภาพรวมและสถิติของเควสทั้งหมดในระบบ
- ง) เรียกดูข้อมูลและกราฟแสดงสถานะ (สเตตัส) ของผู้ใช้แต่ละคนได้

##### 1.4.1.2 ผู้ใช้

- ก) ส่งภารกิจ (เควส) ที่ทำสำเร็จ พร้อมแนบไฟล์หลักฐานและคำอธิบายได้
- ข) ติดตามประวัติและสถานะ (รอตรวจ, อนุมัติ, ปฏิเสธ) ของเควสที่เคยส่งไป

ค) ดุลสถานะ (สเตตัส) ของตนเองที่แสดงผลผ่านกราฟเรดาร์ 2 รูปแบบ คือ หมวก  
วิชาทั่วไป และหมวกวิชาเฉพาะด้าน

ง) เห็นการเปลี่ยนแปลงของคะแนนในกราฟเมื่อมีเควสที่ได้รับการอนุมัติ

#### 1.4.1.3 ผู้ชม

ก) ติดตามสถิติผู้เล่นจากรายการแข่งขันพับจี

ข) เปรียบเทียบสถิติผู้เล่นในทัวร์นาเมนต์

ค) ดูกราฟการเปรียบเทียบผู้เล่น

#### 1.4.2 ฮาร์ดแวร์ที่ใช้ในการพัฒนา

1.4.2.1 โน้ตบุ๊ก เอเซอร์ (Acer) รุ่น ไนโตร (Nitro) AN515-44 ซีพียู เอเอ็มดี ไรเซ็น™ 5 ซีรีส์ (AMD Ryzen™ 5 4000H) เป็นเครื่องหลักสำหรับการพัฒนาและทดสอบระบบ

#### 1.4.3 ซอฟต์แวร์ที่ใช้ในการพัฒนา

1.4.3.1 ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 11 (Microsoft Windows 11)

1.4.3.2 พีเอชพี (PHP) เวอร์ชัน 8.0 ขึ้นไป สำหรับการพัฒนาระบบฝั่งเซิร์ฟเวอร์, เอชทีเอ็มแอล (HTML), ซีเอสเอส (CSS), และจาวาสคริปต์ (JavaScript) สำหรับการพัฒนาคิวต์ต่อผู้ใช้

1.4.3.3 บูทสเตรป (Bootstrap) เวอร์ชัน 5 สำหรับการออกแบบโครงสร้างเว็บ และชาร์ตเจเอส (Chart.js) สำหรับการแสดงผลกราฟ

1.4.3.4 วิวีสตูดิโอโค้ด (Visual Studio Code)

1.4.3.5 กิต (Git) สำหรับการควบคุมเวอร์ชันของโค้ด

#### 1.4.4 บริการแบบคลาวด์ที่ใช้ในการพัฒนา

1.4.4.1 เว็บโฮสติ้งทั่วไปที่รองรับภาษา PHP สำหรับการนำระบบขึ้นใช้งานจริง

1.4.4.3 กิตฮับ (GitHub) สำหรับการจัดเก็บและควบคุมเวอร์ชันของโค้ด

## 1.6 ประโยชน์ที่ได้คาดว่าจะได้รับ

1.6.1 ผู้ดูแลระบบสามารถติดตามและประเมินพัฒนาการของผู้ใช้แต่ละคนได้อย่างสะดวกและรวดเร็วผ่านข้อมูลสรุปและกราฟ ทำให้การให้ข้อเสนอแนะเป็นไปอย่างมีประสิทธิภาพ

1.6.2 ผู้ใช้สามารถเห็นภาพรวมความสามารถของตนเองผ่านกราฟสถานะ (สเตตัส) และมีประวัติภารกิจ (เควส) ที่ทำสำเร็จอย่างเป็นระบบสำหรับใช้อ้างอิงและวางแผนการพัฒนาตนเอง

1.6.3 ได้ระบบที่เป็นศูนย์กลางในการส่งเสริมและวัดผลการเรียนรู้ ซึ่งช่วยสร้างแรงจูงใจให้แก่ผู้ใช้ และทำให้กระบวนการประเมินผลโปร่งใสและตรวจสอบได้

## บทที่ 2

### หลักการและทฤษฎีที่เกี่ยวข้อง

ในการพัฒนาระบบติดตามความก้าวหน้าทางการเรียนรู้ Edu Quest ผู้จัดทำได้ศึกษาข้อมูล แนวคิด และเทคโนโลยีที่เกี่ยวข้องเพื่อใช้เป็นพื้นฐานในการออกแบบและพัฒนาระบบ โดยสามารถสรุปสาระสำคัญได้ดังนี้

#### 2.1 ระบบงานเดิม

##### 2.1 ระบบงานเดิม

โดยทั่วไป การติดตามความก้าวหน้าทางการเรียนรู้หรือการพัฒนาทักษะในปัจจุบันยังคงอาศัยเครื่องมือและกระบวนการที่หลากหลายและไม่เชื่อมโยงกันเป็นระบบ ผู้เรียนมักใช้เครื่องมือส่วนตัว เช่น สมุดบันทึก, โปรแกรมสเปรดชีต (Spreadsheet), หรือแอปพลิเคชันจดบันทึกทั่วไปในการวางแผนและบันทึกสิ่งที่ทำไปแล้ว เมื่อทำภารกิจหรือการบ้านเสร็จสิ้น จะทำการส่งงานผ่านช่องทางต่างๆ ที่ไม่ตายตัว เช่น การส่งอีเมล, แอปพลิเคชันสนทนา (LINE, Discord), หรือผ่านระบบจัดการการเรียนรู้ (LMS) ที่สถานศึกษาจัดหาให้

ในขณะเดียวกัน ผู้สอนหรือผู้ดูแลต้องคอยรวบรวมและตรวจสอบงานจากหลายช่องทาง ทำให้การประเมินผลและการให้ข้อเสนอแนะเกิดความล่าช้า ข้อมูลความคืบหน้าของผู้เรียนแต่ละคนจะถูกจัดเก็บแยกจากกันในรูปแบบที่แตกต่างกันไป ทำให้การวิเคราะห์ภาพรวมเพื่อประเมินจุดแข็งหรือจุดที่ต้องพัฒนาทำได้ยาก และต้องอาศัยการรวบรวมข้อมูลด้วยตนเองซึ่งใช้เวลานานและอาจเกิดข้อผิดพลาดได้ง่าย

ระบบงานเดิมนี้อาจมีข้อจำกัดในด้านความกระจัดกระจายของข้อมูล, ขาดความเป็นศูนย์กลางในการส่งและประเมินผล, และไม่มีเครื่องมือแสดงผลความก้าวหน้าในเชิงภาพที่ชัดเจน ซึ่งส่งผลโดยตรงต่อแรงจูงใจของผู้เรียนและประสิทธิภาพในการติดตามผลของผู้ดูแล

#### 2.2 ระบบงานอื่นที่เกี่ยวข้อง

ระบบเช่น Google Classroom หรือ Moodle ถูกออกแบบมาเพื่อสนับสนุนการเรียนการสอนในระบบการศึกษาเป็นหลัก มีจุดเด่นด้านการสร้างรายวิชา, การส่งงาน, และการรวบรวมไฟล์

เพื่อให้คะแนน อย่างไรก็ตาม ระบบเหล่านี้มักเน้นการให้เกรดเป็นรายชิ้นงานมากกว่าการแสดงผลภาพรวมของ "ชุดทักษะ" ที่ผู้เรียนได้พัฒนาขึ้น อีกทั้งกลไกการใช้งานยังเป็นทางการสูง อาจขาดแรงจูงใจในระยะยาว และไม่ยืดหยุ่นพอสำหรับการติดตามการพัฒนาทักษะนอกหลักสูตร

เครื่องมืออย่าง Trello, Asana, หรือ Todoist มีความสามารถสูงในการจัดระเบียบและติดตามสถานะของงาน (เช่น ยังไม่เริ่ม, กำลังทำ, เสร็จแล้ว) ซึ่งช่วยให้การบริหารจัดการโครงการมีประสิทธิภาพ อย่างไรก็ตาม แอปพลิเคชันเหล่านี้ถูกสร้างมาเพื่อติดตาม "การเสร็จสิ้นของงาน" ไม่ใช่ "การพัฒนาความสามารถ" จากงานนั้นๆ โดยขาดกลไกการประเมินผลเชิงคุณภาพ, การให้คะแนนเพื่อสะท้อนการเติบโต, และการแสดงผลทักษะในภาพรวม

แอปพลิเคชันอย่าง Habitica หรือ Duolingo ประสบความสำเร็จอย่างสูงในการใช้กลไกแบบเกมเพื่อสร้างแรงจูงใจในการสร้างนิสัยหรือเรียนรู้ทักษะเฉพาะทาง เช่น การเรียนภาษา แต่ระบบเหล่านี้มักถูกออกแบบมาให้มีวัตถุประสงค์เดียว ไม่ยืดหยุ่นพอที่จะรองรับภารกิจที่หลากหลายและซับซ้อนซึ่งต้องอาศัยการส่งหลักฐานและการตรวจสอบโดยผู้ดูแลเหมือนในระบบการศึกษาหรือการฝึกอบรม

เมื่อพิจารณาในภาพรวม แม้จะมีเครื่องมือที่มีประสิทธิภาพในแต่ละด้าน แต่ยังขาดระบบที่สามารถ บูรณาการ กระบวนการส่งมอบภารกิจที่ยืดหยุ่น, การตรวจสอบและให้คะแนนโดยผู้ดูแล, และการแสดงผลความก้าวหน้าของชุดทักษะต่างๆ ในภาพรวมด้วยกลไกที่สร้างแรงจูงใจ ผู้ใช้จำนวนมากจึงยังต้องใช้หลายแอปพลิเคชันร่วมกัน ทำให้ข้อมูลกระจัดกระจายและขาดความต่อเนื่อง โครงการงาน "Quest & Status" จึงถูกพัฒนาขึ้นเพื่อเข้ามาเติมเต็มช่องว่างนี้โดยเฉพาะ โดยมุ่งเน้นที่การสร้างแพลตฟอร์มเดียวที่ตอบโจทย์ทั้งการจัดการ การประเมินผล และการสร้างแรงจูงใจได้อย่างครบวงจร

## 2.3 องค์ความรู้ที่เกี่ยวข้อง

ในการพัฒนาโครงการระบบติดตามความก้าวหน้า "Edu Quest" นี้ ผู้จัดทำได้เลือกใช้ชุดเทคโนโลยีที่เหมาะสมทั้งฝั่งระบบปฏิบัติการ, ภาษาที่ใช้ในการพัฒนา, เฟรมเวิร์ก, ไลบรารี, และเครื่องมือต่างๆ ที่เกี่ยวข้อง ดังนี้

### 2.3.1 ไมโครซอฟท์ วินโดวส์ 11 (Microsoft Windows 11)

โครงการนี้พัฒนาบนระบบปฏิบัติการ ไมโครซอฟท์ วินโดวส์ 11 เนื่องจากมีความเสถียรและรองรับซอฟต์แวร์เครื่องมือพัฒนาที่หลากหลาย อีกทั้งยังมีการจัดการไฟล์และเครือข่ายที่สะดวก รวมถึงคุณสมบัติด้านความปลอดภัยและการอัปเดตสม่ำเสมอ เหมาะสำหรับการพัฒนาทดสอบ และจำลองสภาพแวดล้อมใช้งานของเว็บแอปพลิเคชันภายในเครื่องเดียวกันอย่างมีประสิทธิภาพ



ภาพที่ 2.1 ไมโครซอฟท์ วินโดวส์ 11

ที่มา <https://news.microsoft.com/th-th/2021/06/25/windows11-th/>

### 2.3.2 พีเอชพี (PHP)

ภาษาหลักที่ใช้ในการพัฒนาแกนของระบบฝั่งเซิร์ฟเวอร์ (Backend) ทั้งหมด มีจุดเด่นคือถูกออกแบบมาเพื่อการพัฒนาเว็บโดยเฉพาะและได้รับความนิยมสูง ในโครงการนี้ใช้ PHP สำหรับจัดการการเข้าสู่ระบบของผู้ใช้ (Session), ประมวลผลข้อมูลจากฟอร์มทั้งหมด เช่น การสร้างคิวส การอัปโหลดไฟล์ และการอนุมัติคิวสโดยผู้ดูแล รวมถึงทำหน้าที่สำคัญในการอ่านและเขียนข้อมูลลงไฟล์เจสัน (JSON) ซึ่งทำหน้าที่เป็นฐานข้อมูลของระบบ





ภาพที่ 2.2 พีเอชพี (PHP)

ที่มา <https://commons.wikimedia.org/wiki/File:PHP-logo.svg>

### 2.3.3 วิซวล สตูดิโอ โค้ด (Visual Studio Code)

ใช้เป็นเครื่องมือหลักในการเขียนโค้ด (Code Editor) เนื่องจากมีขนาดเล็ก ทำงานรวดเร็ว และรองรับการพัฒนาภาษา PHP, HTML, CSS และ JavaScript ได้เป็นอย่างดีผ่านส่วนขยาย (Extension) ที่หลากหลาย ภายในโปรแกรมมีเครื่องมือที่ช่วยอำนวยความสะดวกครบครัน เช่น เทอร์มินัลในตัว, ระบบดีบั๊ก, และการเชื่อมต่อกับ Git และ GitHub ทำให้การเขียนโค้ดและทดสอบโปรแกรมเป็นไปอย่างรวดเร็วและมีประสิทธิภาพ

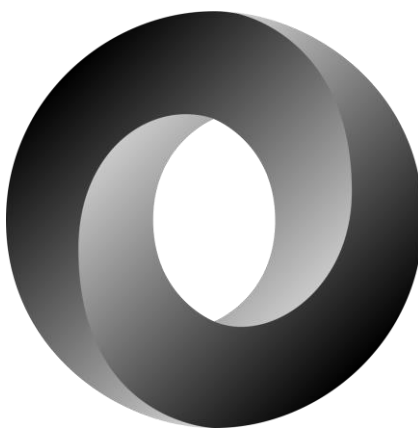


ภาพที่ 2.3 วิซวล สตูดิโอ โค้ด (Visual Studio Code)

ที่มา <https://www.stickpng.com/img/icons-logos-emojis/tech-companies/visual-studio-code>

#### 2.3.4 เจสัน (JSON)

รูปแบบข้อมูลที่ใช้เป็นฐานข้อมูลหลักของทั้งระบบ มีจุดเด่นที่โครงสร้างไม่ซับซ้อน มนุษย์สามารถอ่านทำความเข้าใจได้ง่าย และโปรแกรมสามารถประมวลผลได้รวดเร็ว ในโครงงานนี้ได้ใช้ไฟล์ JSON ในการจัดเก็บข้อมูล 3 ส่วนหลักคือ quests.json สำหรับเก็บข้อมูลเควส, stats.json สำหรับเก็บสถานะผู้ใช้, และ catalog.json สำหรับจัดกลุ่มรายวิชา การใช้ JSON ทำให้ระบบไม่ต้องพึ่งพาฐานข้อมูลที่ซับซ้อน ง่ายต่อการติดตั้งและใช้งาน

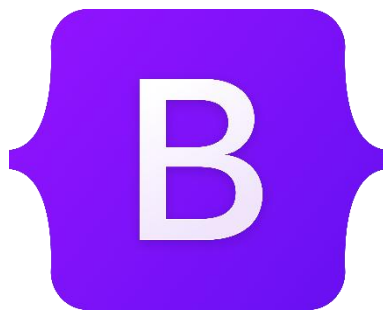


ภาพที่ 2.4 เจสัน (JSON)

ที่มา [https://commons.wikimedia.org/wiki/File:JSON\\_vector\\_logo.svg](https://commons.wikimedia.org/wiki/File:JSON_vector_logo.svg)

#### 2.3.5 บูทสเตรป (Bootstrap)

เฟรมเวิร์กสำหรับส่วนหน้าบ้าน (Frontend Framework) ที่ได้รับความนิยมอย่างสูง ใช้ในการออกแบบโครงสร้างและหน้าตาของเว็บแอปพลิเคชันทั้งหมด มีระบบกริด (Grid System) ที่ช่วยให้หน้าเว็บรองรับการแสดงผลบนอุปกรณ์ทุกขนาด (Responsive) และมีส่วนประกอบสำเร็จรูป เช่น การ์ด, ปุ่ม, และหน้าต่าง Modal ทำให้การพัฒนาส่วนติดต่อผู้ใช้มีความรวดเร็วและสวยงามเป็นมาตรฐานเดียวกัน



ภาพที่ 2.5 บูทสเตรป (Bootstrap)

ที่มา [https://commons.wikimedia.org/wiki/File:Bootstrap\\_logo.svg](https://commons.wikimedia.org/wiki/File:Bootstrap_logo.svg)

### 2.3.6 ชาร์ตเจเอส (Chart.js)

ไลบรารีจาวาสคริปต์ (JavaScript Library) ที่ใช้สำหรับสร้างกราฟและแผนภูมิ ในโครงการนี้ Chart.js เป็นเครื่องมือสำคัญในการแสดงผลข้อมูล "สเตตัส" ของผู้ใช้ออกมาในรูปแบบของกราฟเรดาร์ (Radar Chart) ซึ่งเป็นหัวใจหลักในการนำเสนอข้อมูลความสามารถของผู้ใช้ให้เห็นเป็นภาพที่ชัดเจนและเข้าใจง่าย



ภาพที่ 2.6 ชาร์ตเจเอส (Chart.js)

ที่มา [https://commons.wikimedia.org/wiki/File:Chart.js\\_logo.svg](https://commons.wikimedia.org/wiki/File:Chart.js_logo.svg)

### 2.3.7 กิต (Git)

ระบบควบคุมเวอร์ชัน (Version Control System) ที่ช่วยบันทึกการเปลี่ยนแปลงของโค้ดโปรแกรมในแต่ละช่วงเวลา ทำให้สามารถย้อนกลับไปยังเวอร์ชันก่อนหน้า, เปรียบเทียบความแตกต่าง, และทำงานร่วมกันได้อย่างเป็นระเบียบ ในโครงงานนี้ใช้ Git สำหรับบันทึกการแก้ไขโค้ดในแต่ละขั้นตอน ก่อนส่งขึ้นไปจัดเก็บบน GitHub



ภาพที่ 2.7 กิต (Git)

ที่มา <https://www.stickpng.com/img/icons-logos-emojis/tech-companies/git-full-logo>

### 2.3.8 กิตฮับ (GitHub)

บริการบนคลาวด์สำหรับเก็บซอร์สโค้ดโดยใช้ระบบของ Git ช่วยให้ผู้พัฒนาสามารถสำรองโค้ด, ติดตามประวัติการเปลี่ยนแปลง, และทำงานร่วมกับผู้อื่นได้อย่างเป็นระบบ ในโครงงานนี้ใช้ GitHub เป็นที่เก็บโค้ดหลักของโปรเจกต์ และเป็นศูนย์กลางในการสำรองข้อมูลโค้ดทั้งหมด



ภาพที่ 2.8 กิตฮับ (GitHub)

ที่มา <https://github.com/logos>

### 2.3.9 เอชทีเอ็มแอล (HTML)

ภาษาหลักที่ใช้ในการสร้างโครงสร้างของหน้าเว็บทั้งหมด เช่น การจัดวางหัวข้อ, เนื้อหา, ฟอรั่มสำหรับกรอกข้อมูล, และองค์ประกอบต่างๆ ในโครงงานนี้ HTML ทำหน้าที่เป็นโครงร่างของหน้าเว็บทุกหน้า ตั้งแต่หน้าล็อกอิน, หน้าส่งแควส, ไปจนถึงหน้าของผู้ดูแลระบบ เพื่อให้ข้อมูลถูกจัดวางอย่างเป็นระเบียบและเข้าถึงได้สะดวก



ภาพที่ 2.9 เอชทีเอ็มแอล (HTML)

ที่มา [https://commons.wikimedia.org/wiki/File:HTML5\\_logo\\_and\\_wordmark.svg](https://commons.wikimedia.org/wiki/File:HTML5_logo_and_wordmark.svg)

### 2.3.10 ซีเอสเอส (CSS)

ภาษาที่ใช้ในการกำหนดสไตล์และรูปแบบการแสดงผลของหน้าเว็บที่สร้างจาก HTML ในโครงงานนี้ CSS ถูกใช้เพื่อกำหนดธีมสีของระบบ (Dark Theme), กำหนดแบบอักษร, ระยะห่าง, และจัดวางเลย์เอาต์ของส่วนประกอบต่างๆ เช่น การ์ดแสดงแควส และปุ่ม ให้มีความสวยงามและสอดคล้องกันทั่วทั้งแอปพลิเคชัน



ภาพที่ 2.10 ซีเอสเอส (CSS)

ที่มา [https://commons.wikimedia.org/wiki/File:CSS3\\_logo\\_and\\_wordmark.svg](https://commons.wikimedia.org/wiki/File:CSS3_logo_and_wordmark.svg)

### 2.3.11 จาวาสคริปต์ (JavaScript)

ภาษาโปรแกรมที่ทำงานบนเบราว์เซอร์ของผู้ใช้ ใช้เพื่อให้หน้าเว็บสามารถโต้ตอบกับผู้ใช้ได้ ในโครงงานนี้ JavaScript มีบทบาทสำคัญในการจัดการส่วนที่ต้องมีการเปลี่ยนแปลงข้อมูลแบบไดนามิก เช่น การแสดงผลกราฟด้วย Chart.js, การแสดงตัวอย่างไฟล์ก่อนอัปโหลด, การเปิดหน้าต่าง Modal เพื่อแสดงกราฟของผู้ใช้ในหน้าแอดมิน, และการส่งคำขอเพื่อดึงข้อมูลจากเซิร์ฟเวอร์โดยไม่ต้องโหลดหน้าเว็บใหม่



ภาพที่ 2.11 จาวาสคริปต์ (JavaScript)

ที่มา <https://commons.wikimedia.org/wiki/File:Javascript-shield.svg>

## บทที่ 3

### วิธีการดำเนินงาน

#### 3.1 การศึกษาเบื้องต้น

##### 3.1.1 ระบบงานเดิม

จากการศึกษากระบวนการติดตามความก้าวหน้าทางการเรียนรู้และการพัฒนาทักษะที่ใช้อยู่ในปัจจุบัน พบว่าทั้งผู้เรียนและผู้สอน/ผู้ดูแล มักประสบปัญหาในการ รวบรวม, ประมวลผล, และแสดงผล ข้อมูลให้ครบถ้วนในทีเดียว โดยข้อมูลมักกระจุกกระจายอยู่ตามเครื่องมือ และช่องทางต่างๆ เช่น สมุดบันทึกส่วนตัว, ไฟล์สเปรดชีต, อีเมล, และแอปพลิเคชันสนทนา การใช้ งานจริงจึงต้องอาศัยการทำงานหลายขั้นตอน ผู้เรียนต้องรวบรวมไฟล์งานและส่งผ่านช่องทางที่ตกลง กันไว้ ในขณะที่ผู้ดูแลต้องเปิดดูงานจากหลายแหล่ง, ประเมินผล, และบันทึกคะแนนกลับลงในไฟล์ ของตนเอง ซึ่งกระบวนการเหล่านี้ส่งผลให้การให้ข้อเสนอแนะเกิดความล่าช้า, เสี่ยงต่อความ คลาดเคลื่อนของข้อมูล, และยากต่อการตรวจสอบประวัติย้อนหลัง

นอกจากนี้ ระบบงานเดิมยังมีข้อจำกัดในเชิงกระบวนการ คือ ขาดความเป็นศูนย์กลางใน การจัดเก็บข้อมูล ทำให้ไม่สามารถมองเห็นภาพรวมพัฒนาการของผู้เรียนแต่ละคนได้ทันที หาก ต้องการวิเคราะห์จุดแข็งหรือจุดอ่อนจำเป็นต้องนำข้อมูลจากหลายที่มาารวมกันด้วยตนเอง อีกทั้ง แพลตฟอร์มทั่วไปยังไม่ได้ถูกออกแบบมาเพื่อสร้างแรงจูงใจโดยตรง ขาดกลไกการให้รางวัลหรือการ แสดงผลความสำเร็จที่ชัดเจน ทำให้ผู้เรียนอาจรู้สึกว่าการทำภารกิจเป็นเพียงหน้าที่ที่ต้องทำให้เสร็จ ลินไป

จากปัญหาข้างต้น สามารถสรุปประเด็นสำคัญได้ว่า ระบบงานเดิมยังไม่ตอบโจทย์การเป็น "ศูนย์กลางข้อมูลเพื่อการพัฒนาตนเอง" ที่ผู้ใช้ต้องการ กล่าวคือยังขาดเครื่องมือที่สามารถบูรณาการ ขั้นตอนการส่งงาน, การประเมินผล, และการแสดงผลความก้าวหน้าเชิงภาพได้ในทีเดียว จึงเกิดความ จำเป็นในการพัฒนาระบบใหม่ที่รวมกระบวนการทั้งหมดเข้าด้วยกัน, ลดภาระงานที่ต้องทำด้วยมือ, ลดข้อผิดพลาด, และสนับสนุนการเรียนรู้ด้วยข้อมูลที่ชัดเจนและสร้างแรงจูงใจ

##### 3.1.2 ระบบงานใหม่

จากข้อจำกัดของระบบงานเดิม จึงได้ออกแบบและพัฒนา ระบบงานใหม่ Edu Quest เพื่อติดตามความก้าวหน้าทางการเรียนรู้ โดยรวมขั้นตอนการ ส่งภารกิจ (เควส), การ ตรวจสอบ, การให้คะแนน, และการแสดงผล ไว้ในเว็บแอปพลิเคชันเดียว

ระบบงานใหม่จะทำหน้าที่เป็นศูนย์กลาง โดยผู้ใช้จะส่งเควสพร้อมหลักฐานผ่านฟอร์มบน

หน้าเว็บ ข้อมูลทั้งหมดจะถูกบันทึกลงในไฟล์เจสัน (JSON) บนเซิร์ฟเวอร์พร้อมสถานะ "รอตรวจ" จากนั้นผู้ดูแลระบบจะเข้ามาตรวจสอบและประเมินผลผ่านหน้าแอปพลิเคชันสำหรับผู้ดูแล เมื่อผู้ดูแลทำการ อนุมัติ และกำหนดคะแนน ระบบจะทำการ อัปเดตสถานะคะแนนและคำนวณคะแนนสะสมของผู้ใช้โดยอัตโนมัติ ข้อมูลที่อัปเดตแล้วจะถูกบันทึกลงในไฟล์เจสันทันที ซึ่งทำหน้าที่เสมือนแคชข้อมูล ช่วยลดความซับซ้อนและเพิ่มความเร็วในการแสดงผลครั้งต่อไป

ผลลัพธ์จะถูกนำเสนอผ่านหน้าเว็บที่ผู้ใช้สามารถเห็นประวัติคะแนนของตนเอง และที่สำคัญคือ กราฟเรดาร์ ที่แสดงภาพรวม สถิติ หรือความสามารถในด้านต่างๆ ของตนเองได้อย่างชัดเจน ช่วยให้ผู้ใช้และผู้ดูแลใช้ ข้อมูลเชิงประจักษ์ที่ตรวจสอบได้ มาประกอบการวางแผนการเรียนรู้ได้ทันที ระบบนี้ช่วยลดภาระการติดตามข้อมูลที่กระจัดกระจาย, ลดความคลาดเคลื่อนจากการบันทึกด้วยมือ, และทำให้กระบวนการประเมินผลมีประสิทธิภาพ โปร่งใส และสร้างแรงจูงใจให้แก่ผู้เรียนได้ดียิ่งขึ้น

### 3.2 การกำหนดความต้องการของระบบ

การพัฒนาระบบติดตามความก้าวหน้า Edu Quest ต้องศึกษาความต้องการของเจ้าของระบบและผู้ใช้ระบบ เพื่อให้ได้ข้อกำหนดความต้องการ (Requirements) สำหรับใช้เป็นแนวทางในการพัฒนาระบบ

#### 3.2.1 ขอบเขตของระบบ

ระบบติดตามความก้าวหน้า Edu Quest มีขอบเขตการทำงานดังนี้

##### 3.2.1.1 ขอบเขตของระบบที่สามารถทำได้

- ก) ผู้ใช้สามารถเข้าสู่ระบบและส่งภารกิจ (คะแนน) พร้อมแนบไฟล์หลักฐานได้
- ข) ผู้ใช้สามารถดูประวัติและติดตามสถานะคะแนนของตนเองได้
- ค) ผู้ใช้สามารถดูค่าสถานะ (สถิติ) ของตนเองที่แสดงผลเป็นกราฟเรดาร์ได้
- ง) ผู้ดูแลระบบสามารถตรวจสอบ อนุมัติ หรือปฏิเสธคะแนนที่ผู้ใช้ส่งมาได้
- จ) ผู้ดูแลระบบสามารถให้คะแนนและบันทึกข้อเสนอแนะเพิ่มเติมสำหรับแต่ละคะแนนได้

- ฉ) ผู้ดูแลระบบสามารถเรียกดูข้อมูลกราฟสถิติของผู้ใช้คนใดก็ได้ในระบบ

##### 3.2.1.1 ขอบเขตของระบบที่ไม่สามารถทำได้

- ก) ระบบไม่รองรับการลงทะเบียนผู้ใช้ใหม่ด้วยตนเอง การเพิ่มผู้ใช้ต้องทำโดยผู้ดูแลระบบโดยตรงผ่านการแก้ไขไฟล์ข้อมูล
- ข) ระบบไม่สามารถดึงข้อมูลภารกิจหรือคะแนนจากระบบภายนอกอื่น ๆ (เช่น Google Classroom) ได้โดยอัตโนมัติ



ค) ระบบรองรับผู้ดูแลระบบได้เพียงระดับเดียว ไม่มีการแบ่งแยกสิทธิ์การเข้าถึงที่ซับซ้อน

### 3.2.2 ฮาร์ดแวร์ที่ใช้กับระบบงาน

ฮาร์ดแวร์สำหรับพัฒนาและใช้งานระบบแบ่งเป็น 3 ส่วน ดังนี้

#### 3.2.2.1 เครื่องพัฒนาภายใน

ใช้คอมพิวเตอร์ส่วนบุคคลระบบปฏิบัติการ ไมโครซอฟท์ วินโดวส์ 11 สำหรับเขียนโค้ดและทดสอบการทำงานของระบบ โดยมีการจำลองสภาพแวดล้อมของเว็บเซิร์ฟเวอร์ที่รองรับภาษา PHP

#### 3.2.2.2 เครื่องแม่ข่ายบนระบบคลาวด์

เปิดให้บริการเว็บแอปพลิเคชันผ่านผู้ให้บริการเว็บโฮสติ้งที่รองรับภาษา PHP และอนุญาตให้สคริปต์สามารถเขียนไฟล์ลงบนพื้นที่จัดเก็บได้ เพื่อใช้ในการบันทึกและอัปเดตข้อมูลไฟล์เจสัน (JSON) ของระบบ

#### 3.2.2.3 อุปกรณ์ของผู้ใช้ปลายทาง

ผู้ใช้สามารถเข้าถึงระบบผ่านเว็บเบราว์เซอร์มาตรฐานบนอุปกรณ์ต่างๆ หน้าเว็บถูกออกแบบให้รองรับหลายขนาดหน้าจอ (Responsive Design) เพื่อให้ใช้งานได้สะดวกทั้งบนคอมพิวเตอร์, โน้ตบุ๊ก, แท็บเล็ต, และโทรศัพท์มือถือ

### 3.2.3 ซอฟต์แวร์ที่ใช้กับระบบงาน

ซอฟต์แวร์ของระบบแบ่งออกเป็นหลายส่วนที่ทำงานร่วมกัน ดังนี้

3.2.3.1 ส่วนกลาง (Backend) พัฒนาระบบด้วยภาษา **PHP** เพื่อทำหน้าที่เป็นส่วนประมวลผลหลักฝั่งเซิร์ฟเวอร์ รับคำขอจากผู้ใช้งานผ่านเว็บเบราว์เซอร์, จัดการการเข้าสู่ระบบ (Session), ตรวจสอบสิทธิ์การเข้าถึง, ประมวลผลการส่งฟอร์ม, จัดการการอัปโหลดไฟล์ และที่สำคัญคือการอ่านและเขียนข้อมูลทั้งหมดของแอปพลิเคชันลงในไฟล์ข้อมูลรูปแบบ เจสัน (JSON) ซึ่งทำหน้าที่เป็นฐานข้อมูลของระบบ เช่น users.json, quests.json, และ stats.json

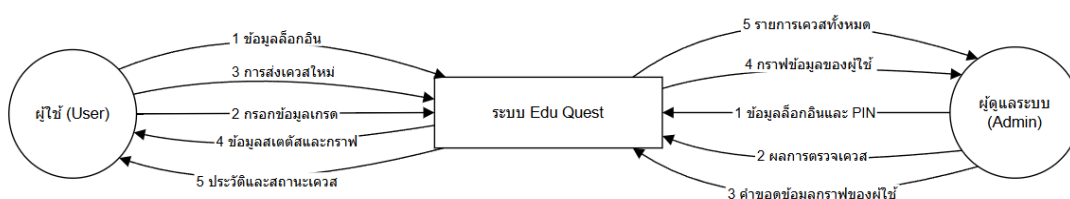
3.2.3.2 สำหรับการพัฒนาและดูแลโค้ด ใช้ วิวอล สตูดิโอ โค้ด (Visual Studio Code) เป็นเครื่องมือหลักในการเขียนโค้ด, จัดการเวอร์ชันของโค้ดด้วย กิต (Git) และจัดเก็บซอร์สโค้ดบน กิตฮับ (GitHub) เพื่อให้ง่ายต่อการติดตามและสำรองข้อมูล ส่วนการเปิดให้บริการจริงจะนำโค้ดทั้งหมดไปติดตั้งบนเว็บโฮสติ้งที่รองรับภาษา PHP

### 3.3 การออกแบบระบบ

การออกแบบระบบประกอบไปด้วยการออกแบบภาพรวมระบบ และการออกแบบส่วนติดต่อกับผู้ใช้

#### 3.3.1 การออกแบบแผนภาพลำดับการทำงานของระบบ

3.3.1.1 แผนภาพลำดับการทำงานของระบบ "Edu Quest" (Data Flow Diagram : Context Diagram)



ภาพที่ 3.1 ภาพรวมการทำงานของระบบ

จากภาพประกอบ ภาพรวมของระบบ "Edu Quest" ทำงานโดยมีผู้ที่มีส่วนเกี่ยวข้อง 2 กลุ่มหลัก คือ ผู้ใช้ (User) และ ผู้ดูแลระบบ (Admin) โดยมีลำดับการทำงานดังนี้

1) ผู้ใช้ และ ผู้ดูแลระบบ เข้าถึงระบบผ่าน ส่วนติดต่อผู้ใช้ (Frontend) ซึ่งเป็นหน้าเว็บที่แสดงผลผ่านเบราว์เซอร์ โดยผู้ใช้จะเข้าถึงหน้าส่งคะแนน (home.php) และผู้ดูแลจะเข้าถึงหน้าตรวจคะแนน (admin\_questions.php)

2) ส่วนติดต่อผู้ใช้ จะส่งคำขอ (Request) ไปยัง เซิร์ฟเวอร์ (Backend) เมื่อมีการกระทำเกิดขึ้น เช่น การล็อกอิน, การกรอกฟอร์มส่งคะแนน, หรือการกดปุ่มอนุมัติคะแนน

3) เซิร์ฟเวอร์ ซึ่งทำงานด้วยภาษา PHP ทำหน้าที่เป็นศูนย์กลางประมวลผลคำขอต่างๆ โดยไม่มีการดึงข้อมูลจากแหล่งภายนอก แต่จะทำงานกับ แหล่งจัดเก็บข้อมูลภายใน ซึ่งเป็นไฟล์เจสัน (JSON) โดยตรง

4) เซิร์ฟเวอร์ จะส่งผลลัพธ์กลับไปยังส่วนติดต่อผู้ใช้ ซึ่งมีสองรูปแบบหลัก คือ การแสดงหน้าเว็บใหม่พร้อมข้อความยืนยัน (เช่น "ส่งคะแนนสำเร็จ") หรือการส่งข้อมูลกลับในรูปแบบเจสัน (JSON) ในกรณีที่เป็นการเรียกใช้ API เฉพาะทาง (เช่น การขอดูกราฟของผู้ใช้ในหน้าแอดมิน)

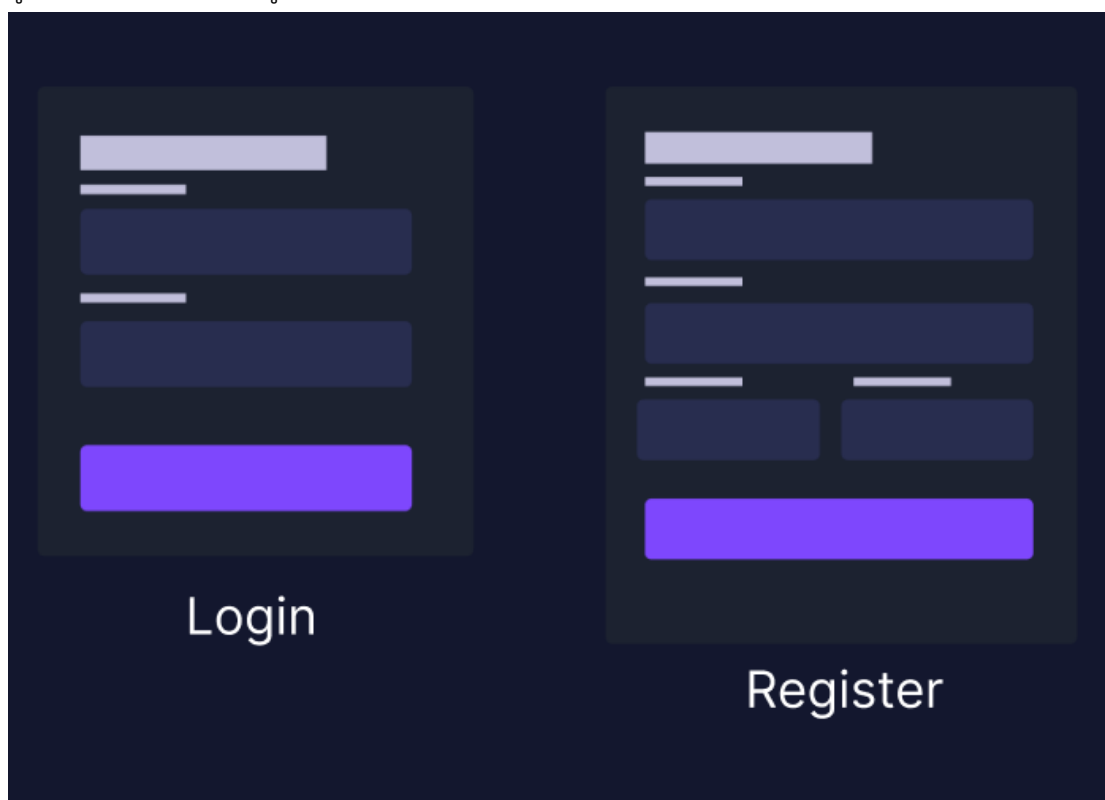
5) ส่วนติดต่อผู้ใช้ จะนำผลลัพธ์ที่ได้ไปแสดงผลให้ผู้ใช้เห็น โดยหากเป็นหน้าเว็บก็จะแสดงผลตามปกติ และหากเป็นข้อมูลเจสัน (JSON) ก็จะถูกนำไปใช้ในการวาดกราฟเรดาร์ด้วยไลบรารี Chart.js ให้ผู้ใช้เห็นเป็นภาพต่อไป

### 3.3.2 การออกแบบส่วนติดต่อกับผู้ใช้

การออกแบบโครงร่างหน้าจอของการพัฒนาระบบนี้ เป็นการออกแบบส่วนประกอบต่างๆ ของหน้าจอ รวมถึงตำแหน่งการจัดวาง เพื่อใช้เป็นแนวทางในการสร้างและพัฒนาเว็บแอปพลิเคชัน โดยมีรายละเอียดในการออกแบบหน้าจอหลัก 4 ส่วนดังต่อไปนี้

#### 3.3.2.1 ผลการออกแบบหน้าเข้าสู่ระบบและสมัครสมาชิก

หน้าเข้าสู่ระบบ (Login) ออกแบบให้มีช่องสำหรับกรอกข้อมูล 2 ช่อง คือ ชื่อผู้ใช้ (Username) และรหัสผ่าน (Password) พร้อมปุ่มสำหรับยืนยันการเข้าสู่ระบบหน้าสมัครสมาชิก (Register) ออกแบบให้มีช่องกรอกข้อมูลมากกว่าหน้าเข้าสู่ระบบ เพื่อใช้สำหรับลงทะเบียนผู้ใช้ใหม่ เช่น การตั้งชื่อผู้ใช้, รหัสผ่าน, และการยืนยันรหัสผ่าน ดังภาพที่ 3.2



ภาพที่ 3.2 หน้าการเข้าถึงระบบ

#### 3.3.2.2 ผลการออกแบบหน้ากรอกข้อมูล

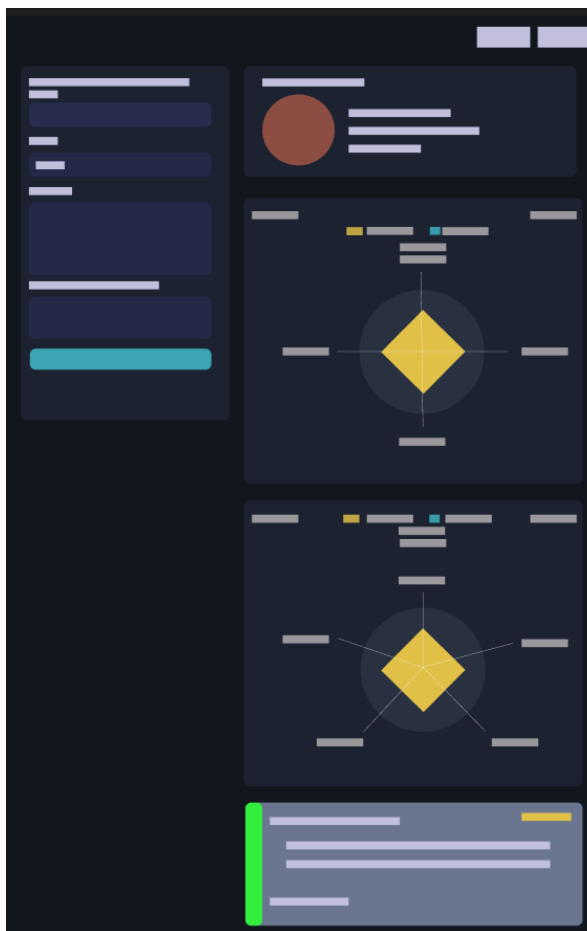
เป็นหน้าที่ออกแบบมาเพื่อให้ผู้ใช้สามารถกรอกข้อมูลผลการเรียน หรือข้อมูลที่ต้องการประเมินผลได้ หน้านี้ประกอบด้วยฟอร์มสำหรับกรอกข้อมูลทางด้านซ้าย และส่วนแสดงผลทางด้านขวา ซึ่งจะแสดงเป็น กราฟเรดาร์ (Radar Chart) เพื่อให้ผู้ใช้เห็นภาพรวมของข้อมูลที่กรอกเทียบกับเกณฑ์มาตรฐานหรือค่าเฉลี่ยได้ทันที ดังภาพที่ 3.3



ภาพที่ 3.3 หน้าผลการออกแบบหน้ากรอกข้อมูล

### 3.3.2.3 ผลการออกแบบหน้าจอหลัก

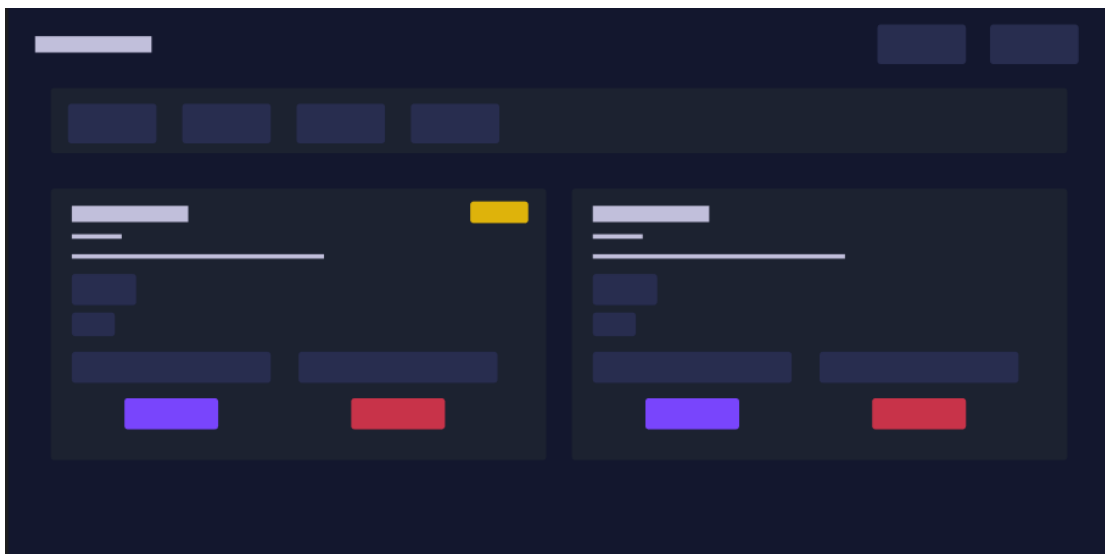
ส่วนข้อมูลผู้ใช้ (User Profile) แสดงข้อมูลเบื้องต้นของผู้ใช้ เช่น รูปโปรไฟล์และชื่อส่วนแสดงผลกราฟ แสดงกราฟเรดาร์ที่สรุปผลคะแนนหรือสถิติในด้านต่างๆ ของผู้ใช้เปรียบเทียบกับค่าเฉลี่ยรวม (คล้ายกับการแสดงผลในหน้ากรอกข้อมูล) เพื่อให้เห็นจุดแข็งหรือส่วนที่ต้องพัฒนาส่วนแจ้งเตือน/ข้อมูลล่าสุด (อ้างอิงจากภาพเวอร์ชันที่สอง) มีกล่องข้อความเพิ่มเติมที่ด้านล่างขวา ซึ่งออกแบบไว้สำหรับการแจ้งเตือนหรือกิจกรรมล่าสุด ดังภาพที่ 3.4



ภาพที่ 3.4 หน้าหน้าจอหลัก

### 3.3.2.3 ผลการออกแบบหน้าสำหรับผู้ดูแลระบบ

เป็นหน้าที่ออกแบบมาสำหรับผู้ดูแลระบบโดยเฉพาะ เพื่อใช้ในการจัดการข้อมูลต่างๆ ในระบบ การออกแบบใช้รูปแบบการ์ด (Card-based layout) ที่แสดงข้อมูลแต่ละรายการแยกกันอย่างชัดเจน ในแต่ละการ์ดจะมีปุ่มสำหรับการจัดการข้อมูล เช่น การแก้ไข (ปุ่มสีม่วง) หรือการลบ (ปุ่มสีแดง) และอาจมีป้ายกำกับ (Tag) เพื่อบอกสถานะของข้อมูลนั้นๆ ด้วย ดังภาพที่ 3.5



ภาพที่ 3.5 หน้าสำหรับผู้ดูแลระบบ

#### 1) หน้าดูกราฟสถิติของผู้ใช้

เมื่อผู้ดูแลระบบต้องการดูรายละเอียดข้อมูลเพิ่มเติมจากหน้าจอหลัก เช่น คลิกที่ส่วนแสดงผลกราฟ จะมีหน้าต่างป๊อปอัพ (Pop-up) แสดงขึ้นกลางจอ โดยมีรายละเอียดดังนี้ ดังภาพที่ 3.6



ภาพที่ 3.6 หน้าดูกราฟสถิติของผู้ใช้

### 3.4 การพัฒนาระบบ

ในการพัฒนาระบบจัดเก็บและวิเคราะห์ข้อมูล (Data Analysis System) นี้ ผู้พัฒนาได้กำหนดขั้นตอนการพัฒนาที่เน้นการจัดการข้อมูลที่ใช้ป้อนเข้ามา และการแสดงผลเชิงวิเคราะห์ เพื่อให้ผู้ใช้เห็นภาพรวมของตนเองได้ง่ายขึ้น ดังต่อไปนี้

3.4.1 เริ่มต้นจากการกำหนดกลุ่มผู้ใช้งานเป้าหมาย คือ ผู้ใช้งานทั่วไป ที่ต้องการบันทึกและติดตามผลงานหรือผลการเรียนของตนเอง และ ผู้ดูแลระบบ เพื่อระบุตัวชี้วัดที่สำคัญ เช่น ผลคะแนนเฉลี่ย, คะแนนในแต่ละหมวดหมู่ และกำหนดรูปแบบการแสดงผลที่ตอบโจทย์การใช้งาน โดยกำหนดขอบเขตของหน้าจอหลัก 4 ส่วน คือ หน้าเข้าสู่ระบบ/สมัครสมาชิก, หน้ากรอกข้อมูล, หน้าจอหลัก (แดชบอร์ด), และหน้าสำหรับผู้ดูแลระบบ

3.4.2 เนื่องจากระบบเน้นรับข้อมูลจากผู้ใช้โดยตรง จึงไม่ได้เชื่อมต่อกับแหล่งข้อมูลภายนอก แต่ได้ออกแบบฐานข้อมูลเพื่อจัดเก็บข้อมูลของผู้ใช้และข้อมูลผลการเรียนอย่างมีโครงสร้าง กำหนดรูปแบบการรับข้อมูลผ่านฟอร์มบนหน้าเว็บแอปพลิเคชัน และแปลงข้อมูลที่ได้รับเข้ามาให้อยู่ในโครงสร้างแบบ JSON เพื่อให้ง่ายต่อการนำไปประมวลผลต่อในส่วนหลังบ้าน

3.4.3 วางโครงสร้างระบบแบบ Client-Server โดยใช้ Python และเฟรมเวิร์ก Flask เป็นส่วน Backend ในการจัดการตรรกะของระบบและ API และใช้ HTML, CSS, และ JavaScript เป็นส่วน Frontend ในการแสดงผล ออกแบบโครงสร้างข้อมูล JSON สำหรับการแลกเปลี่ยนข้อมูลระหว่าง Client และ Server พร้อมทั้งกำหนดโครงสร้างสำหรับ ระบบแคช (Caching) เพื่อจัดเก็บผลลัพธ์ที่ประมวลผลแล้วชั่วคราว ช่วยลดภาระการประมวลผลซ้ำซ้อนเมื่อมีการเรียกดูข้อมูลเดิม

3.4.4 พัฒนา API และจุดให้บริการคำขอ (API Endpoints) ที่จำเป็น เช่น `/api/user/data` สำหรับจัดการข้อมูลของผู้ใช้ และ `/api/user/summary` สำหรับดึงข้อมูลสรุปเพื่อไปแสดงผลบนแดชบอร์ด นอกจากนี้ยังได้สร้างโมดูลสำหรับประมวลผลข้อมูลโดยเฉพาะ เช่น ฟังก์ชัน `calculate_average()` สำหรับคำนวณค่าเฉลี่ย, `format_data()` สำหรับจัดรูปแบบข้อมูลให้พร้อมแสดงผล เพื่อแปลงข้อมูลดิบจากฐานข้อมูลให้เป็นข้อมูลที่มีความหมายและพร้อมใช้งานในส่วนหน้า

3.4.5 สร้างหน้าเว็บเพื่อให้ผู้ใช้สามารถใช้งานระบบได้อย่างสะดวกด้วย **HTML** และ **Bootstrap 5** และใช้ **JavaScript** ร่วมกับไลบรารี **Chart.js** ในการแสดงผลข้อมูลเชิงกราฟิก ดังนั้นหน้าจอหลัก (Dashboard) แสดงผลข้อมูลสรุปของผู้ใช้ด้วย กราฟเรดาร์ (Radar Chart) เพื่อเปรียบเทียบผลงานในแต่ละด้านกับค่าเฉลี่ยโดยรวม หน้ากรอกข้อมูล มีการแสดงผลกราฟเรดาร์เช่นกัน เพื่อให้ผู้ใช้เห็นผลลัพธ์ทันทีหลังจากกรอกข้อมูล

### 3.5 การทดสอบระบบ

การทดสอบระบบถูกแบ่งออกเป็น 4 ระดับหลัก เพื่อให้มั่นใจว่าทุกส่วนประกอบของระบบทำงานได้อย่างถูกต้อง แม่นยำ และตอบสนองต่อการใช้งานได้อย่างมีประสิทธิภาพ โดยเน้นการตรวจสอบความถูกต้องของการประมวลผลข้อมูลที่ใช้บ่อยเข้ามา และการแสดงผลเชิงวิเคราะห์บนหน้าจอ

#### 3.5.1 การทดสอบหน่วย

การทดสอบนี้มุ่งเน้นการตรวจสอบความถูกต้องของฟังก์ชันแต่ละส่วนบน Backend เพื่อยืนยันความแม่นยำในการคำนวณและประมวลผลข้อมูล ทดสอบความถูกต้องของโมดูลคำนวณ: ตรวจสอบฟังก์ชัน `calculate_average()` ว่าสามารถคำนวณค่าเฉลี่ยจากชุดข้อมูลที่ได้รับได้อย่างถูกต้อง ทดสอบตรรกะการประมวลผล: ตรวจสอบฟังก์ชันเฉพาะทาง เช่น `get_grade_status()` เพื่อให้แน่ใจว่าสามารถประมวลผลและให้ผลลัพธ์ตามเงื่อนไขที่กำหนดได้อย่างแม่นยำ

#### 3.5.2 การทดสอบการบูรณาการ

การทดสอบนี้มุ่งเน้นการตรวจสอบการทำงานร่วมกันระหว่างโมดูลต่างๆ และการเชื่อมต่อกับฐานข้อมูล ทดสอบการทำงานของ API ตรวจสอบการเรียกใช้ API Endpoints หลัก เช่น `/api/user/data` เพื่อให้แน่ใจว่าระบบสามารถสร้าง, อ่าน, แก้ไข และลบข้อมูลในฐานข้อมูลได้อย่างถูกต้อง (CRUD Operations) ทดสอบกลไกแคช ทดสอบการทำงานร่วมกันระหว่างฟังก์ชัน `load_cache` และ `save_cache` กับฐานข้อมูล เพื่อตรวจสอบว่าระบบสามารถบันทึกและอ่านข้อมูล



จากแคชได้เมื่อข้อมูลยังไม่หมดอายุ และดึงข้อมูลจากฐานข้อมูลเพื่อประมวลผลใหม่เมื่อแคชหมดอายุแล้ว

### 3.5.3 การทดสอบระบบตั้งแต่ต้นจนจบ

การทดสอบนี้เป็นการจำลองการใช้งานจริง เพื่อตรวจสอบการทำงานของระบบทั้งหมดตั้งแต่ Frontend จนถึง Backend ทดสอบลำดับการใช้งานของผู้ใช้: ทดสอบตั้งแต่การสมัครสมาชิก, เข้าสู่ระบบ, กรอกข้อมูลผ่านฟอร์มในหน้า กรอกข้อมูล ไปจนถึงการแสดงผลข้อมูลบนหน้าจอหลัก ทดสอบการแสดงผลกราฟ: ตรวจสอบความถูกต้องของ กราฟเรดาร์ (Radar Chart) บนหน้าจอหลัก ว่าแสดงค่าสถิติของผู้ใช้เทียบกับค่าเฉลี่ยของระบบได้ตรงตามที่คำนวณไว้จริง การจัดการข้อผิดพลาด ทดสอบการแสดงผลสถานะ กำลังประมวลผล หรือ เกิดข้อผิดพลาด ในกรณีที่ผู้ใช้กรอกข้อมูลไม่ถูกต้อง หรือการเชื่อมต่อกับ Backend ล้มเหลว

### 3.5.4 การทดสอบประสิทธิภาพ

การทดสอบนี้มุ่งเน้นการประเมินความเร็วและความสามารถในการตอบสนองของระบบ ซึ่งเป็นปัจจัยสำคัญต่อประสบการณ์ของผู้ใช้ ทดสอบผลกระทบของแคช ประเมินเวลาในการตอบสนองเมื่อเปิด หน้าจอหลัก ซ้ำๆ เพื่อยืนยันว่ากลไกแคชช่วยลดเวลาในการโหลดข้อมูลและการประมวลผลลงได้อย่างมีนัยสำคัญ ทดสอบความเร็วในการโหลด ทดสอบความเร็วในการโหลดหน้าเว็บหลักๆ เช่น หน้าจอหลัก และหน้ากรอกข้อมูล บนอุปกรณ์ที่แตกต่างกัน (เดสก์ท็อปและมือถือ) เพื่อให้มั่นใจว่าผู้ใช้จะได้รับประสบการณ์ที่รวดเร็วและราบรื่น

## บรรณานุกรม

ชาญวิทย์ เทพารักษ์. (2566). การพัฒนาระบบแดชบอร์ดสำหรับวิเคราะห์และแสดงผลข้อมูลผู้ใช้ผ่านเว็บแอปพลิเคชัน (Development of a Dashboard System for Analyzing and Visualizing User Data via Web Application) [สารนิพนธ์วิทยาศาสตรมหาบัณฑิต, มหาวิทยาลัยเกษตรศาสตร์]

สิริพร วงศ์สวัสดิ์. (2565). การสร้าง Backend Service ด้วย Flask Framework เพื่อรองรับแอปพลิเคชันวิเคราะห์ข้อมูล (Building a Backend Service with the Flask Framework to Support a Data Analysis Application) [สารนิพนธ์วิศวกรรมศาสตรมหาบัณฑิต, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง]