

รายงานข้อเสนอโครงงานวิศวกรรมคอมพิวเตอร์ วิชา 01076014 การเตรียมโครงงานวิศวกรรมคอมพิวเตอร์ ภาคเรียนที่ 2 ปีการศึกษา 2564

1.	ชื่อหัวข้อโครงงาน (ไทย) <u>โปรแกรมสร้างเว็บส่วนหน้า</u>									
2.	ชื่อหัวข้อโครงงาน (อังกฤษ) <u>Web Frontend Builder</u>									
3.	Keyword 3 คำ <u>Low-Code, Front-End Builder, API Tester</u>									
4.	ประเภทโครงงาน (X)									
	☐ 1. HW+SW 2. SW_Dev	☐ 3. Research								
5.	รายชื่อผู้ทำโครงงาน									
	5.1. นาย <u>วันบุรฮาน แวอุเซ็ง</u>	รหัส <u>64015130</u>								
	5.2. นาย อรรณพ ธรรมเที่ยงธรรม	รหัส 64015166								
	5.3. นาย <u>เอกรินทร์ องอาจ</u>	รหัส 64015172								
6.	อาจารย์ที่ปรึกษา									
	6.1. อาจารย์ที่ปรึกษาหลัก <u>ผศ ธนา หงษ์สุวรรณ</u>									
	6.2. อาจารย์ที่ปรึกษาร่วม <u>ผศ.ดร. ชมพูนุท เต็งเจริญ</u>									

1. ที่มาและความสำคัญของปัญหา (Motivation)

การสร้างเว็บไซต์ในปัจจุบันมีความสำคัญอย่างมากในการพัฒนาธุรกิจและการสร้างสรรค์ผลงานใหม่ ๆ อย่างไร ก็ตาม ผู้ใช้ที่ไม่มีความรู้และทักษะด้านการเขียนโปรแกรม และพบว่าเข้าถึงและสร้างเว็บไซต์เหล่านั้นเป็นเรื่องที่ ยากลำบากและซับซ้อน สิ่งนี้อาจทำให้หลายๆคนรู้สึกว่าไม่สามารถเข้าถึงเทคโนโลยีได้ นอกจากนี้ยังมีผู้ใช้ที่สนใจ ในการสร้างเว็บไซต์เพื่อใช้ในธุรกิจของตนเอง แต่ขาดความรู้และทักษะในการเขียนโค้ด นั่นทำให้ประสบปัญหาใน การดำเนินการและการพัฒนาซอฟต์แวร์ที่เหมาะสม

ด้วยเหตุนี้เองที่แพลตฟอร์ม Low-Code กลายเป็นทางเลือกที่ดีและแก้ไขปัญหาดังกล่าวได้อย่างสมบูรณ์ แพลตฟอร์มดังกล่าวอนุญาตให้ผู้ใช้ที่ไม่มีความรู้และทักษะด้านการเขียนโปรแกรมสร้างเว็บไซต์ได้อย่างง่ายดาย โดยมีการใช้เครื่องมือและกราฟิกอินเตอร์เฟซเพื่อสร้างและแก้ไขฟังก์ชันต่าง ๆ ของเว็บไซต์ ผู้ใช้สามารถสร้าง เว็บไซต์ได้โดยไม่จำเป็นต้องเขียนโค้ดเอง ทำให้แพลตฟอร์มดังกล่าวเป็นเครื่องมือที่สามารถนำไปใช้ในธุรกิจ ส่วนตัวหรือการสร้างสรรค์ได้หลากหลายและยืดหยุ่นมากขึ้น

2. วัตถุประสงค์ (Objectives)

- 1. เพื่อลดความซับซ้อนในกระบวนการสร้างเว็บไซต์ ผู้ใช้สามารถใช้เครื่องมือและกราฟิกอินเตอร์เฟซที่ใช้งาน ง่ายในการสร้างและแก้ไข ทำให้ไม่จำเป็นต้องมีความรู้เชิงลึกในการเขียนโค้ด
- 2. เพื่อประหยัดเวลาและทรัพยากรในการพัฒนา การใช้แพลตฟอร์ม Low-Code ช่วยลดเวลาในกระบวนการ สร้างและปรับปรุงเว็บไซต์ โดยทำให้ผู้ใช้สามารถกำหนดและปรับเปลี่ยนสิ่งต่าง ๆ ได้อย่างรวดเร็ว
- 3. เพื่อทดสอบการใช้งาน API ของตนเองในแพลตฟอร์ม Low-Code ผู้ใช้สามารถเชื่อมต่อและใช้งาน API จาก แหล่งที่มาต่าง ๆ เพื่อเพิ่มฟังก์ชันและความสามารถของเว็บไซต์

3. ทฤษฎีที่เกี่ยวข้อง (Theoretical Background)

3.1 แนวคิดของ Low-Code Platform

3.1.1. Low-Code Development

เป็นแนวคิดที่เน้นการใช้เครื่องมือและอินเทอร์เฟซกราฟิกในการสร้างเว็บไซต์ โดยลดระยะเวลาและ ความซับซ้อนในกระบวนการพัฒนา ผู้ใช้สามารถสร้างเว็บไซต์ได้โดยใช้การลากและวางองค์ประกอบที่พร้อม ใช้งาน และกำหนดค่าต่าง ๆ ที่ต้องการ โดยไม่ต้องเขียนโค้ดเป็นจำนวนมาก

3.1.2. Visual Programming

เป็นแนวคิดที่ใช้กราฟิกหรืออินเทอร์เฟซสำหรับการเขียนโปรแกรมแทนการเขียนโค้ดด้วยภาษา โปรแกรมมิ่งแบบแถวบรรทัดเดียว ผู้ใช้สามารถสร้างโปรแกรมโดยการเชื่อมต่อองค์ประกอบและตั้งค่าผ่าน กราฟิกหรืออินเทอร์เฟซที่ใช้งานง่าย ทำให้ผู้ใช้ที่ไม่มีความรู้ด้านการเขียนโค้ดสามารถสร้างเว็บไซต์ได้อย่าง รวดเร็ว

3.1.3. Model-Driven Development

เป็นวิธีการพัฒนาซอฟต์แวร์ที่เน้นการสร้างและใช้งานโมเดล (Models) เป็นส่วนสำคัญใน กระบวนการพัฒนา แพลตฟอร์ม Low-Code ใช้แนวคิด MDD ในการสร้างเว็บไซต์ โดยผู้ใช้สร้างและกำหนด โมเดลของเว็บไซต์ในรูปแบบที่เข้าใจง่าย โดยรวมถึงการกำหนดคุณสมบัติ และความสัมพันธ์ระหว่าง องค์ประกอบต่างๆ ในโมเดล ซึ่งแพลตฟอร์มจะทำการสร้างโค้ดอัตโนมัติจากโมเดลที่กำหนด เพื่อให้ผู้ใช้ไม่ ต้องเขียนโค้ดเอง

3.1.4. Rapid Application Development

เป็นวิธีการพัฒนาซอฟต์แวร์ที่ให้ความสำคัญกับการสร้างเว็บไซต์ในเวลาที่รวดเร็ว แพลตฟอร์ม Low-Code เน้นการพัฒนาเว็บไซต์อย่างรวดเร็วด้วยการใช้งานเครื่องมือและอินเทอร์เฟซที่ง่ายต่อการสร้าง รวมถึงการทำงานร่วมกับทีมผู้พัฒนาอื่นๆ เพื่อลดระยะเวลาในการพัฒนาและปรับปรุงเว็บไซต์

3.1.5. Integration and Connectivity

แพลตฟอร์ม Low-Code สามารถรองรับการเชื่อมต่อและการเชื่อมโยงกับระบบหรือบริการอื่นๆ ผ่าน API (Application Programming Interface) ที่มีอยู่แล้ว ซึ่งช่วยให้ผู้ใช้สามารถใช้ประโยชน์จากการ นำเอาฟังก์ชันหรือข้อมูลจากแหล่งที่มาต่างๆ

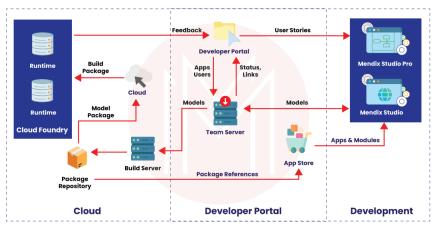
3.1.6. Application Programming Interface

API เป็นช่องทางที่ใช้ในการเชื่อมต่อระบบหรือบริการต่างๆ เพื่อแลกเปลี่ยนข้อมูลหรือฟังก์ชัน ระหว่างเว็บไซต์และแหล่งข้อมูลหรือบริการอื่นๆ ผู้ใช้สามารถนำ API มาใช้ในแพลตฟอร์ม Low-Code เพื่อ เชื่อมต่อกับฐานข้อมูลภายนอก บริการคลังข้อมูลออนไลน์ ระบบชำระเงิน หรือแหล่งข้อมูลอื่นๆ เพื่อเพิ่ม ประสิทธิภาพและความสามารถของเว็บไซต์

4. งานวิจัยที่เกี่ยวข้อง (Related Works)

4.1 Mendix

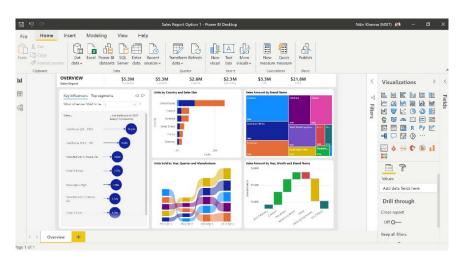
Mendix เป็นแพลตฟอร์ม Low-Code ที่ให้ผู้ใช้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่ายดาย มีการใช้งาน แบบลากและวางที่เป็นเครื่องมือในการสร้างและแก้ไขฟังก์ชันต่าง ๆ ของแอปพลิเคชัน ผู้ใช้สามารถเลือกใช้ตัว ควบคุมที่กำหนดไว้ล่วงหน้าและทำงานร่วมกับฐานข้อมูลในการสร้างแอปพลิเคชันที่มีประสิทธิภาพ Mendix ยังมี เครื่องมือสำหรับการจัดการโครงการและการทำงานร่วมกันของทีมผู้พัฒนาที่ช่วยให้งานเป็นไปอย่างราบรื่น



รูป 4.1 โครงสร้างการทำงานของ Mendix

4.2 Microsoft Power Apps

Microsoft Power Apps เป็นแพลตฟอร์ม Low-Code ที่อนุญาตให้ผู้ใช้สร้างแอปพลิเคชันในองค์กรได้อย่าง รวดเร็วและง่ายดาย ผู้ใช้สามารถสร้างแอปพลิเคชันโดยใช้เครื่องมือแบบลากและวางเพื่อสร้างและปรับแต่งฟอร์ม, ฐานข้อมูล, และเครื่องมืออื่น ๆ นอกจากนี้ยังมีการใช้งานอินเตอร์เฟซกับบริการอื่น ๆ ของ Microsoft เช่น SharePoint, Microsoft Teams, และ Dynamics 365 เพื่อเพิ่มฟังก์ชันและความสามารถของแอปพลิเคชัน



รูป 4.2 หน้าต่างการทำงานของ Microsoft Power Apps

5. ขอบเขตของโครงงาน (Scope)

5.1 สามารถพัฒนาและสร้างแพลตฟอร์มที่ช่วยให้ผู้ใช้สามารถสร้างเว็บไซต์เว็บได้โดยไม่ต้องมีความรู้เทคนิคด้าน การเขียนโปรแกรมเป็นพิเศษ

- 5.2 สามารถใช้งานฟีเจอร์พื้นฐานที่จำเป็นสำหรับการสร้างเว็บไซต์ เช่น การสร้างแบบฟอร์มในการป้อนข้อมูล การจัดการฐานข้อมูลพื้นฐาน การจัดการผู้ใช้งาน และการจัดการสิทธิ์การเข้าถึงข้อมูล
- 5.3 สามารถเชื่อมต่อกับ API ภายนอกเพื่อให้ผู้ใช้สามารถนำเอาข้อมูลหรือบริการจากแหล่งข้อมูลภายนอกมาใช้ ในเว็บไซต์ที่พัฒนาได้
- 5.4 สามารถปรับแต่งและปรับเปลี่ยนรูปแบบของเว็บไซต์ได้อย่างยืดหยุ่น ผู้ใช้สามารถปรับเปลี่ยนเค้าโครงหรือ รูปแบบทั้งหมดของเว็บไซต์ได้ตามต้องการ ได้รวมถึงการปรับแต่งสี รูปแบบเลย์เอาต์ และส่วนประกอบอื่นๆ
- 5.5 สามารถตรวจสอบความถูกต้องและการทดสอบเบื้องต้น ผู้ใช้สามารถตรวจสอบการทำงานของเว็บไซต์และ ทดสอบความถูกต้องของฟังก์ชันพื้นฐานได้ก่อนการเผยแพร่
- 5.6 สามารถให้ผู้ใช้สร้างหน้าเว็บไซต์หลายหน้าได้ โดยผู้ใช้สามารถสร้างหน้าเว็บไซต์ใหม่และกำหนดลำดับหน้า เว็บไซต์ได้
- 5.7 สามารถให้ผู้ใช้สร้างเมนูและการนำทางในเว็บไซต์ได้ โดยให้ผู้ใช้สร้างเมนูหลักและเมนูย่อย และกำหนดลำดับ การเรียงลำดับของเมนูได้

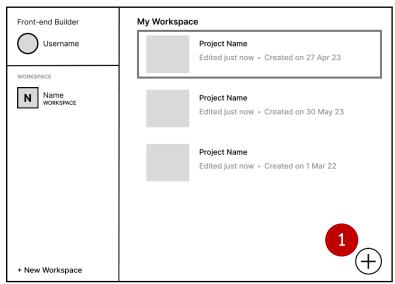
6. การพัฒนาโครงงาน (Project Development)

6.1 ขั้นตอนการพัฒนา (Methodology)

- 1. กำหนดวัตถุประสงค์ ขอบเขต และเป้าหมายของโครงงาน
- 2. สืบค้น และศึกษางานวิจัยที่เกี่ยวข้อง
- 3. ศึกษาการใช้งานของเครื่องมือที่จะใช้ในการดำเนินงาน
- 4. ออกแบบรวบรวมความต้องการของผู้ใช้ (Requirement)
- 5. ออกแบบโครงร่างส่วนติดต่อผู้ใช้ (Wireframe)
- 6. ออกแบบระบบฐานข้อมูล (Database)
- 7. ออกแบบส่วนติดต่อผู้ใช้ (User Interface)
- 8. ออกแบบโครงสร้างการทำงาน (Architecture)
- 9. พัฒนาระบบส่วน Front-End
- 10. พัฒนาระบบส่วน Back-End
- 11.เชื่อมต่อส่วนของ Front-End เข้ากับ Back-End
- 12. ทดสอบประสิทธิภาพการทำงาน และปรับปรุง
- 13. สรุปผลการดำเนินงาน

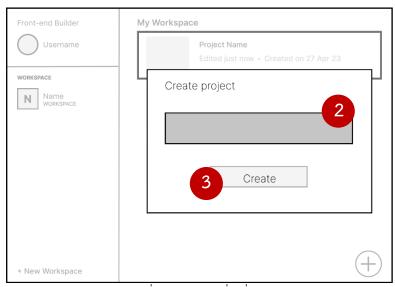
6.2 การออกแบบ (Design)

6.2.1. ขั้นตอนการใช้งานเว็บไซต์เบื้องต้น ให้แสดงสินค้าผ่าน API Method GET และเพิ่มสินค้าในตะกร้า โดยใช้ API Method POST เพื่อเพิ่มข้อมูลตะกร้าสินค้าลง Database(Cart)



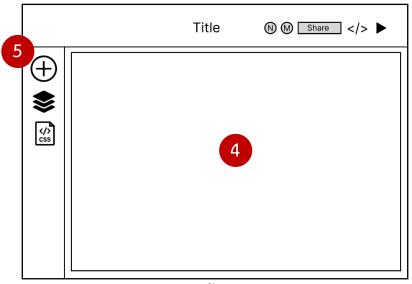
รูป 6.2.1.1 สร้าง Project

1. กดเมนูสร้าง Project เพื่อเริ่มสร้าง Project หน้าเปล่า



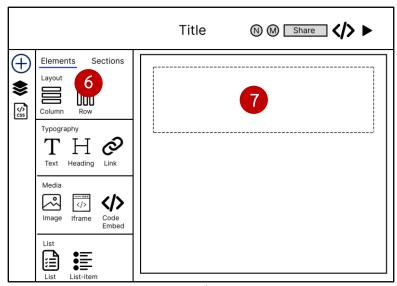
รูป 6.2.1.2 ใส่ชื่อ Project เพื่อเริ่มสร้าง Project

- 2. ใส่ชื่อ Project ที่ต้องการสร้าง
- 3. กด Create เพื่อสร้าง Project



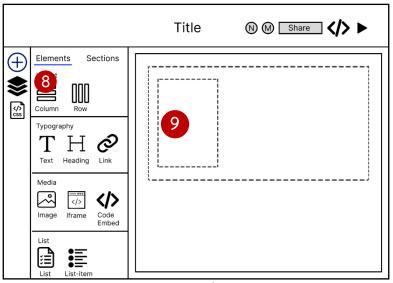
รูป 6.2.1.3 หน้าต่างการทำงาน

- 4. พื้นที่การทำงานของเว็บไซต์ (Background Default : White)
- 5. เลือกเมนู Element เพื่อเปิดแถบเมนู Elements เพิ่มเติม



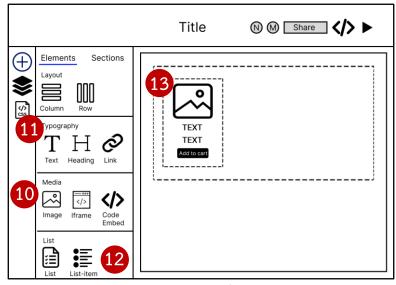
รูป 6.2.1.4 ใช้งาน Row เพื่อจัด Layout บนเว็บไซต์

- 6. กด Row ค้าง แล้วลากบนพื้นที่เว็บไซต์
- 7. หลังจากวาง Row เพื่อจัด Layout จะปรากฏเส้นประ เพื่อแสดงขนาด Layout



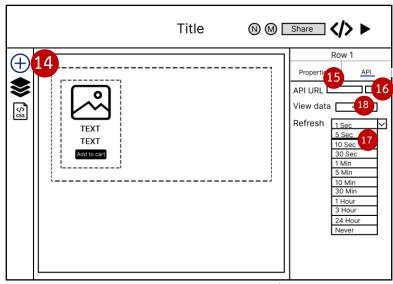
รูป 6.2.1.5 ใช้งาน Column เพื่อจัด Layout บนเว็บไซต์

- 8. กด Column ค้าง แล้วลากบนพื้นที่เว็บไซต์
- 9. หลังจากวาง Column เพื่อจัด Layout แนวตั้ง จะปรากฏเส้นประ เพื่อแสดงขนาด Layout



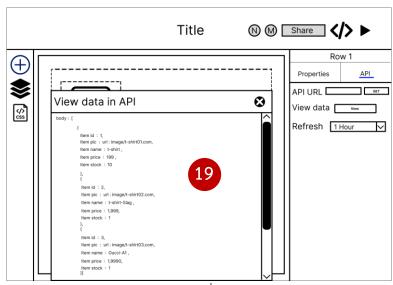
รูป 6.2.1.6 ใส่ Element ในเว็บไซต์ไว้แสดงข้อมูล

- 10. กด Image ค้าง แล้วลากใส่ Layout Column ที่จัดไว้ (หมายเลข 13)
- 11. กด Text ค้าง แล้วลากใส่ Layout Column ที่จัดไว้ (หมายเลข 13)
- 12. กด Button ค้าง แล้วลากใส่ Layout Column ที่จัดไว้ (หมายเลข 13)
- 13. Layout Column ที่ภายในนั้นประกอบไปด้วย Image, Text และ Button



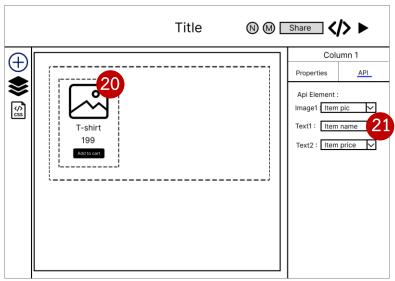
รูป 6.2.1.7 ใส่ข้อมูล API ให้ Row และดูข้อมูลที่ได้มาจากการ GET

- 14. กดที่ layout Row ที่เป็นเส้นประ แล้วแถบ Properties จะแสดงขึ้นมาด้านข้าง
- 15. ใส่ลิงก์ API ที่ต้องการดึงมาเก็บไว้
- 16. กดปุ่ม GET เพื่อดึงข้อมูล
- 17. เลือกเวลารีเฟรชข้อมูล
- 18. กดปุ่ม View Data เพื่อแสดงข้อมูลที่ดึงมาจาก API



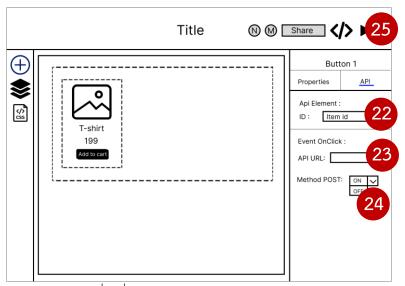
รูป 6.2.1.8 แสดงข้อมูลที่ได้จากการ GET API

19. แสดงข้อมูลที่ดึงมาจาก API



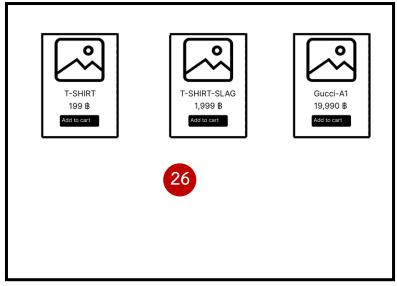
รูป 6.2.1.9 เลือกข้อมูลไปแสดงใน Elements

- 20. กดที่ Layout Column เพื่อเปิดแถบ API
- 21. เลือกข้อมูลจาก API มาแสดงผล (ข้อมูลจะแสดงผลตาม ข้อมูลที่ส่งมากับ API)



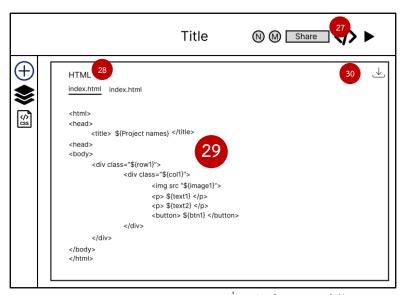
รูป 6.2.1.10 กดปุ่มเพื่อเพิ่มข้อมูลตะกร้าสินค้าในฐานข้อมูลด้วยการ POST API

- 22. เลือก id ของ item เพราะ id จะระบุความเป็นเจ้าของของแต่ละปุ่มนั้น
- 23. ใส่ลิงก์ API เพื่อไว้ POST ข้อมูลลง Database
- 24. ใน Event OnClick บน Button ให้ตั้งค่า Post เป็น ON เพื่อส่งข้อมูลสินค้าที่ต้องการเพิ่มลงตะกร้า ใน Database
- 25. กดเพื่อ Preview เว็บไซต์ที่ทำโครงร่างไว้



รูป 6.2.11 พรีวิวเว็บไซต์

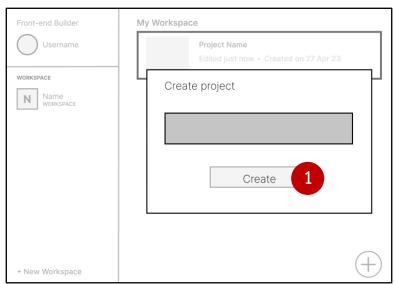
26. พรีวิวเว็บไซต์โดยเปิดแท็บใหม่บนบราวเซอร์ และแสดงผลการเรียกข้อมูลตามจำนวนสินค้าที่มีอยู่ใน API



รูป 6.2.11 กดปุ่ม Tag Code เพื่อพรีวิวในรูปแบบโค้ด

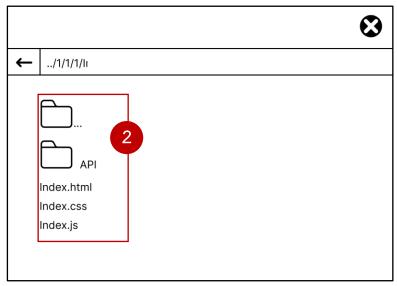
- 27. กดปุ่มเพื่อพรีวิวโค้ด
- 28. เลือกนามสกุลไฟล์โค้ดที่จะพรีวิว (.html, .css, .js)
- 29. แสดงโค้ดทั้งหมดจากนามสกุลไฟล์ที่เลือก
- 30. กดปุ่มเพื่อดาวน์โหลดโค้ดที่พรีวิวมาลงเครื่อง

6.2.2 การเปลี่ยนแปลงภายในไฟล์โปรเจ็ค เมื่อใช้งาน Elements ต่าง ๆ



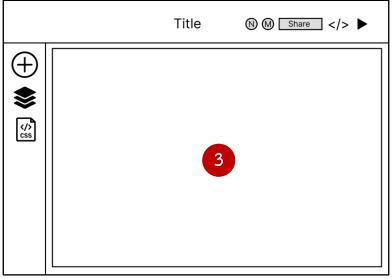
รูป 6.2.2.1 เริ่มการสร้างโปรเจ็ค

1. เริ่มสร้างโปรเจ็ค



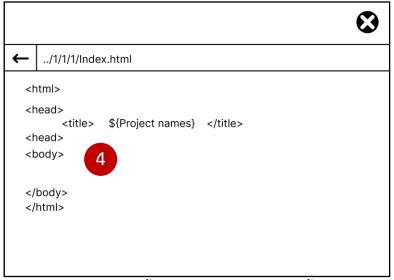
รูป 6.2.2.2 ไฟล์ต่าง ๆ ในโปรเจ็ค

2. ระบบจะสร้างไฟล์ต่างๆ ที่ใช้ในโปรเจ็คภายใต้ที่อยู่ของ ../User_id/workspace_id/project_id



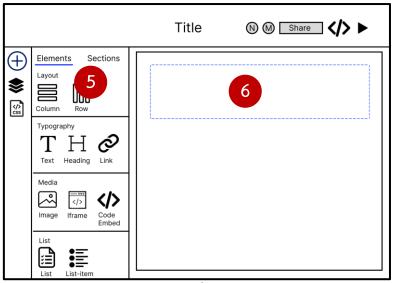
รูป 6.2.2.3 หน้าต่างการทำงาน

3. พื้นที่การทำงานของเว็บไซต์ (Background Default : White)



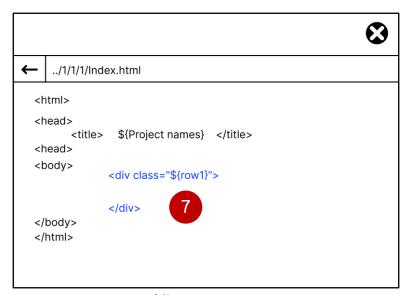
รูป 6.2.2.4 โค้ดพื้นฐานจากการสร้างโปรเจ็คครั้งแรก

4. นี่คือโค้ดเริ่มต้นที่อยู่ในไฟล์ Index.html ที่สร้างขึ้นมาพร้อมสร้างโปรเจ็ค



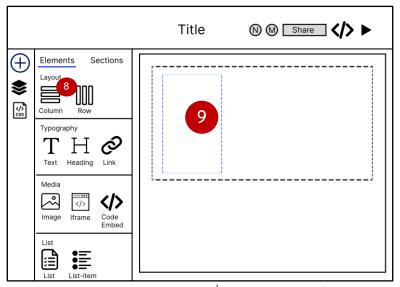
รูป 6.2.2.5 ใช้งาน Row เพื่อจัด Layout บนเว็บไซต์

- 5. กด Row ค้าง แล้วลากบนพื้นที่เว็บไซต์
- 6. หลังจากวาง Row เพื่อจัด Layout จะปรากฏเส้นประ เพื่อแสดงขนาด Layout



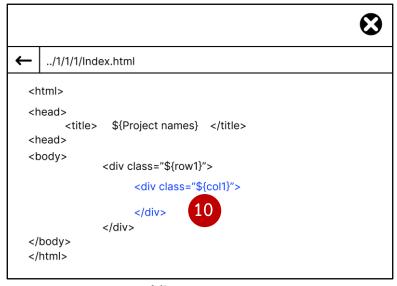
รูป 6.2.2.6 โค้ดหลังจากวาง Layout Row

7. หลังจากวาง Layout Row บนพื้นที่การทำงาน และ Tag <div class="\${row1}"> จะถูกเพิ่มใน ไฟล์ Index.html ภายใต้ Tag <body>



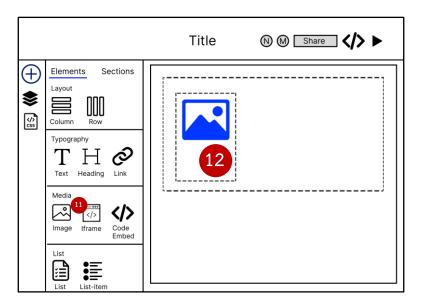
รูป 6.2.2.7 ใช้งาน Column เพื่อจัด Layout บนเว็บไซต์

- 8. กด Column ค้าง แล้วลากบนพื้นที่เว็บไซต์
- 9. หลังจากวาง Column เพื่อจัด Layout แนวตั้ง จะปรากฏเส้นประ เพื่อแสดงขนาด Layout



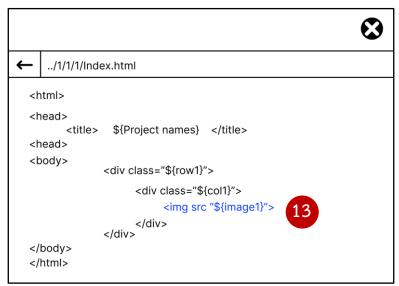
รูป 6.2.2.8 โค้ดหลังจากวาง Layout Col

10. หลังจากวาง Layout Col บนพื้นที่การทำงาน และ Tag <div class="\${col1}"> จะถูกเพิ่มใน ไฟล์ Index.html ภายใต้ Tag <div class="\${row1}">



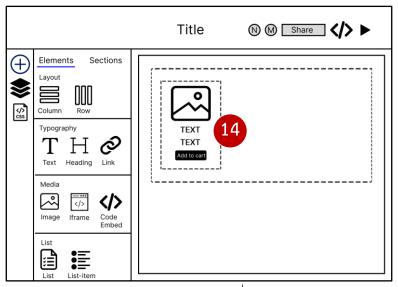
รูป 6.2.2.9 ใส่ Element Image

- 11. กด Image ค้าง แล้วลากใส่ Layout Column ที่จัดไว้ (หมายเลข 12)
- 12. Layout Column ภายในนั้นประกอบไปด้วย Image



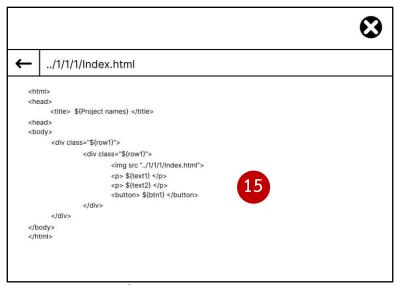
รูป 6.2.2.10 โค้ดหลังจากวาง Element Image

13. หลังจากวาง Element Image บนพื้นที่การทำงาน และ Tag จะถูกเพิ่ม ในไฟล์ Index.html ภายใต้ Tag <div class="\${col1}">



รูป 6.2.2.11 วาง Element ที่ใช้งานจนครบ

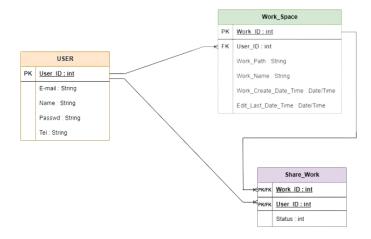
14. ใส่ Elements ที่ใช้งานให้ครบ



รูป 6.2.2.11 โค้ดทั้งหมดหลังจากวาง Element ที่ต้องการครบ

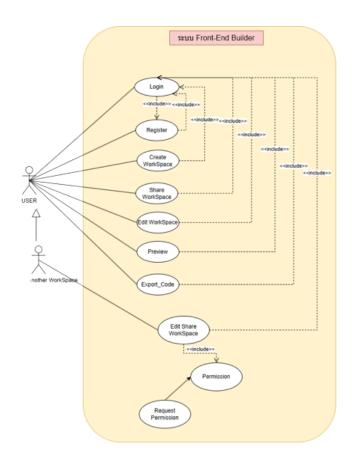
15. โค้ดทั้งหมดหลังจากวาง Element ที่ใช้งานบนพื้นที่เว็บไซต์

6.2.3 ออกแบบ Database ที่ใช้งานภายในเว็บไซต์



Database นี้ที่ใช้งานจะใช้เก็บข้อมูลผู้ใช้ และที่เก็บไฟล์งานโปรเจ็ค โดยที่
1 user สามารถมี workspace ได้ 1 workspace และ 1 Workspace สามารถ มีได้หลายโปรเจ็ค
ในส่วนของพื้นที่เก็บของโฟลเดอร์การทำงานจะเก็บอยู่ในรูปแบบเส้นทาง (Path) ตัวอย่าง
.../User_id/workspace_id/project_id/

6.2.4 ออกแบบ Use Case Diagram



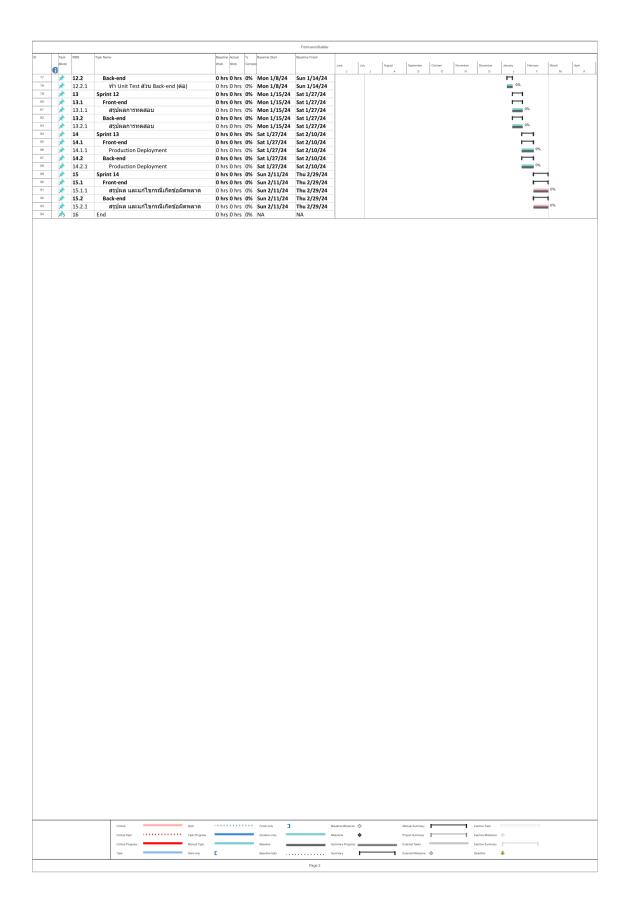
6.3 แนวทางการทดสอบและการวัดประสิทธิภาพ (Test and Performance Evaluation Approaches)

6.3.1. การทดสอบการสร้างเว็บไซต์โดยผู้ใช้ที่ไม่มีความรู้และทักษะด้านการเขียนโปรแกรม ให้สังเกตการ ทำงานของเครื่องมือและกราฟิกอินเตอร์เฟซว่าง่ายต่อการใช้งานหรือไม่ และผลลัพธ์ที่ได้สอดคล้องกับ ความต้องการหรือไม่

- 6.3.2. การทดสอบการปรับแก้ไขและปรับปรุงเว็บไซต์ที่สร้างขึ้นด้วยแพลตฟอร์ม Low-Code โดยทดสอบ ความสามารถในการเปลี่ยนแปลงรูปแบบและฟังก์ชันของเว็บไซต์ และตรวจสอบว่าการแก้ไขสามารถ ทำได้อย่างรวดเร็วและสะดวกสบายหรือไม่
- 6.3.3. การทดสอบการเชื่อมต่อและการใช้งาน API จากแหล่งที่มาต่าง ๆ เพื่อตรวจสอบความสามารถในการ สื่อสารและรับส่งข้อมูลระหว่างแพลตฟอร์ม Low-Code กับแหล่งข้อมูลภายนอก ทดสอบการเรียกใช้ API และตรวจสอบว่ารับข้อมูลอย่างถูกต้องและแสดงผลได้ถูกต้องหรือไม่
- 6.3.4. การทดสอบประสิทธิภาพการทำงานของเว็บไซต์ที่สร้างขึ้นด้วยแพลตฟอร์ม Low-Code โดยใช้ข้อมูล ประสิทธิภาพที่สำคัญ เช่น เวลาที่ใช้ในการโหลดหน้าเว็บไซต์ การตอบสนองของหน้าเว็บไซต์ และ ประสิทธิภาพในการดำเนินการต่าง ๆ ซึ่งสามารถทำได้โดยใช้เครื่องมือและเทคนิคต่าง ๆ เช่น
 - 6.3.4.1. การทดสอบประสิทธิภาพของเว็บไซต์ (Website Performance Testing): ใช้เครื่องมือและ เทคนิคที่ช่วยวัดประสิทธิภาพของเว็บไซต์ เช่น Load Testing เพื่อตรวจสอบความทนทานของ ระบบในการรับข้อมูลจำนวนมาก หรือ Stress Testing เพื่อทดสอบประสิทธิภาพของระบบใน เงื่อนไขการใช้งานที่มีความซับซ้อน
 - 6.3.4.2. การทดสอบประสิทธิภาพของการเรียกใช้งาน API (API Performance Testing): ใช้เครื่องมือ เพื่อทดสอบประสิทธิภาพในการเรียกใช้งาน API โดยวัดเวลาที่ใช้ในการรับส่งข้อมูลระหว่าง แพลตฟอร์ม Low-Code และแหล่งข้อมูลภายนอก ตรวจสอบความเร็วในการตอบสนองและ ประสิทธิภาพของการทำงานของ API
 - 6.3.4.3. การทดสอบประสิทธิภาพของระบบความปลอดภัย (Security Performance Testing): ทดสอบประสิทธิภาพของระบบความปลอดภัยของแพลตฟอร์ม Low-Code โดยการทดสอบการ รักษาความปลอดภัยของข้อมูล การตรวจสอบการเข้าถึงระบบ การจัดการสิทธิ์การเข้าถึง เพื่อให้ มั่นใจในความปลอดภัยของระบบ
 - 6.3.4.4. การทดสอบประสิทธิภาพของการประมวลผล (Processing Performance Testing): การ ทดสอบประสิทธิภาพของการประมวลผลเป็นกระบวนการที่ใช้ เพื่อวัดและตรวจสอบ ประสิทธิภาพของระบบในการประมวลผลข้อมูลในแพลตฟอร์ม Low-Code ว่าสามารถทำงานได้ อย่างมีประสิทธิภาพหรือไม่ โดยการทดสอบนี้มุ่งเน้นไปที่การประมวลผลข้อมูลที่ซับซ้อนหรือมี ปริมาณมากเพื่อดูว่าระบบสามารถจัดการได้ในระดับที่ต้องการ และมีเวลาการประมวลผลที่ ยอมรับได้ในสถานการณ์ปกติหรือสถานการณ์ที่มีการใช้งานอย่างสูงสุด

7. แผนการดำเนินโครงงาน (Gantt Chart)

					_												
	Task Mode	WBS	Task Name	Baseline Actual Work Work	% Compl	Baseline Start le	Baseline Finish	àine	July August	September	October	November	December	January	February	March	April
_ (8	1	Begin	O hee O hee	00/	Mon 7/10/23	Mon 7/10/22	J	J A	s	0	N	D	1	F	М	
2	*	2	Sprint 1			Mon 7/10/23		-	H-1								
2 3 4 5	*	2.1	Front_end			Mon 7/10/23			н								
	*	2.1.1	สร้างโปรเจ็ค React	0 hrs 0 hrs	0%	Mon 7/10/23	Tue 7/18/23		0%								
,	*	2.1.2	สร้าง Component				Tue 7/18/23										
3	*	2.2	Back-end			Wed 7/19/23		_	- 0%								
´	*	2.2.1	สร้างโปรเจ็ค Node.js สำหรับ Web Server	U nrs U nrs	0%	Wed 7/19/23	Fri 7/14/23										
8	*	2.2.2	ดิดตั้ง Library ที่จำเป็น	0 hrs 0 hrs	0%	Tue 7/25/23	Wed 7/26/23	-	₩ 0%								
9	*	2.2.3	ตั้งค่า Node.js Environment สำหรับการ			Thu 7/27/23	Mon 8/28/23	-	0%	_							
			พัฒนา														
10	*	2.2.4	สร้างฐานข้อมูล MySQL สำหรับการพัฒนา	0 hrs 0 hrs	0%	Thu 7/27/23	Fri 7/28/23		₩ 0%								
11		3	Sprint 2	O bre O bre	0%	Tuo 9/1/22	Thu 8/31/23	-	_	_							
12	7	3.1	Front-end			Tue 8/1/23 Tue 8/1/23	Tue 8/15/23	-	_ :_	•							
13	*	3.1.1	สร้าง Component (ต่อ)			Tue 8/1/23	Sat 8/5/23	-	1 0%								
14	*	3.1.2	สร้างหน้า Login - Register			Sun 8/6/23	Thu 8/10/23		■ 09								
15	*	3.1.3	สร้างหน้า Create Project	0 hrs 0 hrs	0%	Fri 8/11/23	Tue 8/15/23		-	%							
16	*	3.2	Back-end				Thu 8/31/23		_ r	_							
17	*	3.2.1	สร้าง API ไว้ใช้ติดต่อกับฐานข้อมูล			Wed 8/16/23	Sun 8/20/23	-		0%							
18	× .	3.2.2 4	สร้างฐานข้อมูล (ต่อ) Sprint 3	O hrs O hrs		Mon 8/21/23	Thu 8/31/23 Sat 9/30/23	-			_						
20	-	4.1	Front-End	0 hrs 0 hrs			Fri 9/15/23	-		<u> </u>	•						
21	*	4.1.1	สร้างหน้าสำหรับออกแบบ Front-end	0 hrs 0 hrs			Tue 9/5/23	-		0%							
22	*	4.1.2	สร้างหน้า Create Project (ต่อ)			Wed 9/6/23	Sun 9/10/23			■ 0%							
23	*	4.1.3	ทดสอบ Library Drag and Drop	0 hrs 0 hrs	0%	Mon 9/11/23	Fri 9/15/23			■ 01	N ₄						
24	*	4.2	Back-end	0 hrs 0 hrs	0%	Sat 9/16/23	Sat 9/30/23			-	_						
25	*	4.2.1	สร้างการเชื่อมต่อ API หน้า Login - Register	0 hrs 0 hrs	0%	Sat 9/16/23	Fri 9/22/23			-	0%						
26		422	พอสวนเพี้น C-d- คอดซึ่งโรย	0 hr- 0 '	On'	Cat 0 /22 /22	Cat 0/20/22	-			na.						
40	×	4.2.2	ทดสอบเพิ่ม Code จากใช้ Library Drag and Drop	o nrs 0 hrs	υ%	sat 9/23/23	Sat 9/30/23				- 0%						
27	*	5	Sprint 4	0 hrs 0 hrs	0%	Sun 10/1/23	Fri 10/20/23	-			_						
28	*	5.1	Front-end			Sun 10/1/23	Tue 10/10/23	1			Ή.						
29	*	5.1.1	สร้างแถบเครื่องมือ Element			Sun 10/1/23	Thu 10/5/23	1			0%						
30	*	5.1.2	สร้างแถบเครื่องมือ Section			Fri 10/6/23	Tue 10/10/23				■ 0%						
31	*	5.2	Back-end	0 hrs 0 hrs	0%	Wed 10/11/23	Fri 10/20/23				Н						
32	*	5.2.1	ทดสอบเพิ่ม Code จากการใช้เครื่องมือ	0 hrs 0 hrs	0%	Wed 10/11/23	Sun 10/15/23				■ 0	%					
22			Element ทดสอบเพิ่ม Code จากการใช้เครื่องมือ	0.1	or:		F-1 40 /22 /22	-				0%					
33	×	5.2.2	ทดสอบเพิ่ม Code จากการใช้เครื่องมือ Section	0 hrs 0 hrs	0%	Mon 10/16/23	Fri 10/20/23				_	0%					
34	-	6	Sprint 5	O hrs O hrs	0%	Sat 10/21/23	Tue 10/31/23	-				_					
35	-	6.1	Front-end			Sat 10/21/23		-			i	—i					
36	*	6.1.1	สร้างแถบเครื่องมือสำหรับแก้ไข CSS			Sat 10/21/23		-			i	0%					
37	*	6.1.2	สร้างแถบเครื่องมือ Layer			Sat 10/21/23						0%					
38	*	6.2	Back-end	0 hrs 0 hrs	0%	Sat 10/21/23	Tue 10/31/23				- 1	\neg					
39	*	6.2.1	ทดสอบเพิ่ม แก้ไข CSS จากแถบเครื่องมือ	0 hrs 0 hrs	0%	Sat 10/21/23	Tue 10/31/23					0%					
	-											-					
40	*	6.2.2	ทดสอบการใช้งานแถบเครื่องมือ Layer			Sat 10/21/23		-				0%					
41	7	7	Sprint 6	0 hrs 0 hrs	0%	Wed 11/1/23	Wed 11/15/23	1									
42	*	7.1	Front-end	0 hrs 0 hrs	0%	Wed 11/1/23	Wed 11/15/23	1				\Box					
						, _, _,	,,,										
43	*	7.1.1	สร้างแถบเครื่องมือ Properties	0 hrs 0 hrs	0%	Wed 11/1/23	Wed 11/15/23	į.					0%				
													ne.				
44	*	7.1.2	สร้างแถบเครื่องมือ API	0 hrs 0 hrs	0%	Wed 11/1/23	Wed 11/15/23	·				-	J7%				
45		7.2	Back-end	O bre O bre	0%	Wed 11/1/22	Wed 11/15/23	-				$\overline{}$					
.		/.2	Dack-end	01113 01113	0,0	Wed 11/1/25	Wed 11/15/25	1									
46	*	7.2.1	ทดสอบการใช้งานแถบ properties	0 hrs 0 hrs	0%	Wed 11/1/23	Wed 11/15/23	ı .					0%				
47	*	8	Sprint 7			Thu 11/16/23		_									
49	*	8.1	Front-end			Thu 11/16/23		-									
49	*	8.1.1	สร้างฟังก์ชันหน้าถัดไปสำหรับ Project Front-end	U nrs U nrs	0%	Thu 11/16/23	Inu 11/30/23					_	- 0.0				
50	*	8.1.2	สร้างฟังก์ชั่น Generate Code	0 hrs 0 hrs	0%	Thu 11/16/23	Thu 11/30/23	1					0%				
51	*	8.1.3	สร้างฟังก์ชันแชร์ Project			Thu 11/16/23							0%				
52	*	8.2	Back-end	0 hrs 0 hrs	0%	Thu 11/16/23	Thu 11/30/23					ь	\neg				
53	*	8.2.1	ทดสอบพึงก์ชั่นต่าง ๆ			Thu 11/16/23							0%				
54	*	9	Sprint 8			Fri 12/1/23	Fri 12/15/23	-									
55	*	9.1	Front-end สร้างฟังก์ชันเชื่อมต่อ API กับ Front-end			Fri 12/1/23	Fri 12/15/23	-						16.			
56 57	*	9.1.1	สร้างพึงก์ชนเชื่อมต่อ API กับ Front-end ทดสอบการแสดงของ API ที่เขื่อมกับหน้า			Fri 12/1/23 Fri 12/1/23	Fri 12/15/23 Fri 12/15/23	-						%			
	×	9.1.2	ทดลอบการแสดงของ API ที่เขอมกับหนา บ้าน	o iiis u nrs	0%	rn 12/1/23	FII 12/15/23										
58	*	9.2	Back-end	0 hrs 0 hrs	0%	Fri 12/1/23	Fri 12/15/23	1									
59	*	9.2.1	ทดสอบการ GET ข้อมูลจาก API			Fri 12/1/23	Fri 12/15/23						0	%			
60	*	9.2.2	ทดสอบการ POST ข้อมูลจาก API			Fri 12/1/23	Fri 12/15/23						°	%			
61	*	10	Sprint 9				Sun 12/31/23							\neg			
62	*	10.1	Front-end				Sun 12/31/23	-					-	-			
63	*	10.1.1	สร้างฟังก์ชัน Preview Project			Sat 12/16/23	Sun 12/31/23	-						0%			
64	×	10.1.2	สร้างฟังก์ชันเชื่อมต่อ API กับ Front-end (ต่อ	u hrs 0 hrs	0%	Sat 12/16/23	Sun 12/31/23						_	- V5			
65	*	10.1.3) ทดสอบการแสดงของ API ที่เขื่อมกับหน้า	0 hrs 0 hrs	0%	Sat 12/16/23	Sun 12/31/23	-					_	0%			
	,	20.1.3	บ้าน (ต่อ)	0 1113	270								_	_			
66	*	10.2	Back-end			Sat 12/16/23	Sun 12/31/23						-	\neg			
67	*	10.2.1	ทดสอบการ GET ข้อมูลจาก API (ต่อ)				Sun 12/31/23						-	0%			
68	*	10.2.2	ทดสอบการ POST ข้อมูลจาก API (ต่อ)			Sat 12/16/23		-					-	0%			
69	*	11	Sprint 10			Mon 1/1/24	Sun 1/7/24	-						Н			
70	*	11.1	Front-end ทำ Unit Test ส่วน Front-end			Mon 1/1/24 Mon 1/1/24	Sun 1/7/24	-						H 0%			
72	*	11.1.1 11.2	ทา Unit Test สวน Front-end Back-end			Mon 1/1/24 Mon 1/1/24	Sun 1/7/24 Sun 1/7/24	-									
73	*	11.2.1	ทำ Unit Test ส่วน Back-end			Mon 1/1/24	Sun 1/7/24 Sun 1/7/24	-						0%			
74	*	12	Sprint 11			Mon 1/8/24	Sun 1/1/24	-						_			
75	*	12.1	Front-end			Mon 1/8/24	Sun 1/14/24	1						н			
76	*	12.1.1	ทำ Unit Test ส่วน Front-end (ต่อ)			Mon 1/8/24	Sun 1/14/24	1						0	%		
			, , , , , , , , , , , , , , , , , , , ,														
			Critical Split			Finish only		Baseline Milestone 🔇	>	Manual Summar	,		Inactive Task				
			Critical Split Task Progress			Durstion-only		dicetore (•	Project Summar	-		Inactive Milesto				
			Critical Progress Manual Task			Baseine		Summary Progress =		External Tasks			Inactive Summa				
			Tank Start only	С		Baseline Spit		turnery I		External Milesto	~ ◊		Deading	4			



8. ประโยชน์ที่คาดว่าจะได้รับ (Expected Benefits)

8.1 ความสะดวกในการสร้างเว็บไซต์ ผู้ใช้ไม่จำเป็นต้องมีความเชี่ยวชาญในการเขียนโปรแกรมเพื่อสร้างเว็บไซต์ ขึ้นมาใหม่ ด้วยเครื่องมือสร้างเว็บไซต์เชิงสร้างสรรค์ เช่น CMS (Content Management System) หรือ เครื่องมือสร้างเว็บไซต์ที่มีอยู่ในระบบ ผู้ใช้สามารถสร้างเว็บไซต์ได้อย่างง่ายและสะดวกด้วยการลาก-วาง องค์ประกอบต่าง ๆ และปรับแต่งรูปแบบต่าง ๆ ตามต้องการของตนเอง ทำให้กระบวนการสร้างเว็บไซต์เป็น เรื่องที่ไม่ซับซ้อนและสนุกสนานมากขึ้นประหยัดเวลาในการพัฒนาโปรแกรมเว็ปส่วนหน้า (Front-End)

- 8.2 ประหยัดเวลาและทรัพยากร การใช้เครื่องมือสร้างเว็บไซต์เชิงสร้างสรรค์ช่วยลดเวลาในกระบวนการสร้าง เว็บไซต์เมื่อเทียบกับการเขียนเองตั้งแต่เริ่มต้น นอกจากนี้ยังไม่ต้องใช้ทรัพยากรมากในการจ้างนักพัฒนา เว็บไซต์หรือทีมพัฒนาให้เสร็จสมบูรณ์ ทำให้ผู้ใช้สามารถปรับแต่งและพัฒนาเว็บไซต์นั้นเป็นไปได้อย่างอิสระ ตามความต้องการและความคิดสร้างสรรค์ของตนเอง ผู้ใช้สามารถทำการแก้ไขหรือปรับปรุงเว็บไซต์ได้ ตลอดเวลาโดยไม่ต้องพึ่งพาผู้อื่น
- 8.3 ความยืดหยุ่นในการปรับแต่ง เครื่องมือสร้างเว็บไซต์เชิงสร้างสรรค์มักมีตัวเลือกและฟีเจอร์ที่หลากหลาย ทำ ให้ผู้ใช้สามารถปรับแต่งเว็บไซต์ให้เหมาะสมกับความต้องการและลักษณะของธุรกิจหรือผลงานที่ต้องการ สร้างขึ้น สามารถเลือกออกแบบที่ตรงกับความต้องการของตนเองได้ ซึ่งเป็นประโยชน์สำคัญที่ช่วยให้ผู้ใช้ สามารถสร้างเว็บไซต์ที่มีความเป็นเอกลักษณ์และแตกต่างจากเว็บไซต์อื่น ๆ
- 8.4 ลดต้นทุนและเวลาในการสร้างเว็บไซต์ การใช้เครื่องมือสร้างเว็บไซต์ช่วยลดความซับซ้อนและเวลาในการ สร้างเว็บไซต์ โดยไม่จำเป็นต้องมีความเชี่ยวชาญในการเขียนโค้ด การสร้างเว็บไซต์ด้วยเครื่องมือเชิง สร้างสรรค์อาจช่วยประหยัดค่าใช้จ่ายในการจ้างนักพัฒนาเว็บไซต์ภายนอกและเพิ่มประสิทธิภาพใน กระบวนการพัฒนา
- 8.5 การทดสอบและปรับปรุง เครื่องมือสร้างเว็บไซต์มักมีฟีเจอร์ในการทดสอบและปรับปรุงเว็บไซต์ ผู้ใช้สามารถ ทดสอบความเร็วการโหลดหน้าเว็บ ประสิทธิภาพของ เพื่อปรับปรุงและปรับแต่งเว็บไซต์ให้สามารถทำงานได้ ดีที่สุด

9. ผลการศึกษาเทคโนโลยีที่ใช้พัฒนา

9.1 HTML

HTML (HyperText Markup Language) คือภาษาคอมพิวเตอร์ที่ใช้สำหรับสร้างและแสดงเว็บไซต์ บนอินเทอร์เน็ต โดย HTML ใช้รหัสแท็ก (tag) เพื่อบอกว่าเนื้อหาบนหน้าเว็บไซต์ควรแสดงอย่างไร และจะ เรียกใช้รูปแบบต่าง ๆ เช่น ข้อความ รูปภาพ ลิงก์ ตาราง และอื่นๆ โดย HTML เป็นส่วนหนึ่งของเทคโนโลยี เว็บและเป็นภาษาพื้นฐานที่จำเป็นสำหรับการพัฒนาเว็บไซต์ ในปัจจุบัน HTML ได้พัฒนาออกมาเป็น เวอร์ชันล่าสุดคือ HTML5 ซึ่งมีฟีเจอร์และความสามารถที่มากกว่าเวอร์ชันก่อนหน้านี้

9.2 CSS

CSS (Cascading Style Sheets) คือภาษาคอมพิวเตอร์ที่ใช้สำหรับกำหนดรูปแบบและลักษณะต่าง ๆ ของหน้าเว็บไซต์ เช่น สีพื้นหลัง ขนาดและรูปแบบตัวอักษร รูปแบบเส้นขอบ รูปภาพ และอื่น ๆ โดย CSS

ใช้กฎการเขียน (rules) เพื่อเลือกส่วนหนึ่งของเอกสาร HTML และกำหนดรูปแบบการแสดงผลของส่วนนั้น ๆ ซึ่งช่วยให้การออกแบบเว็บไซต์เป็นไปได้อย่างสวยงามและสามารถเข้าถึงได้ง่ายขึ้น นอกจากนี้ CSS ยังช่วย ลดความซ้ำซ้อนในการเขียน HTML และเพิ่มความยืดหยุ่นในการปรับแต่งหน้าเว็บไซต์ให้ตรงตามความ ต้องการของผู้ใช้งาน

9.3 JavaScript

JavaScript คือภาษาโปรแกรมมิ่งสำหรับเว็บไซต์ที่ใช้งานได้บนเว็บเบราว์เซอร์ โดย JavaScript ช่วย ให้ผู้พัฒนาเว็บไซต์สามารถสร้างและควบคุมพฤติกรรมต่าง ๆ ของเว็บไซต์ได้ เช่น การตรวจสอบข้อมูลที่ผู้ใช้ ป้อนเข้ามาในแบบฟอร์ม การสร้างเอฟเฟกต์และการเปลี่ยนแปลงสไตล์หน้าเว็บไซต์ เป็นต้น นอกจากนี้ JavaScript ยังช่วยเพิ่มประสิทธิภาพที่ดีขึ้น ซึ่ง ทำให้ JavaScript เป็นภาษาโปรแกรมมิ่งที่สำคัญและอยู่ในแนวเทคโนโลยีเว็บที่ได้รับความนิยมมากที่สุดใน ปัจจุบัน

9.4 ReactJS

React.js คือไลบรารี (library) สำหรับพัฒนาเว็บเว็บไซต์ ซึ่งเป็นส่วนหนึ่งของเทคโนโลยี JavaScript ที่ใช้งานได้บนเว็บเบราว์เซอร์ โดยเฉพาะในการพัฒนาเว็บเว็บไซต์ที่มีการอัพเดตข้อมูลแบบ real-time หรือ การสร้าง User Interface (UI) ที่เปลี่ยนแปลงได้รวดเร็ว React.js มีความยืดหยุ่นสูงและมีความสามารถใน การสร้าง Component ที่สามารถนำไปใช้ได้ซ้ำได้เป็นส่วนต่อประสานกับการพัฒนา Backend อย่าง Node.js ได้ง่าย นอกจากนี้ React.js ยังมีชุดเครื่องมือ (toolset) สำหรับการทำ Unit testing และ Endto-End testing ที่ช่วยให้การทดสอบและการปรับปรุงคุณภาพของเว็บเว็บไซต์ด้วย React.js ง่ายขึ้น

9.5 ExpressJS

ExpressJS มีลักษณะเป็นเฟรมเวิร์กที่เบาและมีแกนกลางที่เรียกว่า "Express" เป็นตัวควบคุมหลัก ในการกำหนดเส้นทาง (routing) และการจัดการกับคำขอ (request) และการตอบสนอง (response) ของ แอปพลิเคชันเว็บ สิ่งนี้ช่วยให้นักพัฒนาสามารถสร้างเว็บแอปพลิเคชันที่เป็นไปตามความต้องการของพวกเขาได้ อย่างสะดวกและเร็วด้วย JavaScript

9.6 Amazon Web Services (AWS)

Amazon Web Services (AWS) เป็นแพลตฟอร์มคลาวด์ (cloud platform) ที่ให้บริการใน รูปแบบของ Infrastructure as a Service (IaaS) และ Platform as a Service (PaaS) โดย Amazon Web Services เป็นบริการคลาวด์อันดับหนึ่งของโลก มีบริการหลากหลายรูปแบบเช่น การจัดการคลาวด์ (cloud management), บริการคลาวด์คอมพิวติ้ง (cloud computing), บริการเกี่ยวกับฐานข้อมูลและการ จัดเก็บข้อมูล (database and storage services), บริการสำหรับการวิเคราะห์ข้อมูล (analytics), บริการ เกี่ยวกับเครือข่ายและความปลอดภัย (networking and security services) และอื่น ๆ อีกมากมาย โดย Amazon Web Services เป็นบริการที่ใช้งานได้ง่ายและเหมาะสำหรับการใช้งานทั้งในธุรกิจขนาดเล็กและ ขนาดใหญ่ และได้รับการนิยมเป็นอย่างมากในวงการเทคโนโลยี

10. เอกสารอ้างอิง (Reference)

[1] "API คืออะไร", [online]. Available https://aws.amazon.com/th/what-is/api/. 2022.

- [2] "CSS คืออะไร", [online]. Available
 https://www.mindphp.com/คู่มือ/73-คือะไร/2193-css-คืออะไร.html. 2022.
- [3] "HTML คืออะไร", [online]. Available

 https://www.mindphp.com/คู่มือ/73-คือะไร/2026-html-คืออะไร.html. 2022.
- [4] "JavaScript คืออะไร", [online]. Available

 https://www.mindphp.com/คู่มือ/73-คือะไร/2187-java-javascript-คืออะไร.html. 2022.
- [5] "Low code platform คือ", [online]. Available
 https://www.fusionsol.com/blog/low-code-platform-คือ/. 2022.
- [6] "Power Apps คืออะไร", [online]. Available
 https://learn.microsoft.com/th-th/power-apps/powerapps-overview. 2022.
- [7] "POWER APPS MODEL DRIVEN", [online]. Available

 https://www.fusionsol.com/blog/power-apps-model-driven/. 2022.
- [8] "React 101", [online]. Available

 https://www.borntodev.com/2020/07/15/react-101/. 2022.
- [9] **"Visual Programming คืออะไรและทำงานอย่างไร"**, [online]. Available https://appmaster.io/th/blog/visual-programming. 2022.
- [10] "การใช้งาน Express.js บน node.js", [online]. Available http://marcuscode.com/tutorials/nodejs/using-expressjs. 2021.
- [11] **"การประมวลผลบนคลาวด์ด้วย AWS"**, [online]. Available https://aws.amazon.com/th/what-is-aws/. 2022.
- [12] **"การพัฒนาแอปพลิเคชันอย่างรวดเร็วคืออะไร"**, [online]. Available https://appmaster.io/th/blog/rad. 2022.