



รายงานข้อเสนอโครงการวิศวกรรมคอมพิวเตอร์
วิชา 01076014 การเตรียมโครงการวิศวกรรมคอมพิวเตอร์
ภาคเรียนที่ 2 ปีการศึกษา 2564

1. ชื่อหัวข้อโครงการ (ไทย)โปรแกรมสร้างเว็บไซต์.....
2. ชื่อหัวข้อโครงการ (อังกฤษ)Web Frontend Builder.....
3. Keyword 3 คำLow-Code, Front-End Builder, API Tester.....
4. ประเภทโครงการ (X)
☐ 1. HW+SW ☒ 2. SW_Dev ☐ 3. Research
5. รายชื่อผู้ทำโครงการ
5.1. นายวันนุชาน แวอู้ง.....รหัส64015139.....
5.2. นายอรุณพ ธรรมเที่ยงธรรม.....รหัส64015166.....
5.3. นายเอกรินทร์ องอาจ.....รหัส64015172.....
6. อาจารย์ที่ปรึกษา
6.1. อาจารย์ที่ปรึกษาหลักผศ. ธนา หงษ์สุวรรณ.....
6.2. อาจารย์ที่ปรึกษาร่วมผศ.ดร. ชมพูนุท เต็งเจริญ.....

1. ที่มาและความสำคัญของปัญหา (Motivation)

การสร้างเว็บไซต์หรือเว็บไซต์ในปัจจุบันมีความสำคัญอย่างมากในการพัฒนาธุรกิจและการสร้างสรรค์ผลงานใหม่ ๆ อย่างไรก็ตาม ผู้ใช้ที่ไม่มีความรู้และทักษะด้านการเขียนโปรแกรม (coding) และพบว่าเข้าถึงและสร้างเว็บไซต์เหล่านั้นเป็นเรื่องที่ยากลำบากและซับซ้อน สิ่งนี้อาจทำให้หลายๆคนรู้สึกที่ไม่สามารถเข้าถึงเทคโนโลยีได้ นอกจากนี้ยังมีผู้ใช้ที่สนใจในการสร้างเว็บไซต์หรือเว็บไซต์เพื่อใช้ในธุรกิจของตนเอง แต่ขาดความรู้และทักษะในการเขียนโค้ด นั้นทำให้ประสบปัญหาในการดำเนินการและการพัฒนาซอฟต์แวร์ที่เหมาะสม

ด้วยเหตุนี้เองที่แพลตฟอร์ม Low-Code กลายเป็นทางเลือกที่ดีและแก้ไขปัญหาดังกล่าวได้อย่างสมบูรณ์ แพลตฟอร์มดังกล่าวอนุญาตให้ผู้ใช้ที่ไม่มีความรู้และทักษะด้านการเขียนโปรแกรมสร้างเว็บไซต์ได้อย่างง่ายดายโดยมีการใช้เครื่องมือและอินเทอร์เฟซกราฟิกเพื่อสร้างและแก้ไขฟังก์ชันต่าง ๆ ของเว็บไซต์ ผู้ใช้สามารถสร้างเว็บไซต์ได้โดยไม่ต้องเขียนโค้ดให้เอง ทำให้แพลตฟอร์มดังกล่าวเป็นเครื่องมือที่สามารถนำไปใช้ในธุรกิจส่วนตัวหรือการสร้างสรรค์ได้หลากหลายและยืดหยุ่นมากขึ้น

2. วัตถุประสงค์ (Objectives)

1. เพื่อลดความซับซ้อนในกระบวนการสร้างเว็บไซต์ ผู้ใช้สามารถใช้เครื่องมือและอินเทอร์เฟซกราฟิกที่ใช้ทำงานง่ายในการสร้างและแก้ไข ทำให้ไม่จำเป็นต้องมีความรู้เชิงลึกในการเขียนโค้ด
2. เพื่อประหยัดเวลาและทรัพยากรในการพัฒนา การใช้แพลตฟอร์ม Low-Code ช่วยลดเวลาในกระบวนการสร้างและปรับปรุงเว็บไซต์ โดยทำให้ผู้ใช้สามารถกำหนดและปรับเปลี่ยนสิ่งต่าง ๆ ได้อย่างรวดเร็ว
3. เพื่อทดสอบการใช้งาน API ของตนเองในแพลตฟอร์ม Low-Code ผู้ใช้สามารถเชื่อมต่อและใช้งาน API จากแหล่งที่มาต่าง ๆ เพื่อเพิ่มฟังก์ชันและความสามารถของเว็บไซต์

3. ทฤษฎีที่เกี่ยวข้อง (Theoretical Background)

3.1 แนวคิดของ Low-Code Platform

3.1.1. Low-Code Development

เป็นแนวคิดที่เน้นการใช้เครื่องมือและอินเทอร์เฟซกราฟิกในการสร้างเว็บไซต์ โดยลดระยะเวลาและความซับซ้อนในกระบวนการพัฒนา ผู้ใช้สามารถสร้างเว็บไซต์ได้โดยใช้การลากและวางองค์ประกอบที่พร้อมใช้งาน และกำหนดค่าต่าง ๆ ที่ต้องการ โดยไม่ต้องเขียนโค้ดเป็นจำนวนมาก

3.1.2. Visual Programming

เป็นแนวคิดที่ใช้กราฟิกหรืออินเทอร์เฟซสำหรับการเขียนโปรแกรมแทนการเขียนโค้ดด้วยภาษาโปรแกรมมิ่งแบบแถวบรรทัดเดียว ผู้ใช้สามารถสร้างโปรแกรมโดยการเชื่อมต่อองค์ประกอบและตั้งค่าผ่านกราฟิกหรืออินเทอร์เฟซที่ใช้งานง่าย ทำให้ผู้ใช้ที่ไม่มีความรู้ด้านการเขียนโค้ดสามารถสร้างเว็บไซต์ได้อย่างรวดเร็ว

3.1.3. Model-Driven Development

เป็นวิธีการพัฒนาซอฟต์แวร์ที่เน้นการสร้างและใช้งานโมเดล (Models) เป็นส่วนสำคัญในกระบวนการพัฒนา แพลตฟอร์ม Low-Code ใช้แนวคิด MDD ในการสร้างเว็บไซต์ โดยผู้ใช้สร้างและกำหนดโมเดลของเว็บไซต์ในรูปแบบที่เข้าใจง่าย โดยรวมถึงการกำหนดคุณสมบัติ และความสัมพันธ์ระหว่างองค์ประกอบต่างๆ ในโมเดล ซึ่งแพลตฟอร์มจะทำการสร้างโค้ดอัตโนมัติจากโมเดลที่กำหนด เพื่อให้ผู้ใช้ไม่ต้องเขียนโค้ดเอง

3.1.4. Rapid Application Development

เป็นวิธีการพัฒนาซอฟต์แวร์ที่ให้ความสำคัญกับการสร้างเว็บไซต์ในเวลาที่น่ารวดเร็ว แพลตฟอร์ม Low-Code เน้นการพัฒนาเว็บไซต์อย่างรวดเร็วด้วยการใช้งานเครื่องมือและอินเทอร์เฟซที่ง่ายต่อการสร้าง รวมถึงการทำงานร่วมกับทีมผู้พัฒนาอื่นๆ เพื่อลดระยะเวลาในการพัฒนาและปรับปรุงเว็บไซต์

3.1.5. Integration and Connectivity

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

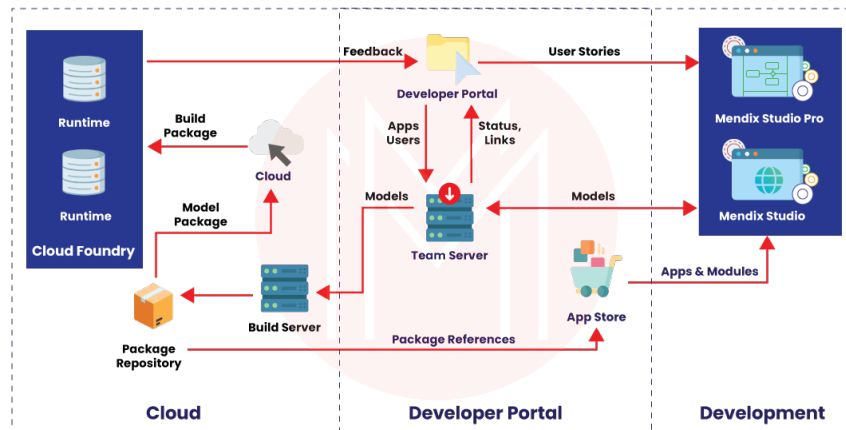
3.1.6. Application Programming Interface

API เป็นช่องทางที่ใช้ในการเชื่อมต่อระบบหรือบริการต่างๆ เพื่อแลกเปลี่ยนข้อมูลหรือฟังก์ชันระหว่างเว็บไซต์และแหล่งข้อมูลหรือบริการอื่นๆ ผู้ใช้สามารถนำ API มาใช้ในแพลตฟอร์ม Low-Code เพื่อเชื่อมต่อกับฐานข้อมูลภายนอก บริการคลังข้อมูลออนไลน์ ระบบชำระเงิน หรือแหล่งข้อมูลอื่นๆ เพื่อเพิ่มประสิทธิภาพและความสามารถของเว็บไซต์

4. งานวิจัยที่เกี่ยวข้อง (Related Works)

4.1 Mendix

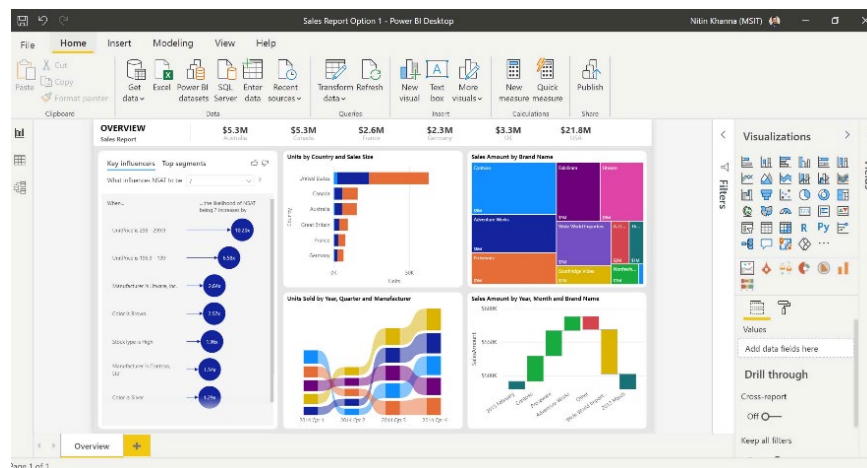
Mendix เป็นแพลตฟอร์ม Low-Code ที่ให้ผู้ใช้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่ายดาย มีการใช้งานแบบลากและวางที่เป็นเครื่องมือในการสร้างและแก้ไขฟังก์ชันต่าง ๆ ของแอปพลิเคชัน ผู้ใช้สามารถเลือกใช้ตัวควบคุมที่กำหนดไว้ล่วงหน้าและทำงานร่วมกับฐานข้อมูลในการสร้างแอปพลิเคชันที่มีประสิทธิภาพ Mendix ยังมีเครื่องมือสำหรับการจัดการโครงการและการทำงานร่วมกันของทีมผู้พัฒนาที่ช่วยให้งานเป็นไปอย่างราบรื่น



รูป 4.1 โครงสร้างการทำงานของ Mendix

4.2 Microsoft Power Apps

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



รูป 4.2 หน้าต่างการทำงานของ Microsoft Power Apps

5. ขอบเขตของโครงการ (Scope)

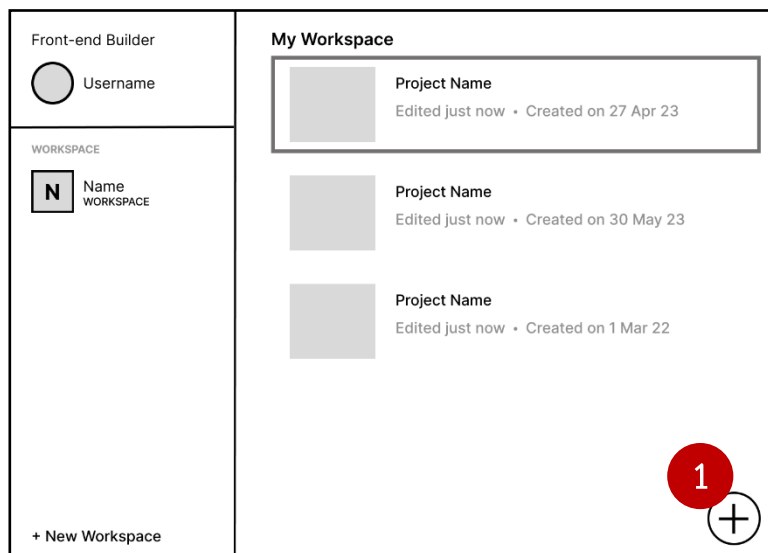
- 5.1 สามารถพัฒนาและสร้างแพลตฟอร์มที่ช่วยให้ผู้ใช้สามารถสร้างเว็บไซต์เว็บได้โดยไม่ต้องมีความรู้เทคนิคด้านการเขียนโปรแกรมเป็นพิเศษ
- 5.2 สามารถใช้งานฟีเจอร์พื้นฐานที่จำเป็นสำหรับการสร้างเว็บไซต์ เช่น การสร้างแบบฟอร์มในการป้อนข้อมูล การจัดการฐานข้อมูลพื้นฐาน การจัดการผู้ใช้งาน และการจัดการสิทธิ์การเข้าถึงข้อมูล
- 5.3 สามารถเชื่อมต่อกับ API ภายนอกเพื่อให้ผู้ใช้สามารถนำเอาข้อมูลหรือบริการจากแหล่งข้อมูลภายนอกมาใช้ในเว็บไซต์ที่พัฒนาได้
- 5.4 สามารถปรับแต่งและการปรับเปลี่ยนรูปแบบของเว็บไซต์ได้อย่างยืดหยุ่น ผู้ใช้สามารถปรับเปลี่ยนเค้าโครงหรือรูปแบบทั้งหมดของเว็บไซต์ได้ตามต้องการ ได้รวมถึงการปรับแต่งสี รูปแบบเลย์เอาต์ และส่วนประกอบอื่น ๆ
- 5.5 สามารถตรวจสอบความถูกต้องและการทดสอบเบื้องต้น ผู้ใช้สามารถตรวจสอบการทำงานของเว็บไซต์ และทดสอบความถูกต้องของฟังก์ชันพื้นฐานได้ก่อนการเผยแพร่
- 5.6 สามารถให้ผู้ใช้สร้างหน้าเว็บไซต์หลายหน้าได้ โดยให้ผู้ใช้สร้างหน้าเว็บไซต์ใหม่และกำหนดลำดับการเรียงลำดับหน้าเว็บไซต์
- 5.7 สามารถให้ผู้ใช้สร้างเมนูและการนำทางในเว็บไซต์ได้ โดยให้ผู้ใช้สร้างเมนูหลักและเมนูย่อย และกำหนดลำดับการเรียงลำดับของเมนู

6. การพัฒนาโครงการ (Project Development)

- 6.1 ขั้นตอนการพัฒนา (Methodology)
 1. กำหนดวัตถุประสงค์ ขอบเขต และเป้าหมายของโครงการ
 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

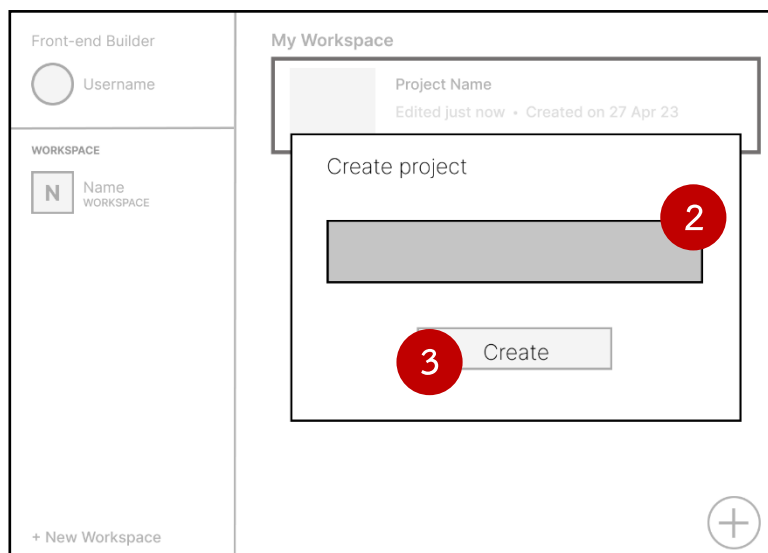
6.2 การออกแบบ (Design)

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



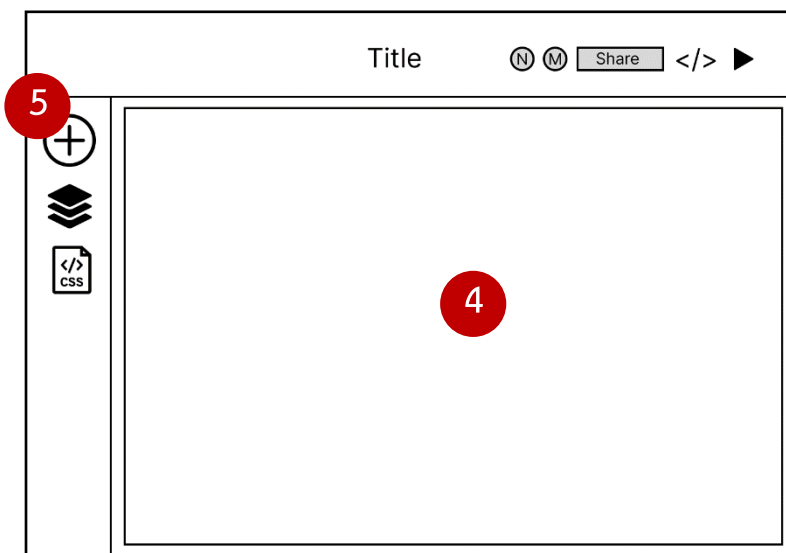
รูป 6.2.1.1 สร้าง Project

1. กดเมนูสร้าง Project เพื่อเริ่มสร้าง Project หน้าเปล่า



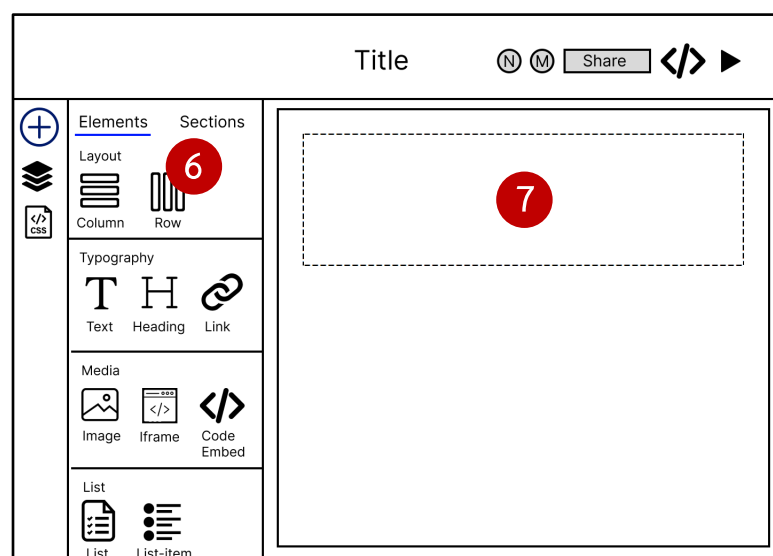
รูป 6.2.1.2 ใส่ชื่อ Project เพื่อเริ่มสร้าง Project

2. ใส่ชื่อ Project ที่ต้องการสร้าง
3. กด Create เพื่อสร้าง Project



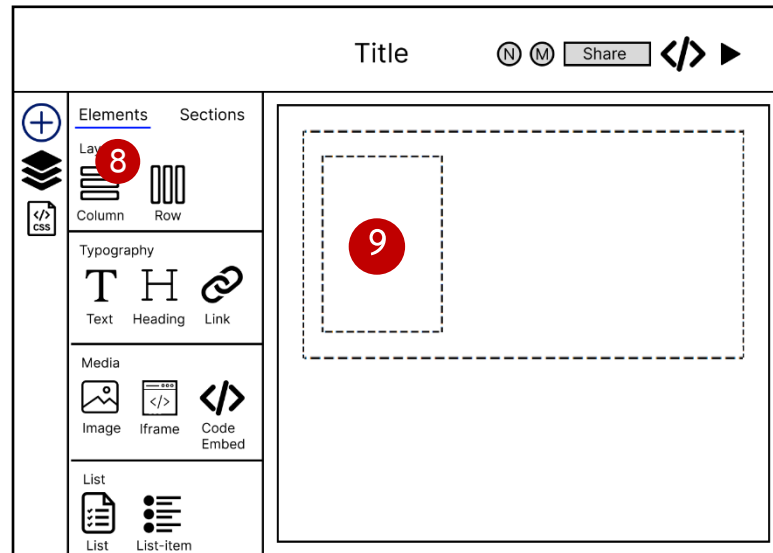
รูป 6.2.1.3 หน้าต่างการทำงาน

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



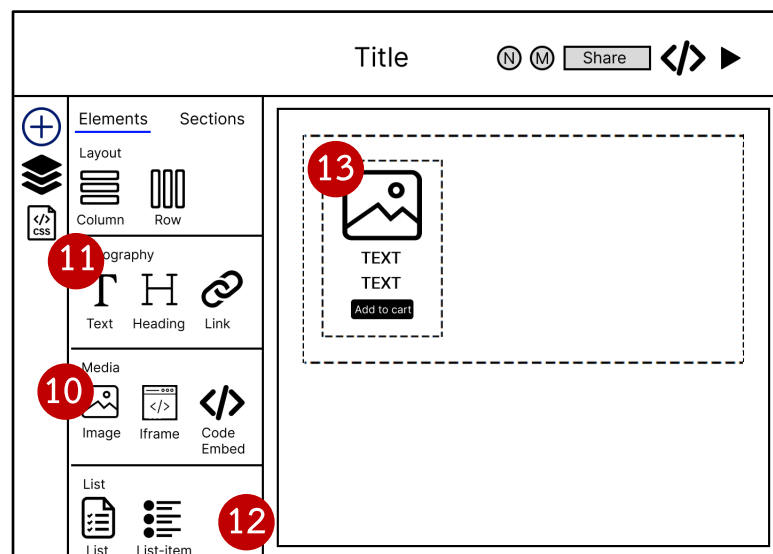
รูป 6.2.1.4 ใช้งาน Row เพื่อจัด Layout บนเว็บไซต์

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



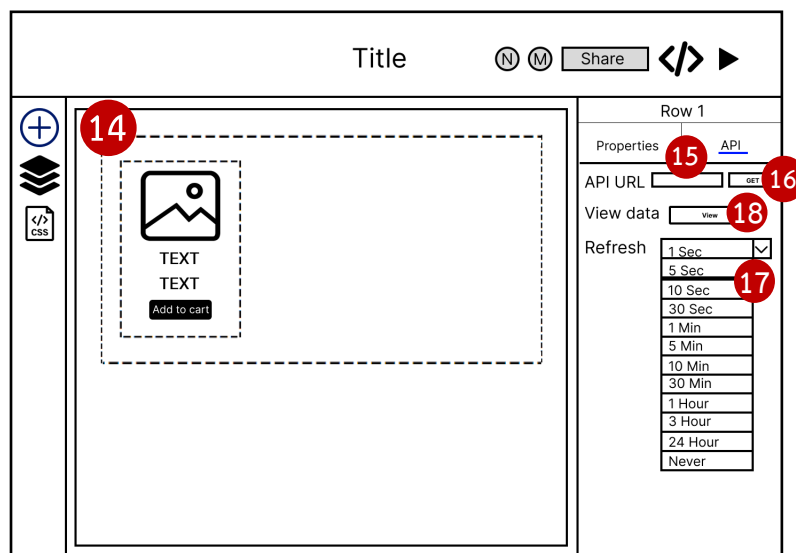
รูป 6.2.1.5 ใช้งาน Column เพื่อจัด Layout บนเว็บไซต์

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



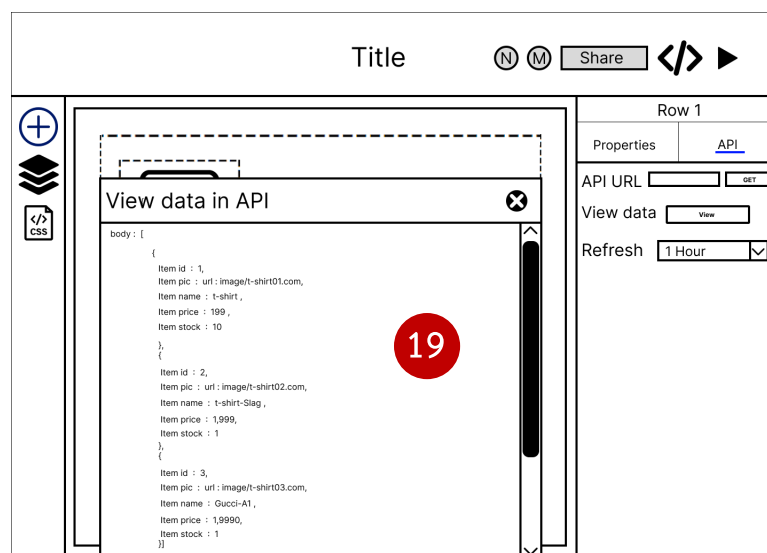
รูป 6.2.1.6 ใส่ Element ในเว็บไซต์ไว้แสดงข้อมูล

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
หมายเหตุเพิ่มเติม (12)Button อยู่ในเมนู Elements แต่ต้องเลื่อนหาด้านล่าง



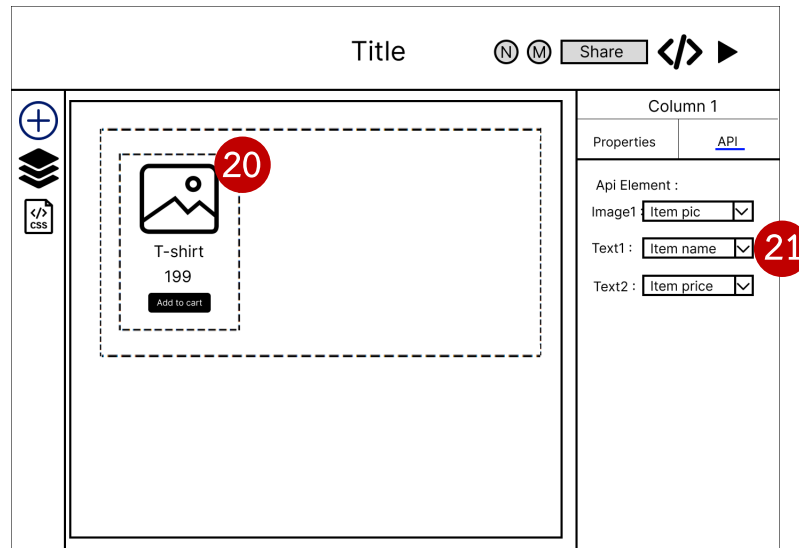
รูป 6.2.1.7 ใส่ข้อมูล API ให้ Row และดูข้อมูลที่ได้มาจากการ GET

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



รูป 6.2.1.8 แสดงข้อมูลที่ได้จากการ GET API

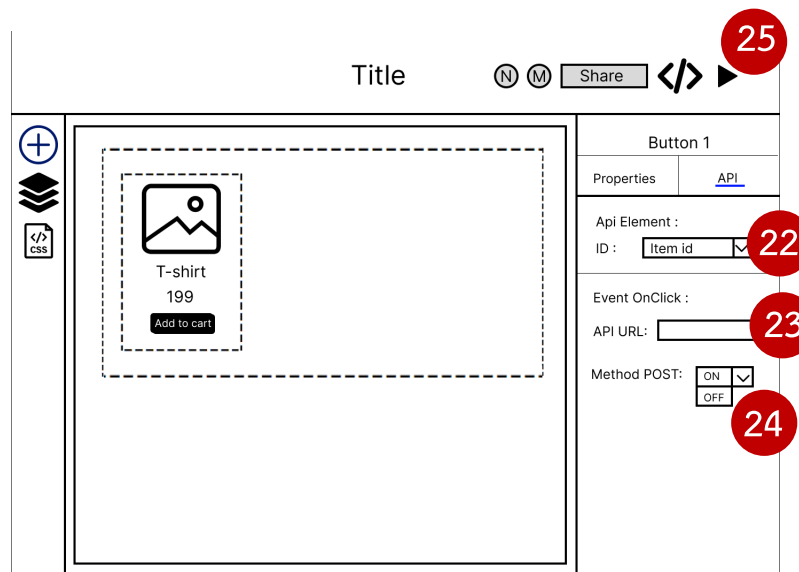
19. แสดงข้อมูลที่ได้มาจากการ GET



รูป 6.2.1.9 เลือกข้อมูลไปแสดงใน Elements

20. กดที่ Layout Column เพื่อเปิดแถบ API

21. เลือกข้อมูลจาก API มาแสดงผล (ข้อมูลจะแสดงผลตาม ข้อมูลที่ส่งมากับ API)



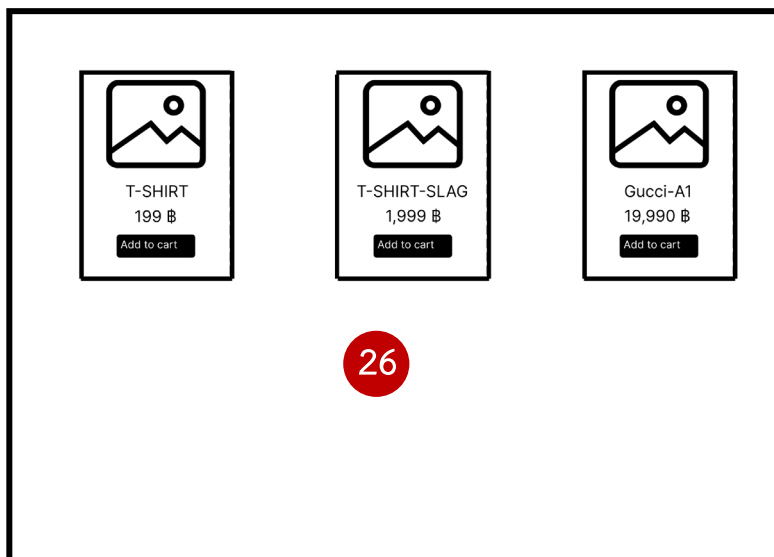
รูป 6.2.1.10 กดปุ่มเพื่อเพิ่มข้อมูลตะกร้าสินค้าในฐานข้อมูลด้วยการ POST API

22. เลือก id ของ item เพราะ id จะระบุความเป็นเจ้าของของแต่ละปุ่มนั้น

23. ใส่ลิงก์ API เพื่อไว้ POST ข้อมูลลง Database

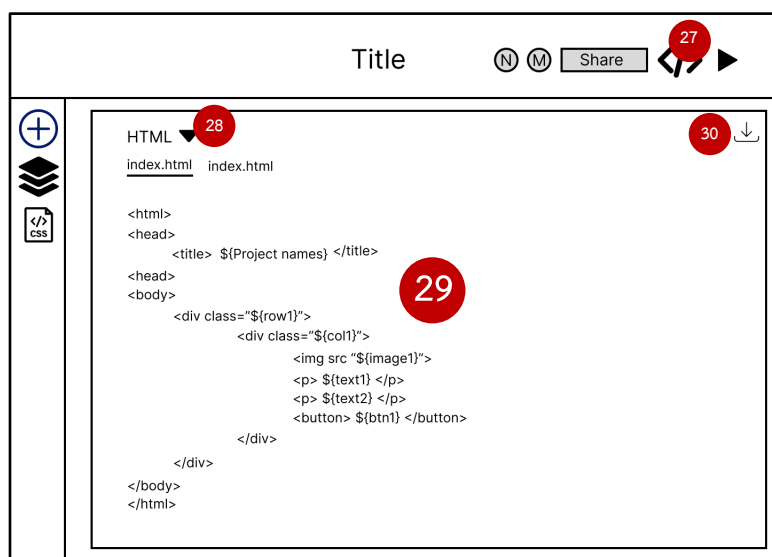
24. ใน Event OnClick บน Button ให้ตั้งค่า Post เป็น ON เพื่อส่งข้อมูลสินค้าที่ต้องการเพิ่มลงตะกร้าใน Database

25. กดเพื่อ Preview เว็บไซต์ที่ทำโครงร่างไว้



รูป 6.2.11 พรีวิวเว็บไซต์

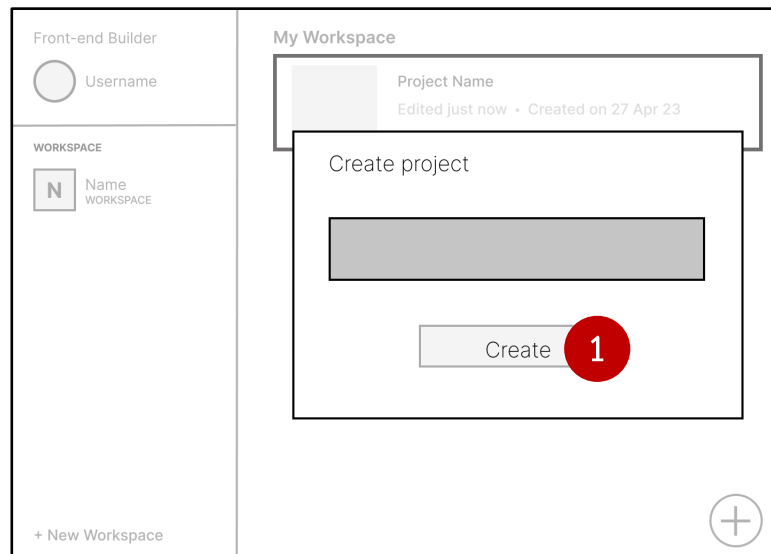
26. พรีวิวเว็บไซต์โดยเปิดแท็บใหม่บนบราวเซอร์ และแสดงผลการเรียกข้อมูลตามจำนวนสินค้าที่มีอยู่ใน API



รูป 6.2.11 กดปุ่ม Tag Code เพื่อพรีวิวในรูปแบบโค้ด

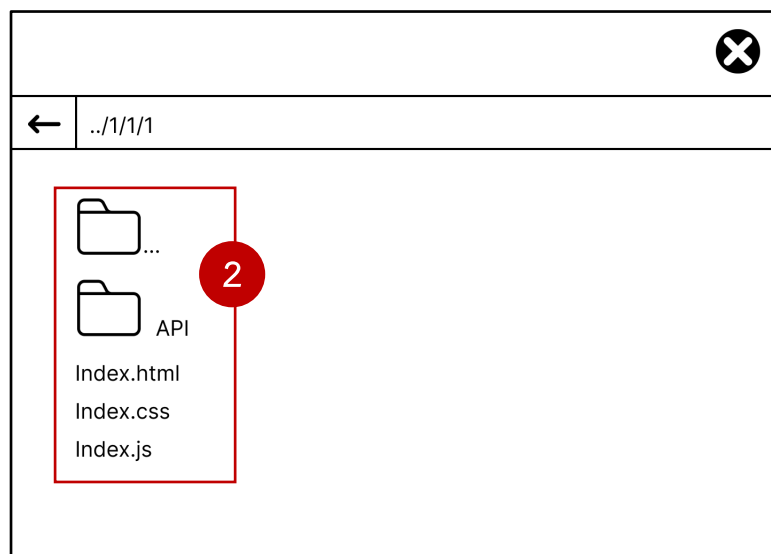
- 27. กดปุ่มเพื่อพรีวิวโค้ด
- 28. เลือกนามสกุลไฟล์โค้ดที่จะพรีวิว (.html, .css, .js)
- 29. แสดงโค้ดทั้งหมดจากนามสกุลไฟล์ที่เลือก
- 30. กดปุ่มเพื่อดาวน์โหลดโค้ดที่พรีวิวมาลงเครื่อง

6.2.3 การเปลี่ยนแปลงภายในไฟล์โปรเจ็ค เมื่อใช้งาน Elements ต่าง ๆ



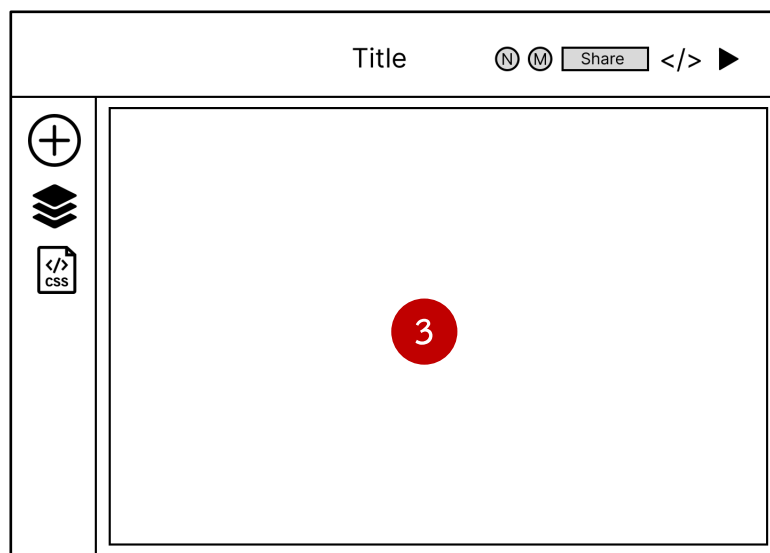
รูป 6.2.2.1 เริ่มการสร้างโปรเจ็ค

1. เริ่มสร้างโปรเจ็ค



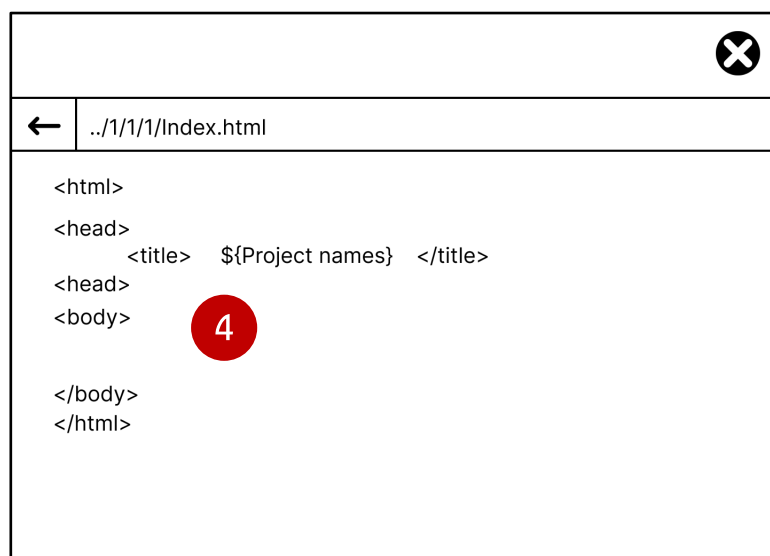
รูป 6.2.2.2 ไฟล์ต่าง ๆ ในโปรเจ็ค

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



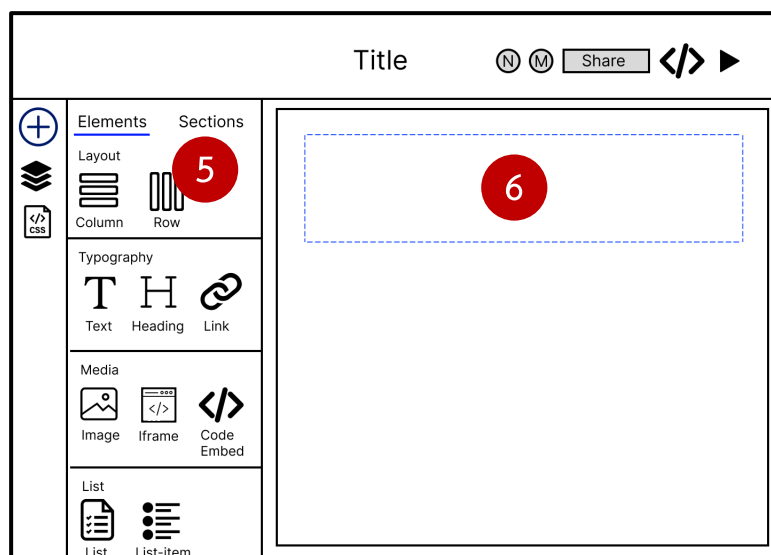
รูป 6.2.2.3 หน้าต่างการทำงาน

3. พื้นที่การทำงานของเว็บไซต์ (Background Default : White)



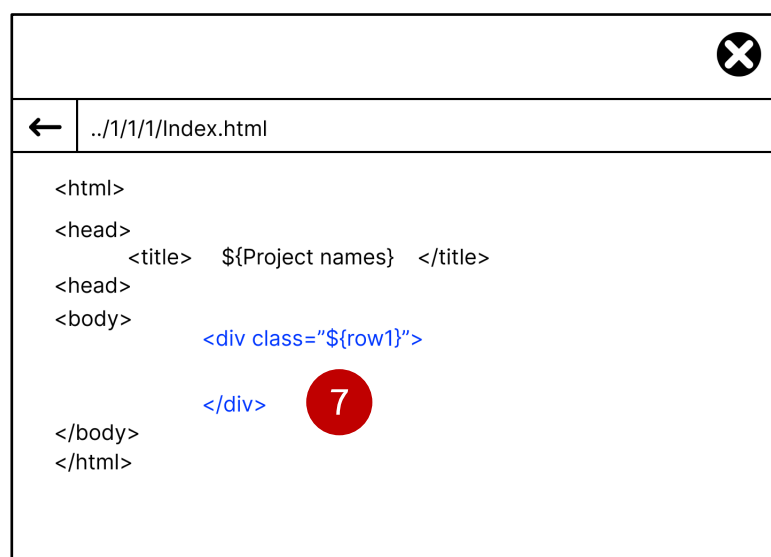
รูป 6.2.2.4 โค้ดพื้นฐานจากการสร้างโปรเจ็คครั้งแรก

4. นี่คือโค้ดเริ่มต้นที่อยู่ในไฟล์ Index.html ที่สร้างขึ้นมาพร้อมสร้างโปรเจ็ค



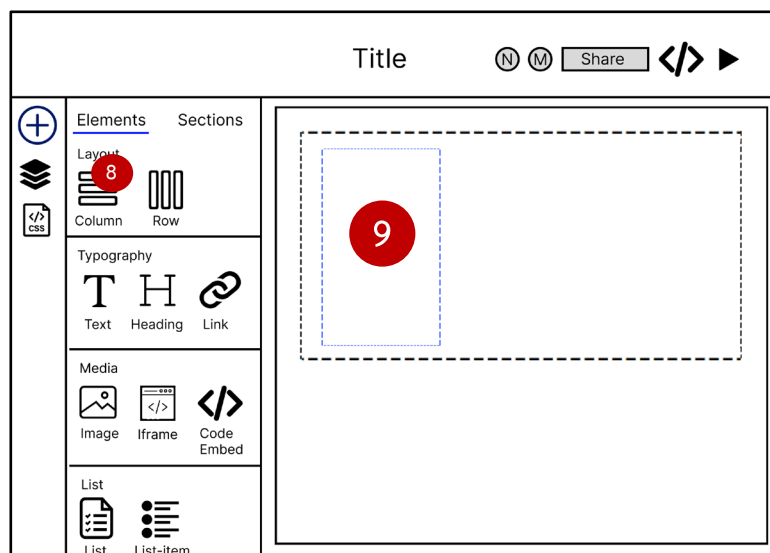
รูป 6.2.2.5 ใช้งาน Row เพื่อจัด Layout บนเว็บไซต์

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



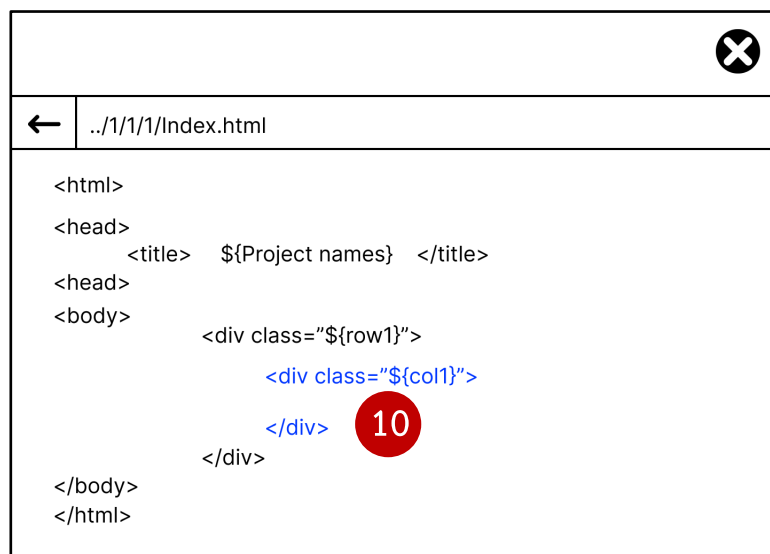
รูป 6.2.2.6 โค้ดหลังจากวาง Layout Row

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



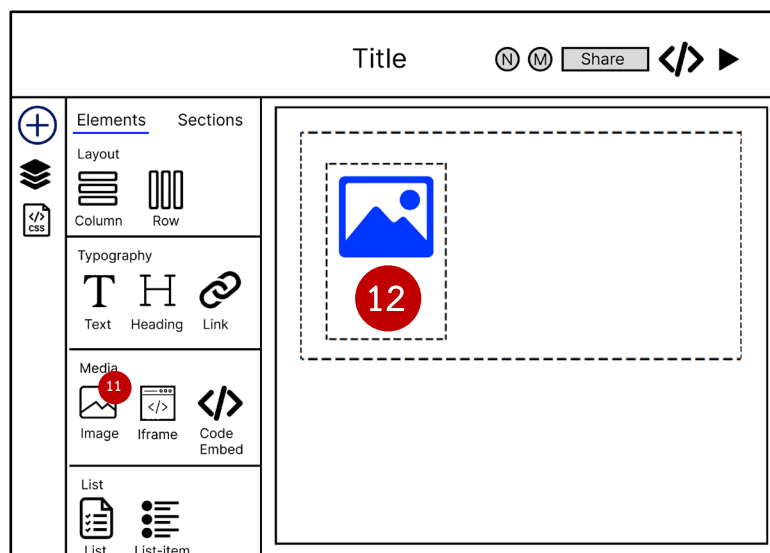
รูป 6.2.2.7 ใช้งาน Column เพื่อจัด Layout บนเว็บไซต์

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



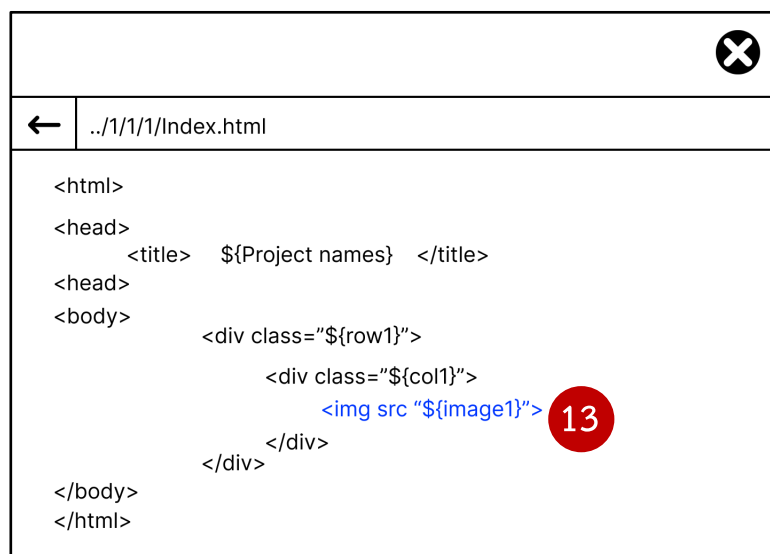
รูป 6.2.2.8 โค้ดหลังจากวาง Layout Col

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



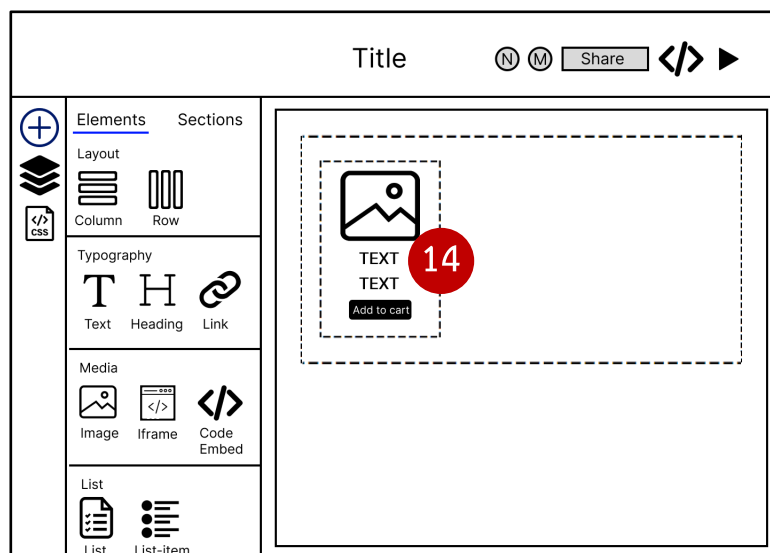
รูป 6.2.2.9 ใส่ Element Image

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



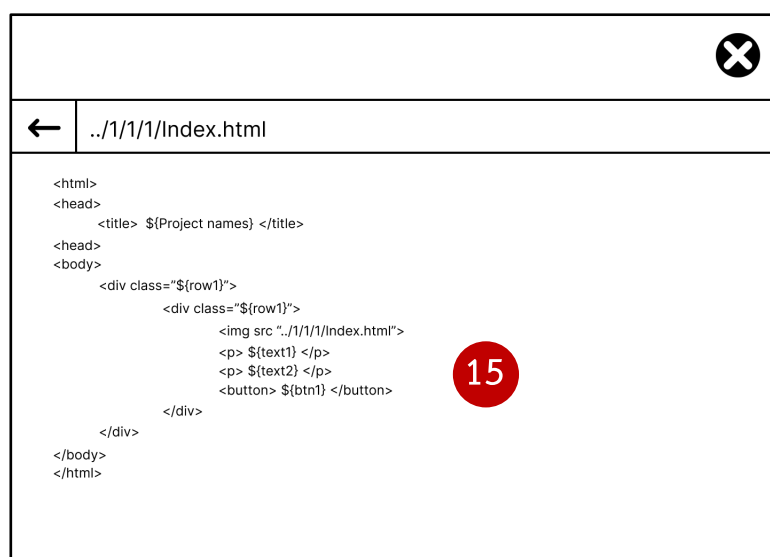
รูป 6.2.2.10 โค้ดหลังจากวาง Element Image

13. หลังจากวาง Element Image บนพื้นที่การทำงาน และ Tag `` จะถูกเพิ่มในไฟล์ Index.html ภายใต้ Tag `<div class='${col1}'>`



รูป 6.2.2.11 วาง Element ที่ใช้งานจนครบ

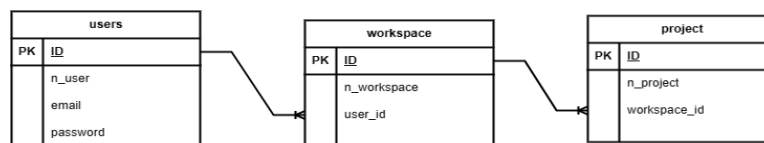
14. ใส่ Elements ที่ใช้งานให้ครบ



รูป 6.2.2.11 โค้ดทั้งหมดหลังจากวาง Element ที่ต้องการครบ

15. โค้ดทั้งหมดหลังจากวง Element ที่ใช้งานบนพื้นที่เว็บไซต์

6.2.3 ออกแบบ Database ที่ใช้งานภายในเว็บไซต์



Database นี้ที่ใช้งานจะใช้เก็บข้อมูลผู้ใช้ และที่เก็บไฟล์งานโปรเจ็ค โดยที่

1 user สามารถมี workspace ได้หลาย workspace และ 1 workspace สามารถ มีได้หลายโปรเจ็ค

ในส่วนของพื้นที่เก็บของไฟล์เดอร์การทำงานจะเก็บอยู่ในรูปแบบเส้นทาง (Path) ตัวอย่าง

.../User_id/workspace_id/project_id/

6.3 แนวทางการทดสอบและการวัดประสิทธิภาพ (Test and Performance Evaluation Approaches)

6.3.1. การทดสอบการสร้างเว็บไซต์โดยผู้ใช้ที่ไม่มีความรู้และทักษะด้านการเขียนโปรแกรม ให้สังเกตการทำงานของเครื่องมือและอินเตอร์เฟซกราฟิกว่าง่ายต่อการใช้งานหรือไม่ และผลลัพธ์ที่ได้สอดคล้องกับความต้องการหรือไม่

6.3.2. การทดสอบการปรับแก้ไขและปรับปรุงเว็บไซต์ที่สร้างขึ้นด้วยแพลตฟอร์ม Low-Code โดยทดสอบความสามารถในการเปลี่ยนแปลงรูปแบบและฟังก์ชันของเว็บไซต์ และตรวจสอบว่าการแก้ไขสามารถทำได้อย่างรวดเร็วและสะดวกสบายหรือไม่

6.3.3. การทดสอบการเชื่อมต่อและการทำงาน API จากแหล่งที่มาต่าง ๆ เพื่อตรวจสอบความสามารถในการสื่อสารและรับส่งข้อมูลระหว่างแพลตฟอร์ม Low-Code กับแหล่งข้อมูลภายนอก ทดสอบการเรียกใช้ API และตรวจสอบว่ารับข้อมูลอย่างถูกต้องและแสดงผลได้ถูกต้องหรือไม่

6.3.4. การทดสอบประสิทธิภาพการทำงานของเว็บไซต์ที่สร้างขึ้นด้วยแพลตฟอร์ม Low-Code โดยใช้ข้อมูลประสิทธิภาพที่สำคัญ เช่น เวลาที่ใช้ในการโหลดหน้าเว็บไซต์ การตอบสนองของหน้าเว็บไซต์ และประสิทธิภาพในการดำเนินการต่าง ๆ ซึ่งสามารถทำได้โดยใช้เครื่องมือและเทคนิคต่าง ๆ เช่น

6.3.4.1. การทดสอบประสิทธิภาพของเว็บไซต์ (Website Performance Testing): ใช้เครื่องมือและเทคนิคที่ช่วยวัดประสิทธิภาพของเว็บไซต์ เช่น Load Testing เพื่อตรวจสอบความทนทานของระบบในการรับข้อมูลจำนวนมาก หรือ Stress Testing เพื่อทดสอบประสิทธิภาพของระบบในเงื่อนไขการใช้งานที่มีความซับซ้อน

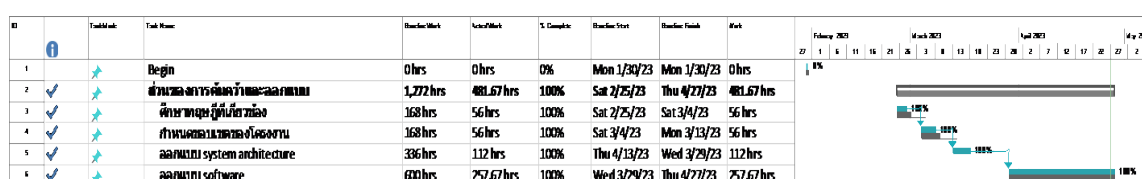
6.3.4.2. การทดสอบประสิทธิภาพของการเรียกใช้งาน API (API Performance Testing): ใช้เครื่องมือเพื่อทดสอบประสิทธิภาพในการเรียกใช้งาน API โดยวัดเวลาที่ใช้ในการรับส่งข้อมูลระหว่างแพลตฟอร์ม Low-Code และแหล่งข้อมูลภายนอก ตรวจสอบความเร็วในการตอบสนองและประสิทธิภาพการทำงานของ API

6.3.4.3. การทดสอบประสิทธิภาพของระบบความปลอดภัย (Security Performance Testing): ทดสอบประสิทธิภาพของระบบความปลอดภัยของแพลตฟอร์ม Low-Code โดยการทดสอบ

การรักษาความปลอดภัยของข้อมูล การตรวจสอบการเข้าถึงระบบ การจัดการสิทธิ์การเข้าถึง เพื่อให้มั่นใจในความปลอดภัยของระบบ

6.3.4.4. การทดสอบประสิทธิภาพของการประมวลผล (Processing Performance Testing): การทดสอบประสิทธิภาพของการประมวลผลเป็นกระบวนการที่ใช้เพื่อวัดและตรวจสอบประสิทธิภาพของระบบในการประมวลผลข้อมูลในแพลตฟอร์ม Low-Code ว่าสามารถทำงานได้อย่างมีประสิทธิภาพหรือไม่ โดยการทดสอบนี้มุ่งเน้นไปที่การประมวลผลข้อมูลที่ซับซ้อนหรือมีปริมาณมากเพื่อดูว่าระบบสามารถจัดการได้ในระดับที่ต้องการ และมีเวลาการประมวลผลที่ยอมรับได้ในสถานการณ์ปกติหรือสถานการณ์ที่มีการใช้งานอย่างสูงสุด

7. แผนการดำเนินงาน (Gantt Chart)



8. ประโยชน์ที่คาดว่าจะได้รับ (Expected Benefits)

- 8.1 ความสะดวกในการสร้างเว็บไซต์ ผู้ใช้ไม่จำเป็นต้องมีความเชี่ยวชาญในการเขียนโค้ดเว็บไซต์เพื่อสร้างเว็บไซต์ขึ้นมาใหม่ ด้วยเครื่องมือสร้างเว็บไซต์เชิงสร้างสรรค์ เช่น CMS (Content Management System) หรือเครื่องมือสร้างเว็บไซต์ที่มีอยู่ในระบบ ผู้ใช้สามารถสร้างเว็บไซต์ได้อย่างง่ายและสะดวกด้วยการลาก-วางองค์ประกอบต่าง ๆ และปรับแต่งรูปแบบต่าง ๆ ตามต้องการของตนเอง ทำให้กระบวนการสร้างเว็บไซต์เป็นสิ่งที่ไม่ซับซ้อนและสนุกสนานมากขึ้นประหยัดเวลาในการพัฒนาโปรแกรมเว็บส่วนหน้า (Front-End)
- 8.2 ประหยัดเวลาและทรัพยากร การใช้เครื่องมือสร้างเว็บไซต์เชิงสร้างสรรค์ช่วยลดเวลาในกระบวนการสร้างเว็บไซต์เมื่อเทียบกับการเขียนโค้ดเว็บไซต์เองตั้งแต่เริ่มต้น นอกจากนี้ยังไม่ต้องใช้ทรัพยากรมากในการจ้างนักพัฒนาเว็บไซต์หรือทีมพัฒนาให้เสร็จสมบูรณ์ ทำให้ผู้ใช้สามารถปรับแต่งและพัฒนาเว็บไซต์นั้นเป็นไปได้อย่างอิสระตามความต้องการและความคิดสร้างสรรค์ของตนเอง ผู้ใช้สามารถทำการแก้ไขหรือปรับปรุงเว็บไซต์ได้ตลอดเวลาโดยไม่ต้องพึ่งพาผู้อื่น
- 8.3 ความยืดหยุ่นในการปรับแต่ง เครื่องมือสร้างเว็บไซต์เชิงสร้างสรรค์มักมีตัวเลือกและฟีเจอร์ที่หลากหลายทำให้ผู้ใช้สามารถปรับแต่งเว็บไซต์ให้เหมาะสมกับความต้องการและลักษณะของธุรกิจหรือผลงานที่ต้องการสร้างขึ้น สามารถเลือกออกแบบที่ตรงกับความต้องการของตนเองได้ ซึ่งเป็นประโยชน์สำคัญที่ช่วยให้ผู้ใช้สามารถสร้างเว็บไซต์ที่มีความเป็นเอกลักษณ์และแตกต่างจากเว็บไซต์อื่น ๆ
- 8.4 ลดต้นทุนและเวลาในการสร้างเว็บไซต์ การใช้เครื่องมือสร้างเว็บไซต์ช่วยลดความซับซ้อนและเวลาในการสร้างเว็บไซต์ โดยไม่จำเป็นต้องมีความเชี่ยวชาญในการเขียนโค้ด การสร้างเว็บไซต์ด้วยเครื่องมือเชิงสร้างสรรค์อาจช่วยประหยัดค่าใช้จ่ายในการจ้างนักพัฒนาเว็บไซต์ภายนอกและเพิ่มประสิทธิภาพในกระบวนการพัฒนา

- 8.5 การทดสอบและปรับปรุง เครื่องมือสร้างเว็บไซต์มักมีฟีเจอร์ในการทดสอบและปรับปรุงเว็บไซต์ ผู้ใช้สามารถทดสอบความเร็วการโหลดหน้าเว็บ ประสิทธิภาพของ เพื่อปรับปรุงและปรับแต่งเว็บไซต์ให้สามารถทำงานได้ดีที่สุด

9. ผลการศึกษาเทคโนโลยีที่ใช้พัฒนา

9.1 HTML

HTML (HyperText Markup Language) คือภาษาคอมพิวเตอร์ที่ใช้สำหรับสร้างและแสดงเว็บไซต์บนอินเทอร์เน็ต โดย HTML ใช้รหัสแท็ก (tag) เพื่อบอกว่าเนื้อหาบนหน้าเว็บไซต์ควรแสดงอย่างไร และจะเรียกใช้รูปแบบต่าง ๆ เช่น ข้อความ รูปภาพ ลิงก์ ตาราง และอื่นๆ โดย HTML เป็นส่วนหนึ่งของเทคโนโลยีเว็บและเป็นภาษาพื้นฐานที่จำเป็นสำหรับการพัฒนาเว็บไซต์ ในปัจจุบัน HTML ได้พัฒนาออกมาเป็นเวอร์ชันล่าสุดคือ HTML5 ซึ่งมีฟีเจอร์และความสามารถที่มากกว่าเวอร์ชันก่อนหน้า

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 ยังช่วยลดความซ้ำซ้อนในการเขียน HTML และเพิ่มความยืดหยุ่นในการปรับแต่งหน้าเว็บไซต์ให้ตรงตามความต้องการของผู้ใช้งาน

9.3 JavaScript

JavaScript คือภาษาโปรแกรมมิ่งสำหรับเว็บไซต์ที่ใช้งานได้บนเว็บเบราว์เซอร์ โดย JavaScript ช่วยให้ผู้พัฒนาเว็บไซต์สามารถสร้างและควบคุมพฤติกรรมต่าง ๆ ของเว็บไซต์ได้ เช่น การตรวจสอบข้อมูลที่ผู้ใช้ป้อนเข้ามาในแบบฟอร์ม การสร้างเอฟเฟกต์และการเปลี่ยนแปลงสไตล์หน้าเว็บไซต์ เป็นต้น นอกจากนี้ JavaScript ยังช่วยเพิ่มประสิทธิภาพของการโหลดหน้าเว็บไซต์ และช่วยให้เว็บไซต์มีประสิทธิภาพที่ดีขึ้น ซึ่งทำให้ JavaScript เป็นภาษาโปรแกรมมิ่งที่สำคัญและอยู่ในแนวเทคโนโลยีเว็บที่ได้รับความนิยมมากที่สุดในปัจจุบัน

9.4 ReactJS

React.js คือไลบรารี (library) สำหรับพัฒนาเว็บเว็บไซต์ ซึ่งเป็นส่วนหนึ่งของเทคโนโลยี JavaScript ที่ใช้งานได้บนเว็บเบราว์เซอร์ โดยเฉพาะในการพัฒนาเว็บเว็บไซต์ที่มีการอัปเดตข้อมูลแบบ real-time หรือการสร้าง User Interface (UI) ที่เปลี่ยนแปลงได้รวดเร็ว React.js มีความยืดหยุ่นสูงและมีความสามารถในการสร้าง Component ที่สามารถนำไปใช้ได้ซ้ำได้เป็นส่วนต่อประสานกับการพัฒนา Backend อย่าง Node.js ได้ง่าย นอกจากนี้ React.js ยังมีชุดเครื่องมือ (toolset) สำหรับการทำ Unit testing และ End-to-End testing ที่ช่วยให้การทดสอบและการปรับปรุงคุณภาพของเว็บเว็บไซต์ด้วย React.js ง่ายขึ้น

9.5 ExpressJS

ExpressJS เป็นเฟรมเวิร์กของภาษา JavaScript ที่ใช้สร้างแอปพลิเคชันเว็บแบบเซิร์ฟเวอร์ไซด์ (server-side) โดย ExpressJS เป็นเฟรมเวิร์กที่เบาและมีขนาดเล็ก แต่มีความยืดหยุ่นและสามารถสร้างเว็บแอปพลิเคชันที่มีประสิทธิภาพได้อย่างรวดเร็ว

ExpressJS มีลักษณะเป็นเฟรมเวิร์กที่เบาและมีแกนกลางที่เรียกว่า "Express" เป็นตัวควบคุมหลักในการกำหนดเส้นทาง (routing) และการจัดการกับคำขอ (request) และการตอบสนอง (response) ของแอปพลิเคชันเว็บ สิ่งนี้ช่วยให้นักพัฒนาสามารถสร้างเว็บแอปพลิเคชันที่เป็นไปตามความต้องการของพวกเขาได้อย่างสะดวกและเร็วด้วย JavaScript

9.6 Amazon Web Services (AWS)

Amazon Web Services (AWS) เป็นแพลตฟอร์มคลาวด์ (cloud platform) ที่ให้บริการในรูปแบบของ Infrastructure as a Service (IaaS) และ Platform as a Service (PaaS) โดย Amazon Web Services เป็นบริการคลาวด์อันดับหนึ่งของโลก มีบริการหลากหลายรูปแบบเช่น การจัดการคลาวด์ (cloud management), บริการคลาวด์คอมพิวติ้ง (cloud computing), บริการเกี่ยวกับฐานข้อมูลและการจัดเก็บข้อมูล (database and storage services), บริการสำหรับการวิเคราะห์ข้อมูล (analytics), บริการเกี่ยวกับเครือข่ายและความปลอดภัย (networking and security services) และอื่น ๆ อีกมากมาย โดย Amazon Web Services เป็นบริการที่ใช้งานได้ง่ายและเหมาะสำหรับการใช้งานทั้งในธุรกิจขนาดเล็กและขนาดใหญ่ และได้รับการนิยมเป็นอย่างมากในวงการเทคโนโลยี

10. เอกสารอ้างอิง (Reference)

10.1 <https://www.w3schools.com/html/>

10.2 <https://www.w3schools.com/css/>

10.3 <https://www.w3schools.com/js/>

10.4 <https://react.dev/>

10.5 <https://expressjs.com/>

10.6 <https://aws.amazon.com/th/>