ข้อ 1.

โครงสร้างการทำงานของ web / application ที่ทำงานออกเป็นสองส่วนที่อาศัย Internet ในการติดต่อส่งข้อมูลกันของฐานข้อมูลฝั่งserver กับ ฝั่งclient

ซึ่งการแสดงผลบนหน้าจอทั้ง บน website หรือ บน application จากมือถือ จะเป็นส่วนในการทำงานของฝั่ง front-end หรือเรียกว่าหน้าบ้าน ซึ่ง Front-end เป็นส่วนที่ผู้ใช้สามารถมองเห็นและปฏิสัมพันธ์กับเว็บไซต์ได้โดยตรง มีลักษณะหลักได้แก่ User Interface (UI) ที่แสดงหน้าตาของเว็บไซต์หรือการดึงข้อมูลจากฐานข้อมูลมาทำการแสดงผลและช่วยให้ผู้ใช้เข้าใจและใช้งานได้ง่าย ๆ ซึ่งมีองค์ประกอบหลายประการ เช่น หน้าโฮม, เว็บเพจ, รูปภาพ, ลิงก์ และสื่อสารกับฝั่ง Back-end โดยใช้ HTTP requests เพื่อดึงข้อมูลหรือส่งข้อมูลกลับไปที่เซิร์ฟเวอร์

เมื่อมีการใช้งานในระบบต่างๆจำเป็นที่จะต้องมีการเก็บข้อมูลในฐานข้อมูลเพื่อรองรับข้อมูลที่มากขึ้นหรือสามารถจัดการกับข้อมูลได้ ซึ่งจะเป็นในส่วนของ Backend หรือเรียกว่าหลังบ้าน Back-end เป็นส่วนที่ทำงานอยู่ในเบื้องหลังและรับผิดชอบในการจัดการข้อมูล, การประมวลผล, และการตอบสนองต่อคำขอจากฝั่ง Front-end มีความสำคัญในการรักษาความปลอดภัยของข้อมูล และในการจัดการการเปลี่ยนแปลงบนเว็บไซต์ การจัการข้อมูลมีหลากหลายรูปแบบ นั้นก็คือ

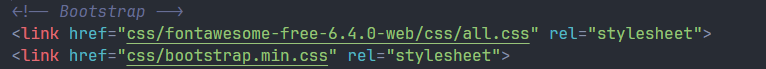
Database server มีหน้าที่จัดเก็บข้อมูลและให้บริการการเข้าถึงข้อมูลในรูปแบบที่มีโครงสร้าง, โดยใช้ระบบจัดการฐานข้อมูล (Database Management System - DBMS) เช่น MySQL, PostgreSQL, Microsoft SQL Server, หรือ Oracle

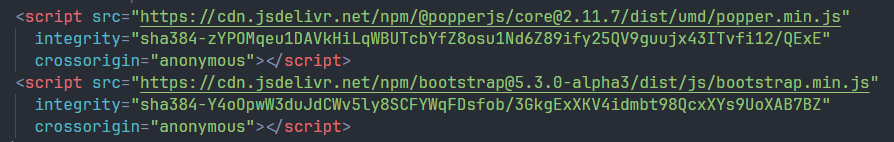
Web server ให้บริการเว็บไซต์หรือแอปพลิเคชัน, รับคำขอจากผู้ใช้และส่งข้อมูลหน้าเว็บหรือแอปพลิเคชันกลับไปยังผู้ใช้

File server มีหน้าที่จัดเก็บและจัดการไฟล์ที่ใช้ในการทำงานของเว็บไซต์หรือแอปพลิเคชัน การทำงานร่วมกันของทั้งสามประการนี้เป็นสิ่งที่สำคัญในการสร้างและบริหารจัดการเว็บไซต์หรือแอปพลิเคชัน ทั้งสามนี้ต้องการความสอดคล้องและประสิทธิภาพในการทำงานเพื่อให้ผู้ใช้สามารถใช้งานได้สมบูรณ์และรวดเร็ว.

ข้อ 2.

**Bootstrap**

เป็นโครงการโอเพนซอร์สที่เป็น Front-end framework ที่ถูกพัฒนาโดยทีม Twitter และมีเป้าหมายเพื่อช่วยให้การสร้างเว็บไซต์และแอปพลิเคชันดูสวยงาม และมีประสิทธิภาพได้ง่ายขึ้น Bootstrap จะเป็น กลุ่มโค้ดที่รวมชุดคำสั่งของ HTML, CSS, JavaScript ไว้ด้วยกัน ที่พัฒนาขึ้นมาเพื่อกำหนดกรอบหรือรูปแบบของการพัฒนาเว็บไซต์

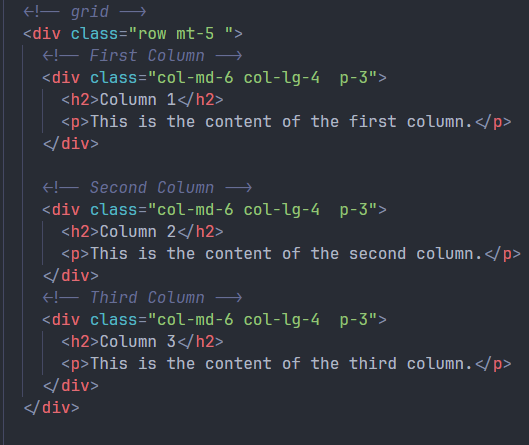
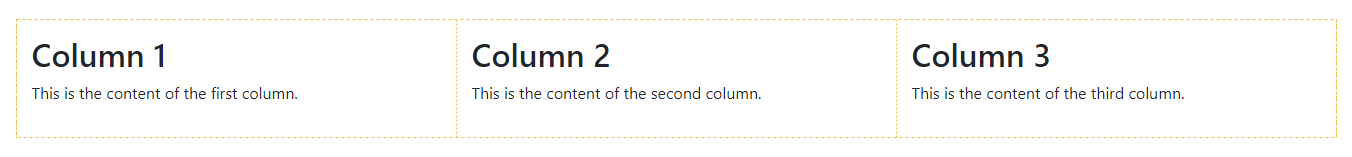
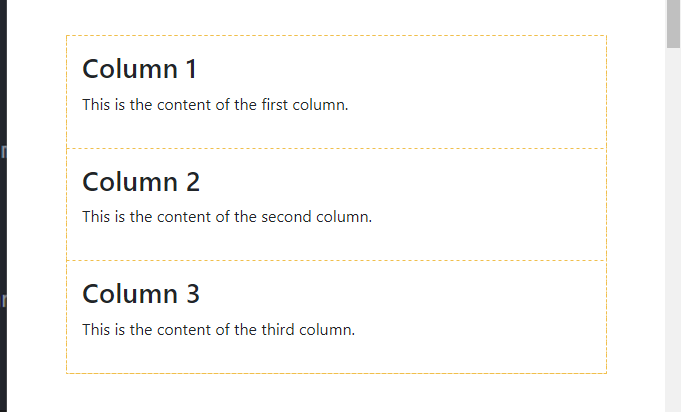


ตัวอย่างการนำเข้า Bootstrap มาใช้ใน project ซึ่งจะเห็นได้ว่ามีการ import ในส่วนของ CSS และ JavaScript Plugin เข้ามาด้วย

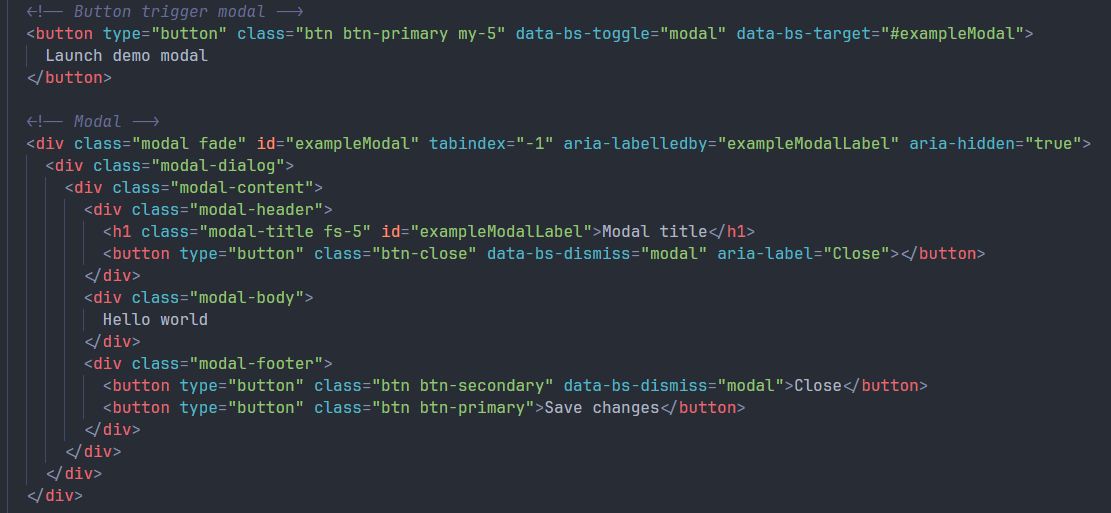
การใช้ Bootstrap

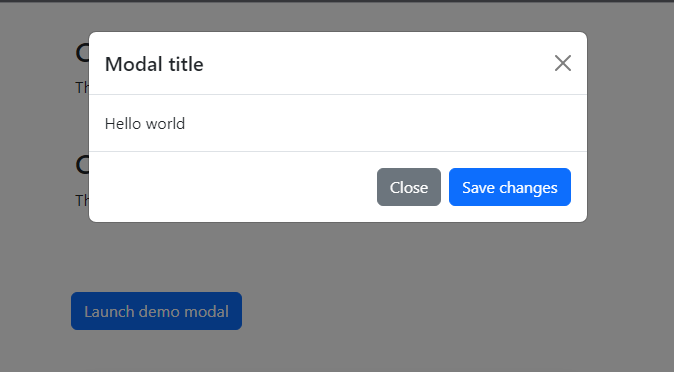
ช่วยลดเวลาในการพัฒนาและทดสอบเว็บไซต์, และช่วยให้โปรเจกต์มีหน้าตาที่สวยงามและ เป็นมาตรฐานได้ง่ายขึ้น. ซึ่งจะทำให้การพัฒนาเว็บไซต์ทำได้มากขึ้นและใช้งานได้ง่ายและสะดวกรวดเร็วยิ่งขึ้น รวมถึงการรองรับ Smart Device (สมาทดีไว) หรือ Mobile First (โมบาย เฟิร์ส ) ที่หลากหลายต่อการใช้งาน โดยรองรับรูปแบบการพัฒนาที่สามารถรองรับหลากหลาบ Platform ให้ใช้งานนั่นเอง

การใช้งาน รูปแบบ Grid System, Bootstrap ใช้ระบบ grid ที่มีคอลัมน์และแถว ทำให้ง่ายต่อการจัดวางและจัดการเนื้อหาของเว็บไซต์.



จาก code ด้านบนผมใช้ คำสั่ง column ในการจัดเรียงองค์ประกอบ ของ column ทั้ง 3 column หาก โดยจะมีการวาวเรียงเป็นยาวเมื่อขนาดหน้าจอกว้าง และจะเป็นแถวยาวเมื่อขนาดหน้าจอเล็กลง

การใช้งาน CSS Style /Component / JavaScript Plugins



Bootstrap อำนวยความสะดวกให้ผู้พัฒนาสร้าง component ต่างได้อย่างรวดเร็ว โดยจะมีcomponent ที่เป็น template ให้เอาใช้ได้ทันที เช่น การใช้ Modal จากการกดปุ่ม และสามารถป้อนคำสั่ง Bootstrap CSS ลงที่ tag ของ HTML ในการตกแต่ง component นั้นได้

**CSS BEM (Blocks, Elements and Modifiers)**

การออกแบบ CSS ด้วย BEM เป็นการเขียน CSS โดยการตั้งชื่อ class โดย

Block คือ element ที่มีความหมายในตัวมันเอง เช่น header input menu

Element เป็นส่วนหนึ่งชอง block ที่ไม่สามารถมีความหมายเองได้และมีความเกี่ยวข้องกับบล็อกที่เป็นส่วนหนึ่งของโครงสร้างหลัก เช่น menu item, list item

Modify ใช้เพื่อเปลี่ยนรูปลักษณ์หรือพฤติกรรมของเนื้อหา เช่น disable ,color ,size big

การเขียน BEM

Block – header , container , menu , checkbox , input

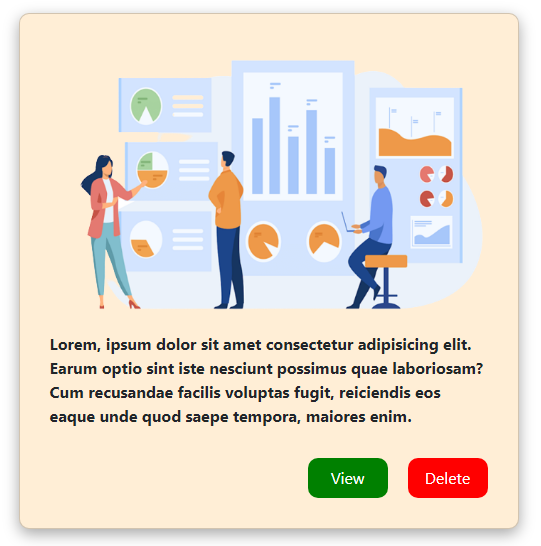
Element - menu\_\_item , list\_\_item , checkbox\_\_caption , header \_\_title

Modifier – block--disabled , block--highlighted , block--checked , block--fixed , block--size-big , block—color-yellow



หน้า HTML ผมได้สร้าง Card ขึ้น โดย block นั้นผมตั้งชื่อ class ว่า card และข้างใน block มีส่วนของ รูปภาพ (card\_\_img) ส่วนอธิบาย (card\_\_desc) ส่วนของปุ่ม (card\_\_btn) และตัวปุ่ม (cart\_btn) ซุ่งทั้งหมดนี้คือ element โดยผมต้องการจะตกแต่งปุ่มต่างๆ จึงใช้ (btn--success) และ (btn---denger ) ซึ่งเป็นการตั้งชื่อแบบ Modify ในการกำหนดรูปแบบปุ่มที่ต้องการ

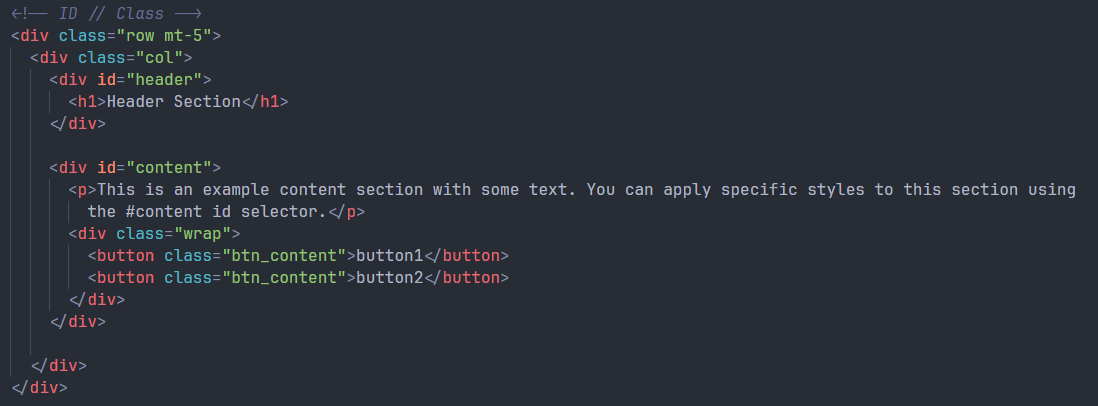
.

เมื่อมีการตั้งชื่อ class กำการกำหนด CSS แต่ละ Class

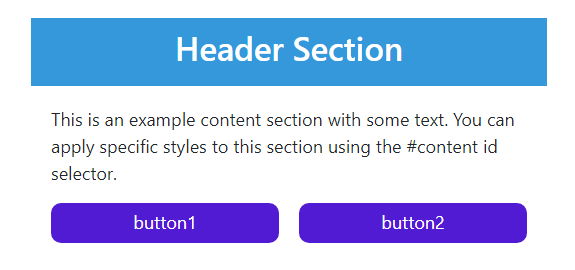
นี่คือผลลัพธ์ของการสร้าง card โดยใช้ CSS รูปแบบ BEM

**#id / .Class**

ID และ Class เป็นตัวเลือกที่ใช้ในการกำหนดรูปแบบ CSS โดย ID ใช้เลือกองค์ประกอบตามตัวระบุเฉพาะ ส่วน class เป็นการเลือกองค์ประกอบตามลักษณะของคลาส ซึ่งelementสามารถใช้classร่วมกันได้







จาก code ผมได้กำหนด id ในส่วนของ header Section และ content เพื่อเป็นการบ่งบอกลักษณะที่เจาะจง แต่ใช้classในส่วนของปุ่ม เพราะต้องการให้ปุ่มนั้นมีลักษณะหรือรูปแบบที่เหมือนกัน จึงสามารถกำหนด class ครั้งเดียวแต่ใช้ได้กับทุกปุ่ม



การใช้งาน id ที่เห็นได้บ่อยนั้นคือการกำหนดตัวตนที่เฉพาะเจาะจง

จาก code ตัว Modal ผมกำหนด id ให้ชื่อว่า exampleModal เพื่อเป็นตัวแสดงผล เมื่อมีการกดปุ่มจาก button ที่อยู่ด้านบน โดย จะมี attribute (data-bs-target) ที่ชี้ไปยัง ”#exampleModal” ซึ่งเป็นการกำหนดให้ id ที่ตรงกับชื่อนั้นมีการ action เกิดขึ้น

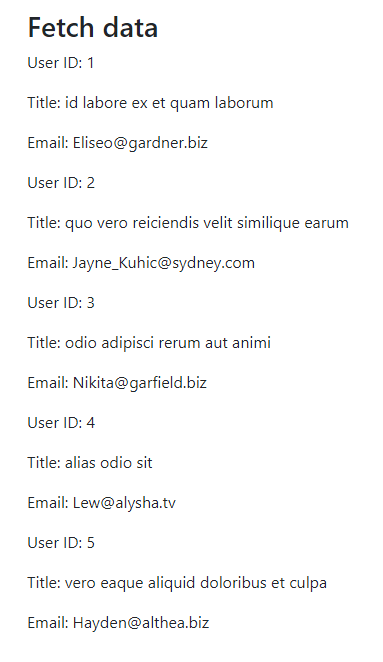
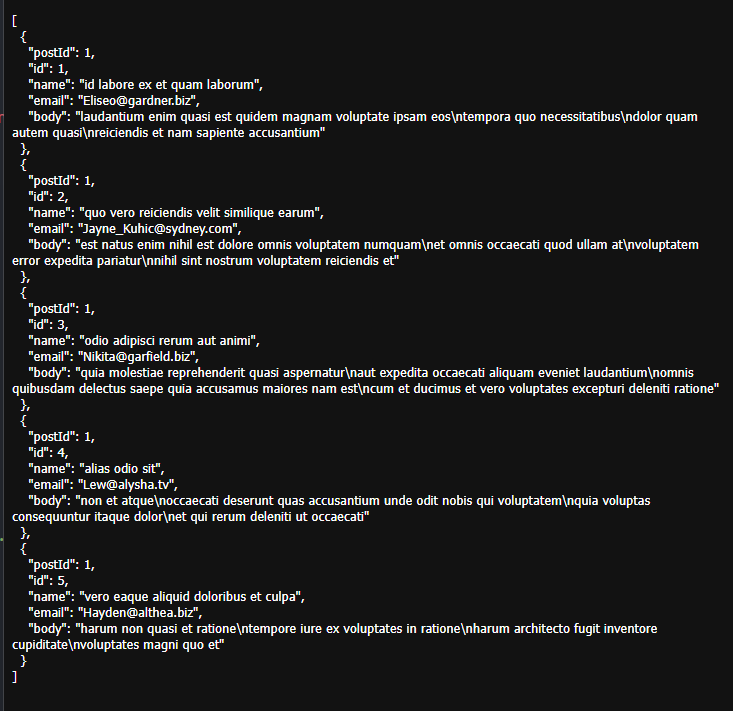
**AJAX**

AJAX ย่อมาจาก "Asynchronous JavaScript and XML" ซึ่งเป็นเทคนิคหรือเทคโนโลยีในการพัฒนาเว็บแอปพลิเคชัน (Web Application) ที่ใช้ JavaScript เพื่อส่งข้อมูลไปยังเซิร์ฟเวอร์และรับข้อมูลกลับมา โดยทำการโหลดข้อมูลโดยไม่ต้องรีเฟรชหน้าเว็บทั้งหมด ทำให้เกิดประสิทธิภาพการใช้งานที่ดีขึ้นและลดการทำงานซ้ำซ้อนของเว็บไปจำนวนมาก

ซึ่งปกติแล้วในภาษาสคริปต์ที่ใช้งานกับเว็บไซต์จะมีการทำงานประมวลผลแบบเป็นลำดับ (synchronous) โดยที่คำสั่งแรกจะต้องประมวลผลให้เสร็จสิ้นก่อนแล้วถึงจะทำงานในคำสั่งถัดไป แต่กระบวนการทำงานแบบเอแจ๊กซ์เมื่อบราวเซอร์ (Browser) ร้องขอข้อมูลไปยังเซิร์ฟเวอร์ (Server) บราวเซอร์จะไปทำงานคำสั่งถัดไปทันที (asynchronous) โดยที่ไม่ต้องรอการตอบกลับจากเซิร์ฟเวอร์ก่อน



จาก code ผมได้ทำการกำหนด function ในการเรียกข้อมูลจาก server ในคำสั่ง fetch เพื่อส่งมายัง browser เมื่อมีการส่งข้อมูลที่ผิดพลาด จะมีการแสดงผล errorทันที โดยที่ไม่ต้องรอการตอบกลับจาก server ก่อน



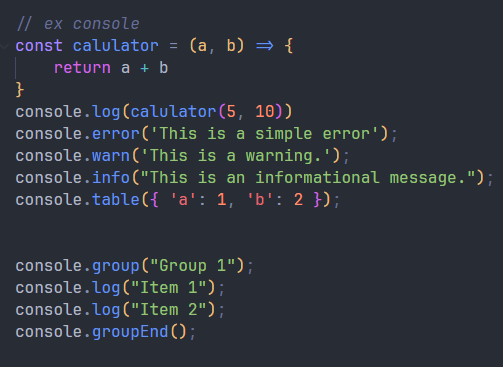
ข้อมูล ฝั่ง server

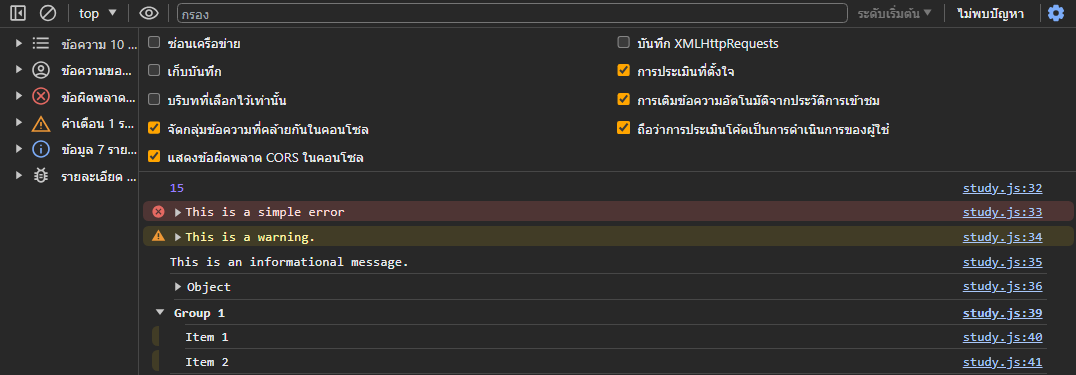
การแสดงผลข้อมูลหลังจากการเรียกข้อมูล

**Console**

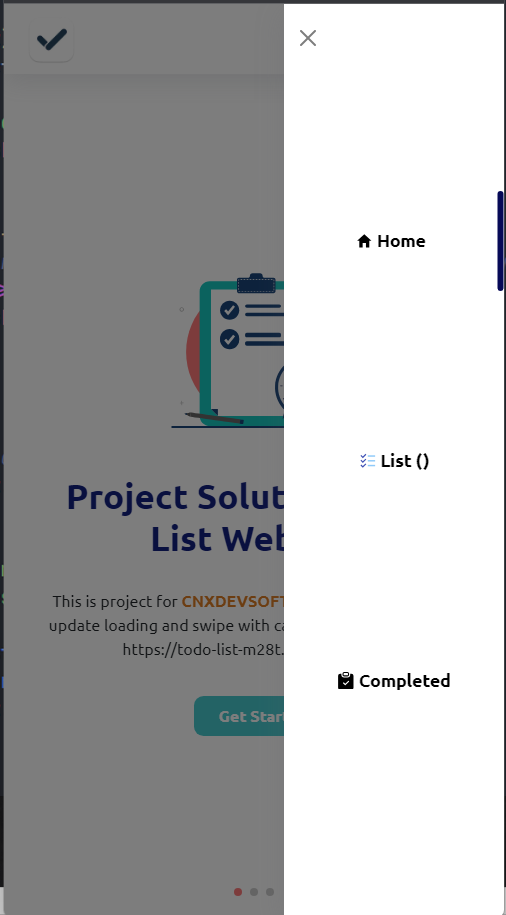
Console เป็นเครื่องมือที่ใช้ในการแสดงข้อมูลต่างๆที่กำหนด หรือข้อผิดพลาดจาก JavaScript ในโปรแกรมเบราว์เซอร์ มีประโยชน์ในการทำ debugging และทดสอบโปรแกรม JavaScript

ใน environment ของ web browser หรือ Node.js runtime environment. console มีหลายเมธอดที่ช่วยให้แสดงข้อมูล, ดูข้อผิดพลาด, และดูข้อมูลอื่น ๆ ที่เกี่ยวข้องกับการทำงานของโปรแกรมของผู้พัฒนา

จาก code ผมได้สร้าง function ในการบวกเลข2ตัว โดยผมจะใส่ค่าเป็นเลข 5 กับ 10 ในการบวกกัน และให้แสดงผลบน console

เมื่อเปิดเครื่องมือ dev tool ใน browser ส่วนของ console จะปรากฏคำตอบของ function

นอกจากconsole เอาไว้ใช้แสดงผลแล้ว ยังสามารถเอาไว้เช็คกับการทำงานของเราได้ เช่น

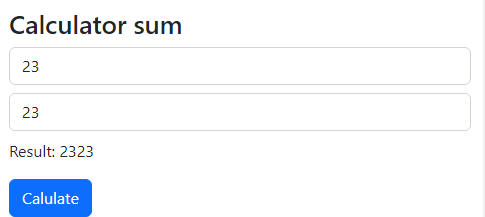


ผมสร้าง function ที่เอาไว้เวลากดปุ่มด้านนอกเมนู จะทำการปิดเมนู โดยหากเกิดปัญหาหากเมนูไม่ปิด ผมสามารถใช้ console.log ในการเช็ดได้ว่าเราได้กดปุ่มที่อยู่ด้านนอกจริงๆหรือไม่ หากเมนูถูกปิดจะให้แสดงผลคำว่า clicked

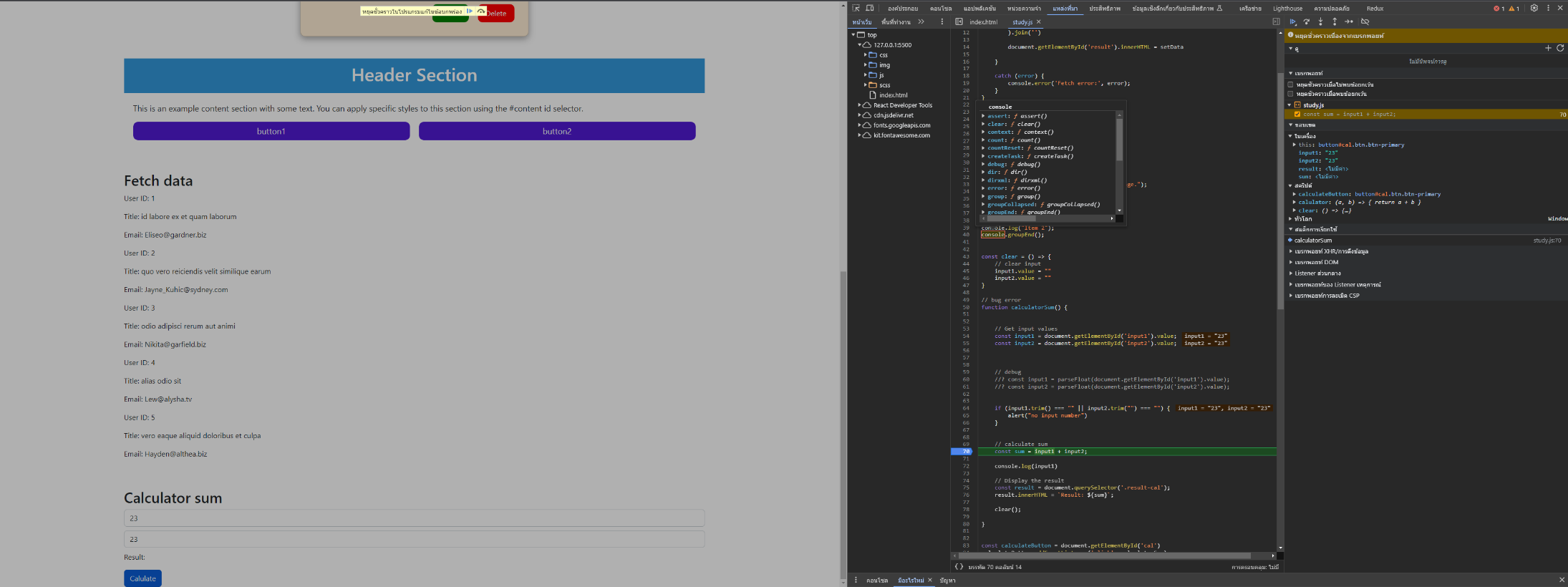
**Debugger**

Debugger คือเครื่องมือหรือโปรแกรมที่ใช้ในการช่วยวิเคราะห์และแก้ไขข้อผิดพลาด (bugs) ในโปรแกรมคอมพิวเตอร์ หรือโปรแกรมที่กำลังทำงาน. Debugger ช่วยให้นักพัฒนาซอฟต์แวร์สามารถตรวจสอบสถานะของโปรแกรมในระหว่างการทำงาน, ตรวจสอบค่าตัวแปร, และทำการหยุดโปรแกรมที่จุดที่กำหนดไว้เพื่อให้นักพัฒนาสามารถวิเคราะห์ข้อมูลและการทำงานของโปรแกรมได้.





จาก code ผมได้สร้าง form ในการบวกเลข จากการป้อนข้อมูลใน input box โดยจะให้มีการแสดงคำตอบที่เกิดจากการบวก ของตัวเลขจากทั้ง2 input box แต่ผลลัพธ์คือการที่ตัวเลขนั้นต่อกันแทนการบวกกัน



ผมได้ใช้ debugger ใน dev tool ในการหาสิ่งที่ผิดพลาด โดยการจิ้มที่ code คำสั่งบวกกัน และสังเกตเห็นว่าค่านั้นที่ป้อนลงใน input box เป็นค่า string ทำให้เกิดการเรียงต่อกันแทนการบวก



ผมเลยแก้ไข code โดยการกำหนดข้อมูลที่ป้อนใน input box ให้เปลี่ยนเป็น type number เพื่อสามารถนำข้อมูลมาบวกันได้นั้นเอง

