



# DaisyUI Web Programming

Intermediate

(Integrative-Generative AI Edition)

Layout System

Navigation & Menus

DaisyUI Utilities

Theme Customization

Theme Customization

Bibliography

Author: Student Price Book Center

# คำนำ

ในยุคปัจจุบัน การพัฒนาเว็บไซต์และแอปพลิเคชันสมัยใหม่ไม่ได้จำกัดอยู่เพียงการสร้างฟังก์ชันการทำงานให้สมบูรณ์หรือการแสดงผลที่ถูกต้องเท่านั้น แต่ยังต้องเน้นไปที่การออกแบบ User Interface และ User Experience ให้สวยงาม เข้าใจง่าย และตอบสนองต่อผู้ใช้ได้อย่างมีประสิทธิภาพ **DaisyUI** ซึ่งเป็นคอมโพเนนต์และยูทิลิตี้เสริมสำหรับ **Tailwind CSS** ได้รับความนิยมอย่างรวดเร็วเนื่องจากช่วยให้นักพัฒนาสามารถสร้างเว็บไซต์ที่สวยงามและทันสมัยได้อย่างรวดเร็ว โดยไม่ต้องเขียน CSS จากศูนย์

หนังสือเล่มนี้ **DaisyUI Web Programming: Intermediate** ถูกออกแบบมาเพื่อพาผู้อ่านที่มีพื้นฐาน Tailwind CSS อยู่แล้ว ไปสู่ระดับถัดไปของการพัฒนาเว็บไซต์ ด้วยการเจาะลึกการใช้งานคอมโพเนนต์และ Utilities ของ DaisyUI พร้อมทั้งการออกแบบ Layout, Navigation, Theme, และองค์ประกอบ interactive ต่าง ๆ ในระดับที่สามารถประยุกต์ใช้กับโปรเจกต์จริงได้

บทที่ 6 **Layout System** เป็นบทเริ่มต้นของเนื้อหาสำคัญ โดยเน้นการสร้างโครงสร้างหน้าเว็บที่ยืดหยุ่นและสวยงาม ผู้อ่านจะได้เรียนรู้การใช้ **Grid** และ **Flex** ร่วมกับ DaisyUI เพื่อจัดวางองค์ประกอบต่าง ๆ เช่น Hero Section, Stats, และ Carousel พร้อมแนวทางการทำ **Responsive Design** ที่ปรับตัวได้กับทุกอุปกรณ์ นอกจากนี้ยังมีตัวอย่างบูรณาการและ Ultimate Example ที่ช่วยให้เห็นภาพการใช้งานจริง

บทที่ 7 **Navigation & Menus** พาผู้อ่านไปสำรวจระบบนำทางที่ครบถ้วน ทั้งการสร้าง **Navbar, Dropdown, Drawer (Sidebar)** และการออกแบบ **Responsive Navbar** แบบ **Mobile-First** รวมถึงแนวทางการใช้ **Breadcrumbs** เพื่อให้ผู้ใช้เข้าใจตำแหน่งของตนเองบนเว็บไซต์อย่างชัดเจน ตัวอย่างแนวประยุกต์และ Integrated Examples จะช่วยให้ผู้อ่านสามารถสร้างระบบนำทางที่สมบูรณ์และมีเอกลักษณ์

บทที่ 8 **DaisyUI Utilities** เจาะลึกองค์ประกอบเสริมที่ช่วยยกระดับ User Experience ทั้ง **Typography (prose)** สำหรับจัดรูปแบบข้อความให้สวยงาม **Loading Spinner, Progress Bar, Skeleton** สำหรับสื่อสารสถานะการทำงาน และ **Tooltip** และ **Indicators** ที่ช่วยเพิ่มความชัดเจนในการโต้ตอบ ตัวอย่าง Ultimate Integrated Examples จะทำให้ผู้อ่านเห็นการใช้งานร่วมกันของ Utilities เหล่านี้ในโปรเจกต์จริง

บทที่ 9 **Theme Customization** นำเสนอวิธีการปรับแต่งธีมของเว็บไซต์และแอปพลิเคชันอย่างละเอียด ตั้งแต่การแก้ไขธีมพื้นฐานใน **tailwind.config.js** การสร้าง **Custom Theme** ของ DaisyUI ไปจนถึงการใช้ **CSS Variables** เพื่อให้การปรับสี ฟอนต์ และสไตล์ต่าง ๆ เป็นไปอย่างยืดหยุ่นและสะดวกต่อการจัดการ ตัวอย่างบูรณาการและ Ultimate Integrated Examples จะช่วยให้ผู้อ่านสามารถสร้างธีมที่มีเอกลักษณ์และนำไปใช้ซ้ำได้

บทที่ 10 **State & Interactivity** เป็นบทสุดท้ายของส่วนเนื้อหาหลัก ซึ่งเน้นการสร้างเว็บไซต์ที่ตอบสนองต่อการโต้ตอบของผู้ใช้ ผู้อ่านจะได้เรียนรู้การสร้าง **Modal toggle** ด้วย input/checkbox และ label, การใช้ **Collapse** และ **Accordion** เพื่อจัดการเนื้อหาซ้อนกัน และการสร้าง **Toast** และ **Alert** แบบ **interactive** ตัวอย่างบูรณาการและ Ultimate Integration Project จะช่วยให้ผู้อ่านเข้าใจการนำ State และองค์ประกอบ interactive มาประยุกต์ใช้อย่างครบวงจร

ตลอดทั้งเล่ม หนังสือเล่มนี้มุ่งเน้นการผสมผสานทฤษฎีและการปฏิบัติ เพื่อให้ผู้อ่านสามารถสร้างเว็บไซต์ที่ครบวงจร ตั้งแต่ Layout, Navigation, Utilities, Theme ไปจนถึง Interactivity ได้อย่างมั่นใจ พร้อมทั้งสามารถประยุกต์ใช้ในโปรเจกต์จริงได้อย่างมีประสิทธิภาพ ด้วยตัวอย่างและ Ultimate Examples ที่ครอบคลุมทุกหัวข้อ

หวังว่าหนังสือ **DaisyUI Web Programming: Intermediate** เล่มนี้จะเป็นคู่มือที่ช่วยพัฒนาทักษะและความเข้าใจของผู้อ่านในระดับกลางไปจนถึงขั้นสูง ทำให้สามารถสร้างเว็บไซต์ที่ทันสมัย สวยงาม และมีประสบการณ์ผู้ใช้ที่ยอดเยี่ยมได้อย่างครบถ้วน

ด้วยรักและปรารถนาดี  
ศูนย์หนังสือราคาหักเรียน

# สารบัญ

หน้า

บทที่ 6 Layout System.....	1
•Layout System	
•Layout System – รายละเอียดเชิงลึก	
•การใช้ Grid + Flex กับ DaisyUI	
•รายละเอียดเชิงลึก ของหัวข้อ Hero section, Stats, Carousel ใน DaisyUI	
•รายละเอียดเชิงลึก ของหัวข้อ Responsive Design ด้วย Tailwind + DaisyUI	
•ตัวอย่างบูรณาการ	
•Ultimate Example	
บทที่ 7 Navigation & Menus.....	63
•Navigation & Menus	
•Navigation & Menus – เชิงลึก	
•Navbar, Dropdown, Drawer (Sidebar) ของ DaisyUI	
•Responsive Navbar (Mobile-First) – เชิงลึก	
•ตัวอย่างโปรแกรมแนวประยุกต์	
•Breadcrumbs ใน DaisyUI	
•Integrated Examples (Breadcrumbs + Navbar + Drawer + Theme Switcher)	
•Ultimate Integrated Examples	
บทที่ 8 DaisyUI Utilities.....	114
•DaisyUI Utilities	
•DaisyUI Utilities – รายละเอียดเชิงลึก	
•Typography (prose) ของ DaisyUI	
•Loading Spinner, Progress Bar, Skeleton	
•ตัวอย่างบูรณาการ Ultimate	
•Tooltip และ Indicators	
•ตัวอย่างบูรณาการ	
•Ultimate Integrated Examples	

บทที่ 9 Theme Customization .....	157
● Theme Customization	
● Theme Customization – Detailed Deep Dive	
● การแก้ไข theme ใน tailwind.config.js	
● เพิ่ม Custom Theme ใน DaisyUI	
● ใช้ CSS Variables ของ DaisyUI	
● ตัวอย่างบูรณาการ	
● Ultimate Integrated Examples	
บทที่ 10 State & Interactivity .....	200
● State & Interactivity	
● State & Interactivity – รายละเอียดเชิงลึก	
● Modal toggle ด้วย input/checkbox + label	
● Collapse & Accordion – รายละเอียดเชิงลึก	
● Toast & Alert – รายละเอียดเชิงลึก	
● ตัวอย่างบูรณาการ	
● ตัวอย่าง Ultimate Integration Project	
บรรณานุกรม .....	247

## บทที่ 6

# Layout System

## (Layout System)

### เนื้อหา

- Layout System
- Layout System – รายละเอียดเชิงลึก
- การใช้ Grid + Flex กับ DaisyUI
- รายละเอียดเชิงลึก ของหัวข้อ Hero section, Stats, Carousel ใน DaisyUI
- รายละเอียดเชิงลึก ของหัวข้อ Responsive Design ด้วย Tailwind + DaisyUI
- ตัวอย่างบูรณาการ
- Ultimate Example

### บทที่ 6: Layout System

#### บทนำ

การออกแบบโครงสร้างของหน้าเว็บหรือแอปพลิเคชันถือเป็นรากฐานสำคัญของการพัฒนา หากเปรียบเทียบเว็บไซต์เป็นอาคาร Layout ก็เปรียบเสมือนผังสถาปัตยกรรมที่กำหนดตำแหน่งของแต่ละองค์ประกอบให้ชัดเจนและมีระเบียบ การจัดวางที่ดีช่วยให้ผู้ใช้งานสามารถเข้าถึงข้อมูลได้อย่างสะดวก เข้าใจลำดับของเนื้อหา และสร้างความประทับใจตั้งแต่แรกเห็น

ในยุคดิจิทัลที่ผู้ใช้งานเข้าถึงข้อมูลผ่านอุปกรณ์ที่หลากหลาย การออกแบบ Layout ไม่ได้เป็นเพียงเรื่องของความสวยงาม แต่ยังต้องรองรับการใช้งานบนหน้าจอที่มีขนาดและสัดส่วนแตกต่างกัน ไม่ว่าจะเป็นจอขนาดใหญ่ของคอมพิวเตอร์ แท็บเล็ต หรือสมาร์ทโฟน แนวคิดเรื่อง Responsive Design จึงเป็นหัวใจที่ทำให้การใช้งานเป็นไปอย่างราบรื่นในทุกบริบท

หนึ่งในเครื่องมือที่ช่วยให้นักพัฒนาจัดการกับ Layout ได้อย่างมีประสิทธิภาพคือ Tailwind CSS ซึ่งมีแนวคิดแบบ Utility-First ที่ให้ความยืดหยุ่นในการปรับแต่งโครงสร้างและสไตล์ได้อย่างละเอียด เมื่อใช้งานร่วมกับ DaisyUI ซึ่งเป็นชุดคอมโพเนนต์เสริมที่ออกแบบมาให้พร้อมใช้งาน นักพัฒนาจะสามารถสร้างหน้าเว็บที่สวยงามและทันสมัยได้โดยไม่ต้องเริ่มต้นจากศูนย์

พื้นฐานสำคัญของการสร้าง Layout อยู่ที่การประยุกต์ใช้ Grid และ Flexbox เครื่องมือทั้งสองนี้มีความสามารถเฉพาะที่ช่วยเสริมกัน Grid ทำให้การแบ่งแถวและคอลัมน์มีความชัดเจน ในขณะที่ Flexbox ช่วยในการจัดเรียงและกระจายองค์ประกอบภายในพื้นที่ย่อยอย่างยืดหยุ่น การผสมผสานเทคนิคทั้งสองเข้ากับ DaisyUI ช่วยให้เราสามารถออกแบบโครงสร้างที่ซับซ้อนและปรับใช้ได้กับสถานการณ์จริง

โครงสร้างของเว็บไซต์สมัยใหม่ยังประกอบด้วยส่วนสำคัญที่ช่วยเสริมประสบการณ์ผู้ใช้ เช่น Hero Section ที่ใช้สร้างความประทับใจแรกเห็นและสื่อสารข้อความหลัก Stats Section ที่นำเสนอข้อมูลเชิงตัวเลขหรือดัชนีสำคัญอย่างกระชับ และ Carousel ที่สร้างการนำเสนอแบบหมุนเวียน ทำให้หน้าเว็บมีความเคลื่อนไหวและน่าสนใจมากยิ่งขึ้น

ความสามารถในการปรับเปลี่ยนตามขนาดหน้าจอคืออีกหนึ่งองค์ประกอบที่ไม่ควรมองข้าม Layout ที่ดีต้องสามารถรักษาความสมดุลและความสวยงามได้แม้เมื่อย่อหรือขยายหน้าจอ ระบบ Breakpoint ของ Tailwind ช่วยให้นักพัฒนากำหนดพฤติกรรมขององค์ประกอบตามอุปกรณ์ได้อย่างแม่นยำ เมื่อผสมผสานกับคอมโพเนนต์ของ DaisyUI ก็จะทำให้การสร้าง Responsive Layout เป็นเรื่องง่ายขึ้นมาก

เนื้อหาในบทนี้จึงมุ่งเน้นการสร้างความเข้าใจที่ครบถ้วนทั้งในเชิงแนวคิดและการปฏิบัติ เพื่อให้ผู้อ่านสามารถออกแบบ Layout ที่สวยงาม แข็งแรง และยืดหยุ่น โดยใช้เครื่องมือที่เป็นมาตรฐานในปัจจุบัน เมื่อเรียนรู้จบบทนี้ ผู้อ่านจะสามารถสร้างหน้าเว็บที่ไม่เพียงดึงดูดสายตา แต่ยังตอบสนองการใช้งานได้จริงบนทุกแพลตฟอร์มและทุกอุปกรณ์

## Layout System

- การใช้ Grid + Flex กับ DaisyUI
- Hero section, Stats, Carousel
- Responsive design ด้วย Tailwind + DaisyUI

### บทที่ 6: Layout System

DaisyUI เป็น **UI component library** บน **Tailwind CSS** ที่ช่วยให้สร้าง layout และส่วนประกอบหน้าเว็บได้ง่ายขึ้น โดยยังคงยืดหยุ่นของ Tailwind CSS ไว้เต็มที่

#### 1. การใช้ Grid + Flex กับ DaisyUI

DaisyUI ใช้ **utility classes** ของ **Tailwind** สำหรับ layout เป็นหลัก เช่น flex, grid, justify-center, items-center, gap-4

แนวคิดการใช้งาน

- **Flex**
  - ใช้จัดเรียง component แบบแนวนอนหรือแนวตั้ง
  - Example: navbar, button group, card row

```
<div class="flex justify-between items-center p-4 bg-base-100">
```

```
<div class="logo">Logo</div>
```

```
<div class="flex gap-4">
```

```
<button class="btn btn-primary">Login</button>
```

```
<button class="btn btn-secondary">Signup</button>
```

```
</div>
```

```
</div>
```

- **Grid**

- ใช้จัด layout แบบตาราง
- Example: product list, card grid, features section

```
<div class="grid grid-cols-3 gap-4 p-4">
```

```
<div class="card bg-base-200 p-4">Card 1</div>
```

```
<div class="card bg-base-200 p-4">Card 2</div>
```

```
<div class="card bg-base-200 p-4">Card 3</div>
```

```
</div>
```

ข้อดี:

- สร้าง layout ได้เร็วโดยไม่ต้องเขียน CSS เพิ่ม
- Flex + Grid ใช้ร่วมกับ DaisyUI component ได้ง่าย

---

## 2. Hero section, Stats, Carousel

DaisyUI มี **component** สำเร็จรูปสำหรับ **layout** ที่นิยมใช้บนหน้าเว็บ

### Hero Section

- ใช้สำหรับ banner หน้าแรก หรือ section แนะนำ
- รองรับ responsive, text overlay, CTA button

```
<div class="hero min-h-screen bg-base-200">
```

```
<div class="hero-content text-center">
```

```
<div class="max-w-md">
```

```
<h1 class="text-5xl font-bold">Welcome to DaisyUI!</h1>
```

```
<p class="py-6">Build beautiful UI components faster using Tailwind + DaisyUI.</p>
```

```
<button class="btn btn-primary">Get Started</button>
```

```
</div>
```

```
</div>
```

```
</div>
```

### Stats

- ใช้แสดงตัวเลขสำคัญ หรือ KPI
- Example: จำนวนผู้ใช้งาน, รายได้, การเข้าชม

```
<div class="stats shadow w-full">
```

```
<div class="stat">
```

```
<div class="stat-title">Users</div>
```

```

    <div class="stat-value">1.2K</div>
  </div>
  <div class="stat">
    <div class="stat-title">Revenue</div>
    <div class="stat-value">$34K</div>
  </div>
  <div class="stat">
    <div class="stat-title">Feedback</div>
    <div class="stat-value">321</div>
  </div>
</div>

```

### Carousel

- ใช้สร้าง slider ภาพหรือ component
- Example: รีวิวลูกค้า, portfolio

```

<div class="carousel w-full rounded-box">
  <div id="slide1" class="carousel-item relative w-full">
    
    <div class="absolute flex justify-between transform -translate-y-1/2 left-5 right-5 top-1/2">
      <a href="#slide3" class="btn btn-circle">◀</a>
      <a href="#slide2" class="btn btn-circle">▶</a>
    </div>
  </div>
  <div id="slide2" class="carousel-item relative w-full">
    
    <div class="absolute flex justify-between transform -translate-y-1/2 left-5 right-5 top-1/2">
      <a href="#slide1" class="btn btn-circle">◀</a>
      <a href="#slide3" class="btn btn-circle">▶</a>
    </div>
  </div>
</div>

```

### 3. Responsive Design ด้วย Tailwind + DaisyUI

- Tailwind CSS ใช้ **breakpoint utilities** (sm:, md:, lg:, xl:)
- DaisyUI component รองรับ responsive โดยไม่ต้องปรับแต่ง CSS เพิ่ม

## ตัวอย่าง

```
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4 p-4">
  <div class="card p-4">Card 1</div>
  <div class="card p-4">Card 2</div>
  <div class="card p-4">Card 3</div>
</div>
```

- ผลลัพธ์:

- จอมือถือ: 1 column
- Tablet: 2 column
- Desktop: 3 column

## Hero section responsive

```
<div class="hero min-h-screen bg-base-200">
  <div class="hero-content flex-col lg:flex-row">
    
  </div>
  <h1 class="text-5xl font-bold">Responsive Hero</h1>
  <p class="py-6">Text และภาพจัดเรียงอัตโนมัติในมือถือและ desktop</p>
  <button class="btn btn-primary">Get Started</button>
</div>
</div>
</div>
```

## ข้อดี:

- ไม่ต้องเขียน media query เอง
- Component สลับ layout อัตโนมัติ
- ใช้ร่วมกับ Theme switcher ของ DaisyUI ได้เลย

## สรุปบทนี้:

- ใช้ **Grid + Flex** จัด layout หลัก
- ใช้ **Hero / Stats / Carousel** สำหรับ section สำเร็จรูป
- ใช้ **responsive utilities** ของ Tailwind + DaisyUI เพื่อรองรับมือถือ/desktop
- สามารถรวมกับ **Theme, Button, Card, Badge, Modal** ได้ทันที

## Layout System – รายละเอียดเชิงลึก

DaisyUI เป็น UI component library บน **Tailwind CSS** ซึ่งทำให้การสร้าง layout ของหน้าเว็บรวดเร็ว และเป็นระบบ โดยยังคง **ความยืดหยุ่น**ของ **Tailwind CSS** ไว้เต็มที่  
บทนี้ครอบคลุม **3 หัวข้อหลัก**

## 1. การใช้ Grid + Flex กับ DaisyUI

### 1.1 Flex

- DaisyUI ใช้ **Tailwind utility classes** (flex, justify-\*, items-\*, gap-\*)
- แนวทาง:
  - flex – กำหนดเป็น container แบบ flexbox
  - flex-col / flex-row – จัดเรียงแนวตั้ง / แนวนอน
  - justify-center / justify-between – การจัดวาง horizontal
  - items-center – การจัดวาง vertical

#### ตัวอย่างเชิงลึก

```
<div class="flex justify-between items-center p-4 bg-base-100 shadow-md">
  <div class="logo font-bold text-xl">MyBrand</div>
  <div class="flex gap-4">
    <button class="btn btn-primary">Login</button>
    <button class="btn btn-secondary">Signup</button>
  </div>
</div>
```

- ผลลัพธ์:
  - Logo อยู่ด้านซ้าย
  - Buttons จัดเรียงแนวนอนด้านขวา พร้อมช่องว่าง (gap-4)

### 1.2 Grid

- ใช้สำหรับ จัด **layout** แบบตาราง เช่น card gallery, feature section
- Tailwind class: grid, grid-cols-\*, gap-\*

#### ตัวอย่างเชิงลึก

```
<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6 p-4">
  <div class="card bg-base-200 p-4 shadow">Card 1</div>
  <div class="card bg-base-200 p-4 shadow">Card 2</div>
  <div class="card bg-base-200 p-4 shadow">Card 3</div>
</div>
```

- ผลลัพธ์ **responsive**:

- มือถือ: 1 column
- Tablet: 2 columns
- Desktop: 3 columns

**ข้อดี:**

- ใช้งานง่าย, เร็ว, ไม่ต้องเขียน CSS เอง
- ใช้ร่วมกับ DaisyUI component ได้ทุกชนิด

**2. Hero Section, Stats, Carousel****2.1 Hero Section**

- เป็น section หน้าแรกของเว็บ ใช้ **text + image + CTA**
- DaisyUI มี hero class ให้พร้อม

**ตัวอย่างเชิงลึก**

```
<div class="hero min-h-screen bg-base-200">
  <div class="hero-content flex-col lg:flex-row">
    
  </div>
  <h1 class="text-5xl font-bold">Welcome to DaisyUI</h1>
  <p class="py-6">สร้าง UI components แบบรวดเร็ว ด้วย Tailwind + DaisyUI</p>
  <button class="btn btn-primary">Get Started</button>
</div>
</div>
</div>
```

- **Responsive:**
  - มือถือ: image + text จัดแนวตั้ง
  - Desktop: image + text จัดแนวนอน (lg:flex-row)

**2.2 Stats**

- ใช้แสดงข้อมูลตัวเลขสำคัญ เช่น users, revenue
- DaisyUI มี class stats และ stat ให้พร้อม

**ตัวอย่างเชิงลึก**

```
<div class="stats shadow w-full">
  <div class="stat">
    <div class="stat-title">Users</div>
    <div class="stat-value">1.2K</div>
```

```

<div class="stat-desc">↗ □ 15%</div>
</div>
<div class="stat">
  <div class="stat-title">Revenue</div>
  <div class="stat-value">$34K</div>
  <div class="stat-desc">↘ □ 5%</div>
</div>
<div class="stat">
  <div class="stat-title">Feedback</div>
  <div class="stat-value">321</div>
  <div class="stat-desc">↗ □ 20%</div>
</div>
</div>

```

- ข้อดี: สวย, อ่านง่าย, responsive ทันที

### 2.3 Carousel

- ใช้สำหรับ ภาพสไลด์หรือ **component** สไลด์
- DaisyUI มี carousel class พร้อม carousel-item

#### ตัวอย่างเชิงลึก

```

<div class="carousel w-full rounded-box">
  <div id="slide1" class="carousel-item relative w-full">
    
    <div class="absolute flex justify-between transform -translate-y-1/2 left-5 right-5 top-1/2">
      <a href="#slide3" class="btn btn-circle">◀</a>
      <a href="#slide2" class="btn btn-circle">▶</a>
    </div>
  </div>
  <div id="slide2" class="carousel-item relative w-full">
    
  </div>
</div>

```

- ข้อดี:
  - ใช้ง่าย, รองรับ navigation buttons
  - สามารถวางร่วมกับ Badge, Card หรือ Hero ได้

---

### 3. Responsive Design ด้วย Tailwind + DaisyUI

#### 3.1 Breakpoints

- Tailwind ใช้ sm:, md:, lg:, xl:
- DaisyUI component รองรับ responsive อัตโนมัติ

#### ตัวอย่างเชิงลึก

```
<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-6 p-4">  
  <div class="card p-4 shadow">Card 1</div>  
  <div class="card p-4 shadow">Card 2</div>  
  <div class="card p-4 shadow">Card 3</div>  
</div>
```

- ผลลัพธ์:
  - มือถือ: 1 column
  - Tablet: 2 column
  - Desktop: 3 column

#### 3.2 Responsive Hero

```
<div class="hero min-h-screen bg-base-200">  
  <div class="hero-content flex-col lg:flex-row">  
      
    <div>  
      <h1 class="text-5xl font-bold">Responsive Hero</h1>  
      <p class="py-6">Text + image จัดเรียงอัตโนมัติ มือถือ/Desktop</p>  
      <button class="btn btn-primary">Get Started</button>  
    </div>  
  </div>  
</div>
```

- Flex จะสลับ layout ตาม lg:flex-row
- Image + Text จัดแนวตั้งบนมือถือ, แนวนอนบน desktop

---

### 4. Tips & Best Practices

- ใช้ **DaisyUI component** เป็น **building blocks**
- ใช้ **Flex/Grid + gap/padding/margin** เพื่อจัด spacing
- ใช้ **responsive utilities** ของ Tailwind เพื่อรองรับทุก device
- รวม **Hero + Stats + Carousel + Badge + Card** ให้เป็นหน้าเว็บหน้าเดียวได้เลย

สรุปเชิงลึก:

- DaisyUI + Tailwind ช่วยให้ layout ทำได้ เร็ว, **responsive**, สวยงาม
- Hero / Stats / Carousel / Flex / Grid สามารถ **combine** เป็นหน้าเว็บสำเร็จรูปได้ทันที
- รองรับ **responsive design** โดยไม่ต้องเขียน media query เอง

## การใช้ Grid + Flex กับ DaisyUI

DaisyUI เป็น UI component library บน **Tailwind CSS** ทำให้เราสามารถใช้ **utility classes** ของ **Tailwind** ร่วมกับ component สำเร็จรูปได้ทันที โดยเฉพาะ layout เช่น **Flex** และ **Grid**

### 1. Flex Layout

**Flex** ใช้สำหรับจัดเรียง component แบบแนวนอนหรือแนวตั้ง

Tailwind class หลักที่ใช้กับ DaisyUI:

Class	ความหมาย
flex	กำหนด container เป็น flexbox
flex-row	จัดเรียง children แนวนอน
flex-col	จัดเรียง children แนวตั้ง
justify-start/center/end/between	จัดแนวนอน
items-start/center/end/stretch	จัดแนวตั้ง
gap-*	ระยะห่างระหว่าง children

#### ตัวอย่าง Flex

```
<div class="flex justify-between items-center p-4 bg-base-100 shadow-md">
  <div class="logo font-bold text-xl">Brand</div>
  <div class="flex gap-4">
    <button class="btn btn-primary">Login</button>
    <button class="btn btn-secondary">Signup</button>
  </div>
</div>
```

#### คำอธิบาย

- flex – container จัดเรียงแบบ flex
- justify-between – Logo ซ้าย / Buttons ขวา
- items-center – align vertical ให้ตรงกลาง

- gap-4 – ช่องว่างระหว่างปุ่ม

ผลการรัน:

- Logo ซ้าย, ปุ่ม Login/Signup ขวา
- Responsive: buttons จะลดขนาดตาม container

## 2. Grid Layout

**Grid** ใช้สำหรับจัด layout แบบตาราง เช่น card gallery, feature section

Tailwind class หลัก:

Class	ความหมาย
grid	กำหนด container เป็น grid
grid-cols-1/2/3...	กำหนดจำนวน column
gap-*	ช่องว่างระหว่าง grid items
sm:, md:, lg:	responsive breakpoints

ตัวอย่าง **Grid**

```
<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6 p-4">
  <div class="card bg-base-200 p-4 shadow">Card 1</div>
  <div class="card bg-base-200 p-4 shadow">Card 2</div>
  <div class="card bg-base-200 p-4 shadow">Card 3</div>
</div>
```

คำอธิบาย

- grid – container เป็น grid
- grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 – responsive: มือถือ 1 column / Tablet 2 / Desktop 3
- gap-6 – spacing ระหว่าง grid items

ผลการรัน:

- Card จัดเรียงอัตโนมัติตามขนาดหน้าจอ
- Responsive layout รองรับมือถือ, tablet, desktop

## 3. การรวม Flex + Grid กับ DaisyUI Component

- สามารถใช้ **Flex/Grid** จัด layout ของ **Card, Button, Badge, Modal** ได้เลย
- DaisyUI component จะปรับตัวเข้ากับ Flex/Grid โดยไม่ต้องแก้ CSS เพิ่ม

ตัวอย่างรวม

```
<div class="grid grid-cols-1 md:grid-cols-3 gap-4 p-4">
```

```

<div class="card bg-base-200 p-4 flex flex-col items-center">
  <div class="text-xl font-bold">Card 1</div>
  <button class="btn btn-primary mt-2">Action</button>
</div>
<div class="card bg-base-200 p-4 flex flex-col items-center">
  <div class="text-xl font-bold">Card 2</div>
  <button class="btn btn-secondary mt-2">Action</button>
</div>
<div class="card bg-base-200 p-4 flex flex-col items-center">
  <div class="text-xl font-bold">Card 3</div>
  <button class="btn btn-accent mt-2">Action</button>
</div>
</div>

```

### คำอธิบาย

- Grid จัด column และ gap
- Flex ในแต่ละ card จัด vertical layout ของ content
- Component ของ DaisyUI (Card + Button) ทำงานร่วมกับ Flex/Grid ได้ทันที

### ข้อดีของการใช้ Flex + Grid กับ DaisyUI

1. สร้าง layout ได้รวดเร็วโดยไม่ต้องเขียน CSS เพิ่ม
2. รองรับ **responsive design** อัตโนมัติด้วย Tailwind breakpoints
3. สามารถรวมกับ component อื่น เช่น Hero, Stats, Carousel, Badge
4. มีความยืดหยุ่นสูง: ปรับ layout ได้ง่ายตามต้องการ

ตัวอย่างโปรแกรมพื้นฐานและแนวประยุกต์ของ DaisyUI ในหัวข้อ **Grid + Flex Layout** โดยจะให้เต็มไฟล์, โครงสร้าง, คำอธิบายโค้ด และผลการรัน

### ตัวอย่างโปรแกรมพื้นฐาน 3 ตัวอย่าง

#### ตัวอย่าง 1: Flex Navbar

ไฟล์: **index.html**

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Flex Navbar</title>
<script src="https://cdn.tailwindcss.com"></script>
<script>
  tailwind.config = { theme: { extend: {} }, plugins: [daisyui] }
</script>
<script src="https://cdn.jsdelivr.net/npm/daisyui@3.6.12/dist/full.js"></script>
</head>
<body class="bg-base-200 p-4">
  <div class="flex justify-between items-center p-4 bg-base-100 shadow-md">
    <div class="font-bold text-xl">MyBrand</div>
    <div class="flex gap-4">
      <button class="btn btn-primary">Login</button>
      <button class="btn btn-secondary">Signup</button>
    </div>
  </div>
</body>
</html>

```

### คำอธิบายโค้ด

- flex justify-between items-center จัด layout logo ซ้าย / buttons ขวา
- DaisyUI ใช้ class btn btn-primary / btn btn-secondary สำหรับปุ่ม

### ผลการรัน:

- Navbar logo ซ้าย, Login/Signup ขวา, responsive

---

## ตัวอย่าง 2: Grid Card Layout

### ไฟล์: index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grid Cards</title>
  <script src="https://cdn.tailwindcss.com"></script>

```

```

<script>
  tailwind.config = { plugins: [daisyui] }
</script>
<script src="https://cdn.jsdelivr.net/npm/daisyui@3.6.12/dist/full.js"></script>
</head>
<body class="bg-base-200 p-4">
  <div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
    <div class="card bg-base-100 p-4 shadow">Card 1</div>
    <div class="card bg-base-100 p-4 shadow">Card 2</div>
    <div class="card bg-base-100 p-4 shadow">Card 3</div>
  </div>
</body>
</html>

```

### คำอธิบายโค้ด

- grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 จัด column responsive
- DaisyUI card สร้าง card layout พร้อม shadow

### ผลการรัน:

- มือถือ 1 column, tablet 2, desktop 3

---

### ตัวอย่าง 3: Flex + Grid รวมกัน

#### ไฟล์: index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Flex + Grid</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script>
    tailwind.config = { plugins: [daisyui] }
  </script>
  <script src="https://cdn.jsdelivr.net/npm/daisyui@3.6.12/dist/full.js"></script>
</head>
<body class="bg-base-200 p-4">

```

```

<div class="grid grid-cols-1 md:grid-cols-2 gap-4">
  <div class="card p-4 flex flex-col items-center shadow">
    <div class="font-bold text-xl">Card 1</div>
    <button class="btn btn-primary mt-2">Action</button>
  </div>
  <div class="card p-4 flex flex-col items-center shadow">
    <div class="font-bold text-xl">Card 2</div>
    <button class="btn btn-secondary mt-2">Action</button>
  </div>
</div>
</body>
</html>

```

### คำอธิบายโค้ด

- Grid จัด 2 column (desktop)
- Flex ภายใน card จัด content vertical
- DaisyUI component ทำงานร่วมกันได้

### ผลการรัน:

- มือถือ 1 column, Desktop 2 column, card ข้างในจัดแนวตั้ง

## ตัวอย่างโปรแกรมแนวประยุกต์ 3 ตัวอย่าง

### ตัวอย่าง 1: Hero + CTA Layout

ไฟล์: `index.html`

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hero Page</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script>
    tailwind.config = { plugins: [daisyui] }
  </script>
  <script src="https://cdn.jsdelivr.net/npm/daisyui@3.6.12/dist/full.js"></script>

```

```

</head>
<body class="bg-base-200">
  <div class="hero min-h-screen bg-base-100">
    <div class="hero-content flex-col lg:flex-row">
      
      <div>
        <h1 class="text-5xl font-bold">Welcome to DaisyUI</h1>
        <p class="py-6">สร้าง UI layout รวดเร็ว ด้วย Tailwind + DaisyUI</p>
        <button class="btn btn-primary">Get Started</button>
      </div>
    </div>
  </div>
</body>
</html>

```

ผลการรัน:

- Hero responsive, image + text จัด layout อัตโนมัติ

## ตัวอย่าง 2: Stats + Grid Cards

ไฟล์: `index.html`

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Stats + Cards</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script>
    tailwind.config = { plugins: [daisyui] }
  </script>
  <script src="https://cdn.jsdelivr.net/npm/daisyui@3.6.12/dist/full.js"></script>
</head>
<body class="bg-base-200 p-4">
  <div class="stats shadow w-full mb-6">
    <div class="stat">

```

```

    <div class="stat-title">Users</div>
    <div class="stat-value">1.2K</div>
  </div>
  <div class="stat">
    <div class="stat-title">Revenue</div>
    <div class="stat-value">$34K</div>
  </div>
  <div class="stat">
    <div class="stat-title">Feedback</div>
    <div class="stat-value">321</div>
  </div>
</div>
<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
  <div class="card bg-base-100 p-4 shadow">Card A</div>
  <div class="card bg-base-100 p-4 shadow">Card B</div>
  <div class="card bg-base-100 p-4 shadow">Card C</div>
</div>
</body>
</html>

```

ผลการรัน:

- Stats แสดง KPI, grid cards แสดงข้อมูล product/feature

### ตัวอย่าง 3: Carousel + Badge Layout

ไฟล์: index.html

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Carousel + Badge</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script>
    tailwind.config = { plugins: [daisyui] }
  </script>

```

```

<script src="https://cdn.jsdelivr.net/npm/daisyui@3.6.12/dist/full.js"></script>
</head>
<body class="bg-base-200 p-4">
  <div class="carousel w-full rounded-box mb-6">
    <div id="slide1" class="carousel-item relative w-full">
      
    </div>
    <div id="slide2" class="carousel-item relative w-full">
      
    </div>
  </div>
  <div class="flex gap-4 justify-center">
    <span class="badge badge-primary">New</span>
    <span class="badge badge-secondary">Sale</span>
    <span class="badge badge-accent">Hot</span>
  </div>
</body>
</html>

```

ผลการรัน:

- Carousel แสดงภาพ slide
- Badges แสดง label สำหรับ highlight feature/product

---

### Ultimate Integration Example 1: Marketing Landing Page

ไฟล์: **index.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ultimate Landing Page</title>
  <script src="https://cdn.tailwindcss.com"></script>
  <script>
    tailwind.config = { plugins: [daisyui] }
  </script>

```

```
<script src="https://cdn.jsdelivr.net/npm/daisyui@3.6.12/dist/full.js"></script>
</head>
<body data-theme="light" class="bg-base-200">

<!-- Navbar -->
<div class="flex justify-between items-center p-4 bg-base-100 shadow-md">
  <div class="font-bold text-xl">Brand</div>
  <div class="flex gap-4">
    <button class="btn btn-primary">Login</button>
    <button class="btn btn-secondary">Signup</button>
  </div>
</div>

<!-- Hero Section -->
<div class="hero min-h-screen bg-base-100 mt-4">
  <div class="hero-content flex-col lg:flex-row">
    
    <div>
      <h1 class="text-5xl font-bold">Welcome to DaisyUI Ultimate</h1>
      <p class="py-6">รวมทุก layout และ component สำคัญของ DaisyUI ในหน้าเดียว</p>
      <button class="btn btn-primary">Get Started</button>
    </div>
  </div>
</div>

<!-- Stats Section -->
<div class="stats shadow w-full mt-8">
  <div class="stat">
    <div class="stat-title">Users</div>
    <div class="stat-value">2.5K</div>
    <div class="stat-desc">↗  20%</div>
  </div>
  <div class="stat">
    <div class="stat-title">Revenue</div>

```

```
<div class="stat-value">$50K</div>
<div class="stat-desc">↗ □ 15%</div>
</div>
<div class="stat">
  <div class="stat-title">Feedback</div>
  <div class="stat-value">450</div>
  <div class="stat-desc">↘ □ 5%</div>
</div>
</div>

<!-- Cards Section -->
<div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6 mt-8 p-4">
  <div class="card bg-base-100 p-4 flex flex-col items-center shadow">
    <div class="font-bold text-xl">Feature 1</div>
    <p class="py-2">คำอธิบาย feature 1</p>
    <button class="btn btn-primary mt-2">Explore</button>
  </div>
  <div class="card bg-base-100 p-4 flex flex-col items-center shadow">
    <div class="font-bold text-xl">Feature 2</div>
    <p class="py-2">คำอธิบาย feature 2</p>
    <button class="btn btn-secondary mt-2">Explore</button>
  </div>
  <div class="card bg-base-100 p-4 flex flex-col items-center shadow">
    <div class="font-bold text-xl">Feature 3</div>
    <p class="py-2">คำอธิบาย feature 3</p>
    <button class="btn btn-accent mt-2">Explore</button>
  </div>
</div>

<!-- Carousel Section -->
<div class="carousel w-full rounded-box mt-8">
  <div id="slide1" class="carousel-item relative w-full">
    
    <div class="absolute flex justify-between transform -translate-y-1/2 left-5 right-5 top-1/2">
```