

```
div class="flex flex-col md:flex-row gap-6">
  <!-- Sidebar -->
  <div class="md:w-1/4 bg-white dark:bg-gray-900 p-6 rounded-2xl shadow-xl sticky top-20 h-fit">
    Sidebar content
  </div>

  <!-- Main Content -->
  <div class="flex-1 bg-gray-50 dark:bg-gray-800 p-6 rounded-2xl shadow-xl">
    Main content
  </div>
</div>

◆ Parent Classes
1 flex
Kya karta hai: Parent ko flex container banata hai.
```

UI: Children elements ek line me arrange hote hain (default row)

2 flex-col
Kya karta hai: Flex direction column set karta hai

UI: Children vertically stack honge (mobile view)

3 md:flex-row
Kya karta hai: Medium screens ($\geq 768\text{px}$) par flex direction row

UI: Desktop/tablet → Sidebar + main content side-by-side

4 gap-6
Kya karta hai: Children ke beech 1.5rem (24px) gap

UI: Sidebar aur main content ek dusre se door rahenge

◆ Sidebar Classes
1 md:w-1/4
Medium+ screens → width = 25%

Mobile → default full width

2 bg-white dark:bg-gray-900
Background white light mode, dark gray dark mode

3 p-6
Padding = 1.5rem

Content andar se gap lega

4 rounded-2xl

Corners bahut gol

Soft, modern card look

5 shadow-xl

Strong shadow

Card upar float ho raha lagta hai

6 sticky top-20

Scroll ke saath sidebar 20 units (5rem) top se chipka rahe

Perfect for long pages

7 h-fit

Height content ke hisaab se automatically adjust

Full screen nahi le raha, sirf jitna content hai

◆ Visual UI Example

Mobile (<768px)

markdown

Copy code

[Sidebar (full width)]

| Sidebar content |

[Main Content (full width)]

| Main content |

Vertical stack (flex-col)

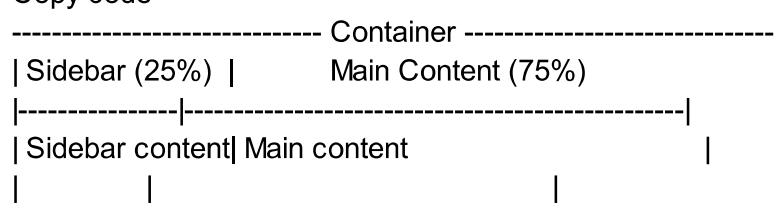
Gap 24px (gap-6)

Sidebar full width

Desktop ($\geq 768\text{px}$)

markdown

Copy code



Horizontal layout (flex-row)

Sidebar fixed width 25%

Gap 24px between sidebar and main content

Sticky sidebar top 5rem → scroll ke saath chipka rahega

```
div class="flex justify-between mb-6 items-center">
  <h2 class="text-3xl font-extrabold flex items-center gap-2 text-gray-800">
    All Products <span class="text-xl"></span>
  </h2>
</div>

<div class="flex flex-col md:flex-row gap-6">
  <!-- SIDEBAR -->
  <div class="md:w-1/4 bg-white dark:bg-gray-900 p-6 rounded-2xl shadow-xl sticky top-20 h-fit">
    <h3 class="font-bold mb-4 text-lg text-gray-800">Filters</h3>
    </div>
  </div>
  ◆ Part 1: Header Container
  html
  Copy code
<div class="flex justify-between mb-6 items-center">
  Classes & Logic:
  flex → Flex container banaya → children horizontal line me arrange honge
  justify-between → First child left, last child right → space evenly distribute
  items-center → Children vertically center align
  mb-6 → Bottom margin 1.5rem → spacing header ke neeche

```

UI Visual:

markdown

Copy code

All Products [Icon/Buttons]
(vertical center aligned)

Header me left me title

Right side me optional buttons/icons

Clean spacing, aligned center

◆ Part 2: Header Title

html

Copy code

```
<h2 class="text-3xl font-extrabold flex items-center gap-2 text-gray-800">  
  All Products <span class="text-xl"></span>  
</h2>
```

Classes & Logic:

text-3xl → Large heading size

font-extrabold → Bold heading

flex items-center gap-2 → Heading ke andar icons/labels ho to horizontally aligned + vertical center, gap 0.5rem

text-gray-800 → Dark gray color for readability

UI Visual:

css

Copy code

[All Products ] ← example icon in span
Text aur icon aligned, gap maintain

Professional clean look

◆ Part 3: Flex Container for Sidebar + Content

html

Copy code

```
<div class="flex flex-col md:flex-row gap-6">
```

Classes & Logic:

flex flex-col → Mobile: vertical stacking

md:flex-row → Desktop: horizontal layout

gap-6 → 1.5rem gap between sidebar and content

UI Visual (Mobile vs Desktop):

Mobile (<768px)

css

Copy code

[Sidebar]

[Content]

Desktop (≥768px)

css

Copy code

[Sidebar] [Content]

25% 75%

Responsive layout

Spacing clean (gap-6)

◆ Part 4: Sidebar

html

Copy code

```
<div class="md:w-1/4 bg-white dark:bg-gray-900 p-6 rounded-2xl shadow-xl sticky top-20 h-fit">
```

```
    <h3 class="font-bold mb-4 text-lg text-gray-800">Filters</h3>
```

```
</div>
```

Classes & Logic:

Class Logic / Effect

md:w-1/4 Desktop: 25% width

bg-white dark:bg-gray-900 Light/dark mode background

p-6 Padding inside sidebar

rounded-2xl Rounded corners

shadow-xl Strong shadow, card look

sticky top-20 Scroll ke saath 5rem top se chipka rahe

h-fit Height content ke hisaab se

Sidebar Heading:

font-bold mb-4 text-lg text-gray-800 → Bold, readable heading, spacing below

UI Visual:

Desktop:

markdown

Copy code

```
[ Sidebar Card ]
```

Filters

Option1

Option2

Option3

Mobile:

css

Copy code

```
[ Sidebar Card ]
```

Filters

Option1

Option2

Option3

```
[ Main Content Below ]
```

✓ Summary (Logical Flow)

Header → flex justify-between items-center → left-right alignment + vertical center

Header title → flex items-center gap-2 → text + optional icon aligned

Content container → flex flex-col md:flex-row gap-6 → responsive layout

Sidebar → styled card with sticky + shadow + padding + rounded corners

ep 1: md:w-1/4

Logic:

Medium screens ($\geq 768\text{px}$) par width = 25% of parent container

- Mobile (<768px) → default full width
- Desktop → sidebar 25%, main content 75%

UI Visual:

Mobile:

```
[ Sidebar (full width) ]
```

Desktop:

```
[ Sidebar (25%) ] [ Main Content (75%) ]
```

◆ Step 2: bg-white dark:bg-gray-900

Logic:

- Light mode → white background
- Dark mode → dark gray background

CSS logic (Tailwind uses `dark:` variant):

```
background-color: white;
```

```
@media (prefers-color-scheme: dark) {  
    background-color: #1f2937; /* gray-900 */  
}
```

UI Visual:

- Card looks bright in light theme
 - Card looks dark in dark theme
-

◆ Step 3: p-6

Logic:

Padding = 1.5rem = 24px

- Sidebar ke andar ka content border ke touch se 24px gap me dikhe

UI Visual:

```
-----  
| padding space |  
| Filters       |  
| Option 1      |  
| Option 2      |  
-----
```

◆ Step 4: rounded-2x1

Logic:

Corners ko bahut smooth rounded karo
CSS:

```
border-radius: 1rem; /* 16px */
```

UI Visual:

- Sharp rectangle → soft, modern card
 - Rounded corners noticeable especially shadow ke saath
-

◆ Step 5: **shadow-xl**

Logic:

Strong shadow effect card ke neeche
CSS (Tailwind):

```
box-shadow: 0 20px 25px rgba(0,0,0,0.1);
```

UI Visual:

[Sidebar Card floating above background]
subtle shadow below

- Card thoda “elevated” lagta hai
 - Modern UI feel
-

◆ Step 6: **sticky top-20**

Logic:

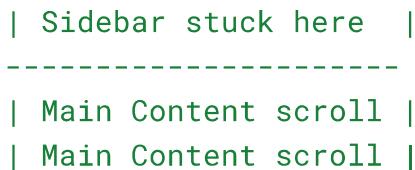
Sidebar **scroll ke saath chipka rahe**

- **sticky** → position: sticky
- **top-20** → 5rem (80px) top offset

UI Visual:

- User scroll kare → sidebar **top se 80px gap ke saath fixed**
- Content scroll hota hai, sidebar scroll ke saath upar nahi jaata

Scroll ↓



◆ Step 7: **h-fit**

Logic:

Sidebar ki height **content ke hisaab se** adjust ho

- Agar content chhota → sidebar chhota
- Agar content bada → sidebar bada
- Full viewport height automatically nahi le raha

UI Visual:

Sidebar height = jitna content hai

◆ Step 8: Sidebar Heading

```
<h3 class="font-bold mb-4 text-lg text-gray-800">Filters</h3>
```

- **font-bold** → bold heading
- **mb-4** → bottom margin 1rem → spacing from options
- **text-lg** → 1.125rem → readable heading

- `text-gray-800` → dark gray color

UI:

`Filters`

[Option 1]
[Option 2]

◆ Full Sidebar UI (Desktop Example)

Sidebar Card (25% width)
Filters
Option 1
Option 2
Option 3

Shadow visible, corners rounded
Sticky top 80px while scrolling

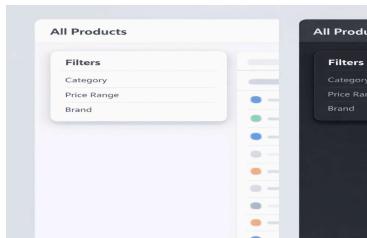
Mobile:

Sidebar Card (full width)
Filters
Option 1
Option 2

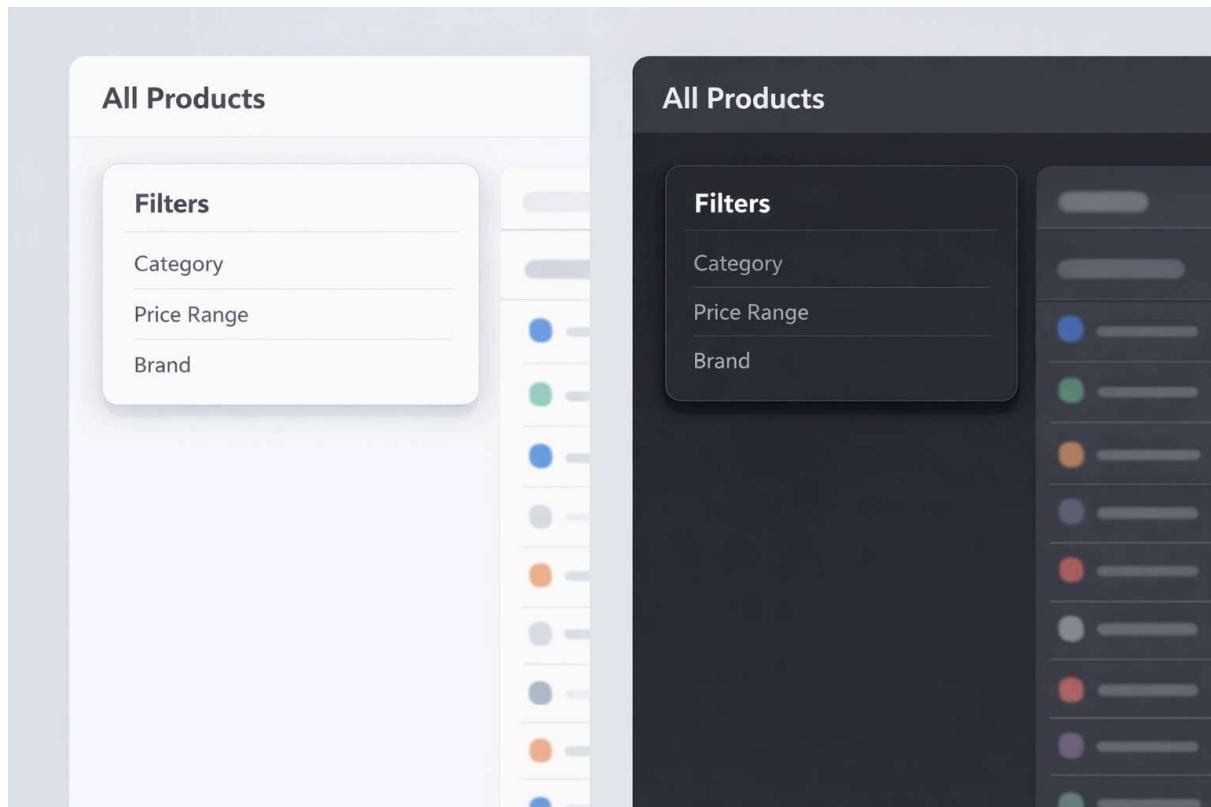
- Stacks vertically
 - Responsive layout
-

✓ Summary

Class	Purpose
md:w-1/4	Desktop width 25%, mobile full width
bg-white dark:bg-gray-900	Light/dark theme background
p-6	Padding inside card
rounded-2xl	Smooth rounded corners
shadow-xl	Elevated card look
sticky top-20	Sidebar top ke saath chipka rahe scroll me
h-fit	Height adjusts according to content



md:w-3/4



Logic:

Medium screens ($\geq 768\text{px}$) par width = 75% of parent container

- Mobile ($< 768\text{px}$) → default full width
- Desktop → content area = 75%, sidebar = 25%

UI Visual:

Mobile (<768px):

```
[ Product Area (full width) ]
```

Desktop ($\geq 768\text{px}$):

```
[ Sidebar 25% ] [ Product Area 75% ]
```

◆ Step 2: grid

Logic:

Container ko **CSS Grid** banata hai

- Children automatically grid items ban jaate hain

UI Visual:

- Flexible layout for cards
 - Rows aur columns me neatly arrange
-

◆ Step 3: grid-cols-1

sm:grid-cols-2 lg:grid-cols-3

Logic:

Class	Screen Size	Columns
grid-cols-1	default	1 column
sm:grid-col-s-2	≥640px	2 columns
lg:grid-col-s-3	≥1024px	3 columns

UI Visual:

Mobile (<640px):

[Card1]
[Card2]
[Card3]

Tablet ($\geq 640px$):

[Card1] [Card2]
[Card3] [Card4]

Desktop ($\geq 1024px$):

[Card1] [Card2] [Card3]
[Card4] [Card5] [Card6]

- Responsive layout automatically
 - Cards uniform spacing + alignment
-

◆ Step 4: gap-6

Logic:

Grid items ke beech **gap = 1.5rem (24px)**

- Horizontal + vertical dono me spacing

UI Visual:

[Card1] [Card2] [Card3]
24px 24px 24px
[Card4] [Card5] [Card6]

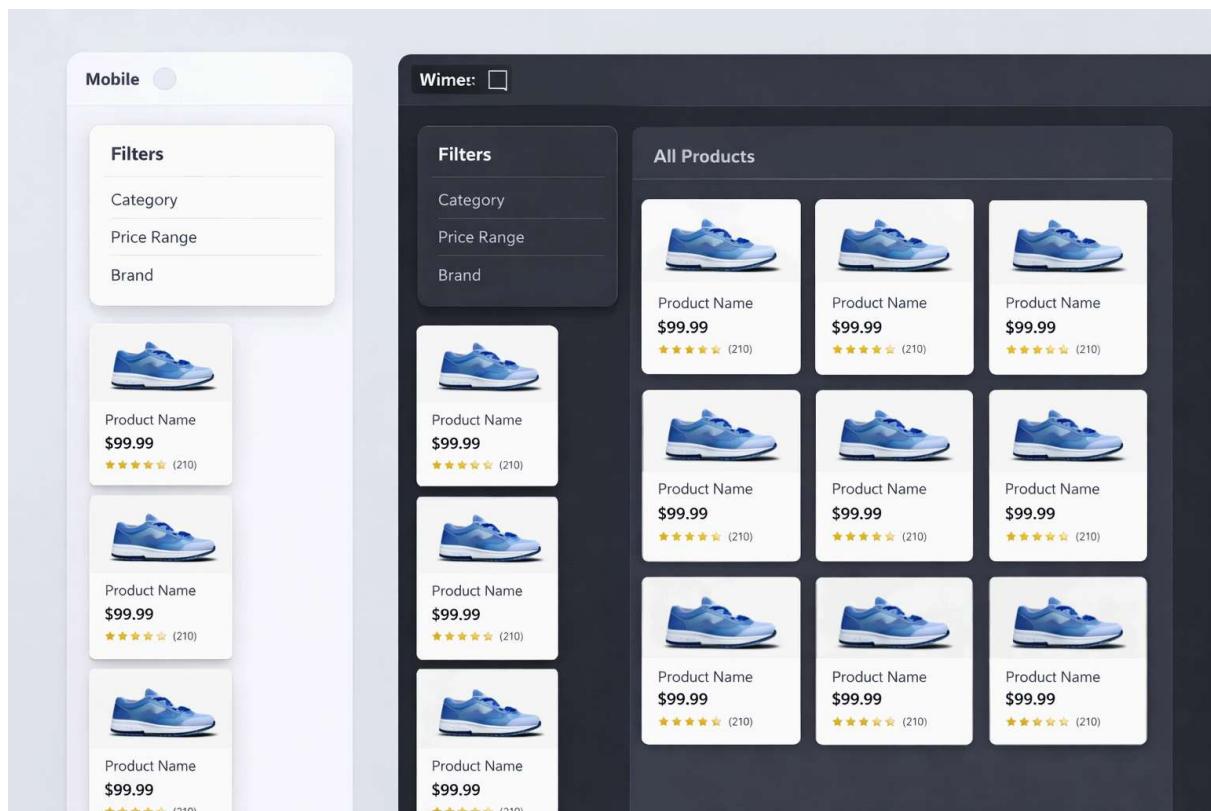
- Cards ek dusre se chipke nahi rahenge
 - Neat, modern grid layout
-

◆ Summary

Class	Purpose
md:w-3/4	Desktop: content area 75% width
grid	Create CSS grid container
grid-cols-1 sm:grid-cols-2 lg:grid-cols-3	Responsive column layout
gap-6	1.5rem spacing between cards

✓ Result:

- Mobile: single column stacked
- Tablet: 2 columns
- Desktop: 3 columns
- Spacing consistent
- Perfect for **product listing/grid layout** in admin or e-commerce panel



transition-shadow

Logic:

Box shadow changes ko smooth animate karo

CSS:

```
transition-property: box-shadow;
```

UI Visual:

- Hover ya active par shadow abruptly change nahi hota
- Smooth fade effect aata hai

Example:

Normal → subtle shadow

Hover → stronger shadow (smoothly)

◆ 2 duration-300

Logic:

Transition duration **300ms**

CSS:

```
transition-duration: 300ms;
```

UI Visual:

- Shadow change 0.3 seconds me smoothly animate hota hai
 - Quick but noticeable
-

◆ **3 flex flex-col**

Logic:

Flex container banaya aur **vertical stacking** set kiya

- Children ek ke neeche ek stack honge

UI Visual:

```
[ Image ]  
[ Product Name ]  
[ Price ]
```

◆ **4 cursor-pointer**

Logic:

Mouse hover par cursor **pointer** ban jaaye

CSS:

```
cursor: pointer;
```

UI Visual:

- User ko clearly indicate karta hai ki **clickable element** hai
-

◆ **5 bg-white**

Logic:

Background color **white**

- Card light mode me white background ke saath clean dikhe
- Dark mode me optionally dark background ke saath switch kiya ja sакта hai

UI Visual:



White background, smooth shadow on hover

◆ Combined Effect

- **Vertical card** (image, title, price)
 - **Clickable** with pointer cursor
 - **Hover effect:** Shadow smoothly increase (transition-shadow + duration-300)
 - Clean **white card** for product listing
-

Agar chaho, main iska **realistic product card mockup** bana sакта hoon jisme **shadow hover effect visible + vertical layout + cursor pointer** sab dikhe 😊

Kya main wo bana doon?

w-full h-full

Logic:

Element ki **width aur height parent container ke 100% set karo**

CSS:

```
width: 100%;  
height: 100%;
```

UI Visual:

- Image parent card ya container ke andar **poori space cover kare**
 - No gaps inside card
-

◆ 2 object-cover

Logic:

Image container ke size ke hisaab se **cropping + cover**

CSS:

```
object-fit: cover;
```

UI Visual:

- Image stretch ya distort nahi hoti
- Container ka poora area cover ho jata hai
- Centered + nicely cropped edges

Example:

```
[ Container ]  
+-----+  
| Image covers |  
| entire area |  
+-----+
```

◆ **3 transform**

Logic:

Element par **transform effects** allow karo

- Required for scale, rotate, translate, etc.
 - Without this, **scale-105** won't animate
-

◆ **4 group-hover:scale-105**

Logic:

Parent element (with **group**) hover hone par **image 105% scale ho jaye**

CSS:

```
transform: scale(1.05);
```

UI Visual:

- Hover par image thoda zoom in
- Card hover effect smooth + interactive

Normal: [IMAGE]
Hover: [IMAGE slightly zoomed]

◆ **5 transition-transform**

Logic:

Transform changes (like scale) **smoothly animate karo**

CSS:

```
transition-property: transform;
```

UI Visual:

- Scale effect sudden nahi hoga
 - Smooth zoom animation
-

◆ **duration-300**

Logic:

Transition duration **300ms**

CSS:

```
transition-duration: 300ms;
```

UI Visual:

- Zoom effect 0.3 seconds me smoothly complete
 - Quick but noticeable
-

◆ **Combined Effect (Visual UI)**

- Image **fully covers card** (`w-full h-full object-cover`)
- Hover par **slightly zoom in** (`scale-105`)
- Smooth animation (`transition-transform duration-300`)

```
+-----+  
| IMAGE (normal) |  
+-----+
```

Hover →

```
+-----+  
| IMAGE zoomed 105% |  
+-----+
```

- Modern e-commerce feel
- Interactive, user-friendly product card

line-clamp-2

Logic:

Text ko maximum 2 lines tak limit karo
Overflow hone par “...” ellipsis show kare

CSS (Tailwind uses plugin):

```
display: -webkit-box;  
-webkit-line-clamp: 2;  
-webkit-box-orient: vertical;  
overflow: hidden;
```

UI Visual:

Long text example:

Some very long product description that goes on...
→ Displayed as:
Some very long product
description...

- Only **2 lines visible**
 - Rest truncated with ellipsis
 - Perfect for product titles, card descriptions
-

◆ **break-words**

Logic:

Long words/text ko wrap kar do

CSS:

```
overflow-wrap: break-word;  
word-break: break-word;
```

UI Visual:

- Agar word bahut lamba ho (like `SuperLongProductNameWithoutSpaces`)
- Wo container ke andar **wrap hoke next line me aa jaye**

`SuperLongProductName
WithoutSpaces`

- Card layout break nahi hogा
 - Responsive and clean
-

◆ **Combined Effect**

- Text **maximum 2 lines** (`line-clamp-2`)

- Long words **wrap ho jaye** (**break-words**)

UI Visual:

```
+-----+  
| Super Long Product Name |  
| WithoutSpaces... |  
+-----+
```

transition-all

Logic:

Har CSS property change smoothly animate kare

CSS:

```
transition: all 0.3s ease;
```

UI Visual:

- Hover effects ya background change abrupt nahi hote
 - Smooth animation, professional feel
-

◆ Step 2: duration-300

Logic:

Transition duration = **300ms**

UI:

- Hover, gradient, color changes 0.3 seconds me smooth
-

◆ Step 3: `text-white`

Logic:

Text color `white`

- Works with colored background
 - Readable on blue/indigo gradient
-

◆ Step 4: `font-semibold`

Logic:

Text weight `600` → `semi-bold`

- Button text slightly bold for emphasis
-

◆ Step 5: `text-lg`

Logic:

Text size `large (~1.125rem)`

- Clear, visible on buttons/cards
-

◆ Step 6: Conditional Classes `(p.inStock ? ... : ...)`

 If `inStock = true`

```
bg-gradient-to-r from-blue-500 to-indigo-500  
hover:from-indigo-500 hover:to-blue-500
```

- **Gradient background:** left → blue-500, right → indigo-500
- **Hover effect:** gradient colors swap → interactive feel
- Smooth due to `transition-all duration-300`

UI Visual:

```
[ BUY BUTTON ]  
Normal: Blue → Indigo gradient  
Hover: Indigo → Blue gradient  
Text: White, semi-bold, large
```

✖ If `inStock = false`

```
bg-gray-400 cursor-not-allowed text-gray-700
```

- Button `gray`, disabled feel
- Cursor: not-allowed → shows ✖ on hover
- Text: gray-700 → subtle contrast

UI Visual:

```
[ OUT OF STOCK BUTTON ]  
Gray background, gray text  
Cursor: not allowed  
No hover effect
```



Combined Effect

Condition	Background	Text	Cursor	Hover	Transition
-----------	------------	------	--------	-------	------------

In Stock	Gradient Blue → Indigo	White	Pointer	Gradient swap	Smooth 300ms
Out of Stock	Gray-400	Gray-70	Not allowed	None	Smooth (no effect)

d:w-1/3

- Medium screens ($\geq 768\text{px}$) par **width = 33%**
- Sidebar / card ke horizontal layout me use
- Desktop: 1 row me 3 images aligned

2 w-full

- Default / mobile: **full width**
- Mobile stacked layout me 100% width

3 h-40

- Height = 10rem (160px)
- Fixed height on mobile / small screens

4 sm:h-auto

- Small screens ($\geq 640\text{px}$) par **height automatic**
- Image natural height ke hisaab se adjust ho jaye

5 overflow-hidden

- Container ke bahar ka content **clip ho jaye**
- Image container se bahar na jaaye

6 bg-gray-100

- Light gray background
 - Image load hone se pehle placeholder effect
 - Clean, neutral background
-

◆ Child `` Classes

1 `w-full h-full`

- Image **parent container ke size ko cover kare**
- Horizontal & vertical full coverage

2 `object-contain`

- Image **distortion avoid kare**
- Maintain aspect ratio
- Fit inside container

3 `p-4`

- Padding = 1rem (16px)
 - Image edges aur container border se gap
-

◆ UI Visual Example

Mobile (<768px)

```
+-----+  
| [ IMAGE ] | <- full width, fixed height 160px  
+-----+
```

Tablet/Desktop ($\geq 768px$)

```
+-----+-----+-----+
| IMAGE1 | IMAGE2 | IMAGE3 | <- each 1/3 width, auto height
+-----+-----+-----+
```

- Images horizontally aligned
 - Aspect ratio preserved (`object-contain`)
 - Padding ensures edges nahi chipakte
 - `bg-gray-100` acts as clean placeholder
-

Summary

Class	Purpose
<code>md:w-1/3</code>	Desktop: 1/3 width
<code>w-full</code>	Mobile: full width
<code>h-40</code>	Fixed height on small screens
<code>sm:h-auto</code>	Auto height on small+ screens
<code>overflow-hidden</code>	Clip overflowing content
<code>bg-gray-100</code>	Placeholder background
<code>w-full</code>	Image covers container
<code>h-full</code>	
<code>object-contain</code>	Maintain image aspect ratio
<code>p-4</code>	Padding inside container

opacity-50

Logic:

Element ki **transparency = 50%**

CSS:

```
opacity: 0.5;
```

UI Visual:

- Element **semi-transparent / faded** lagta hai
- Users ko signal milta hai ki ye **active nahi hai**

[Out of Stock] <- faded

- Perfect for disabled buttons or cards
-



2

cursor-not-allowed

Logic:

Hover par **cursor = not-allowed**

CSS:

```
cursor: not-allowed;
```

UI Visual:

- Cursor me **✗ / forbidden icon** dikhai deta hai
 - User clearly samajh jata hai ki element click nahi ho sakta
-

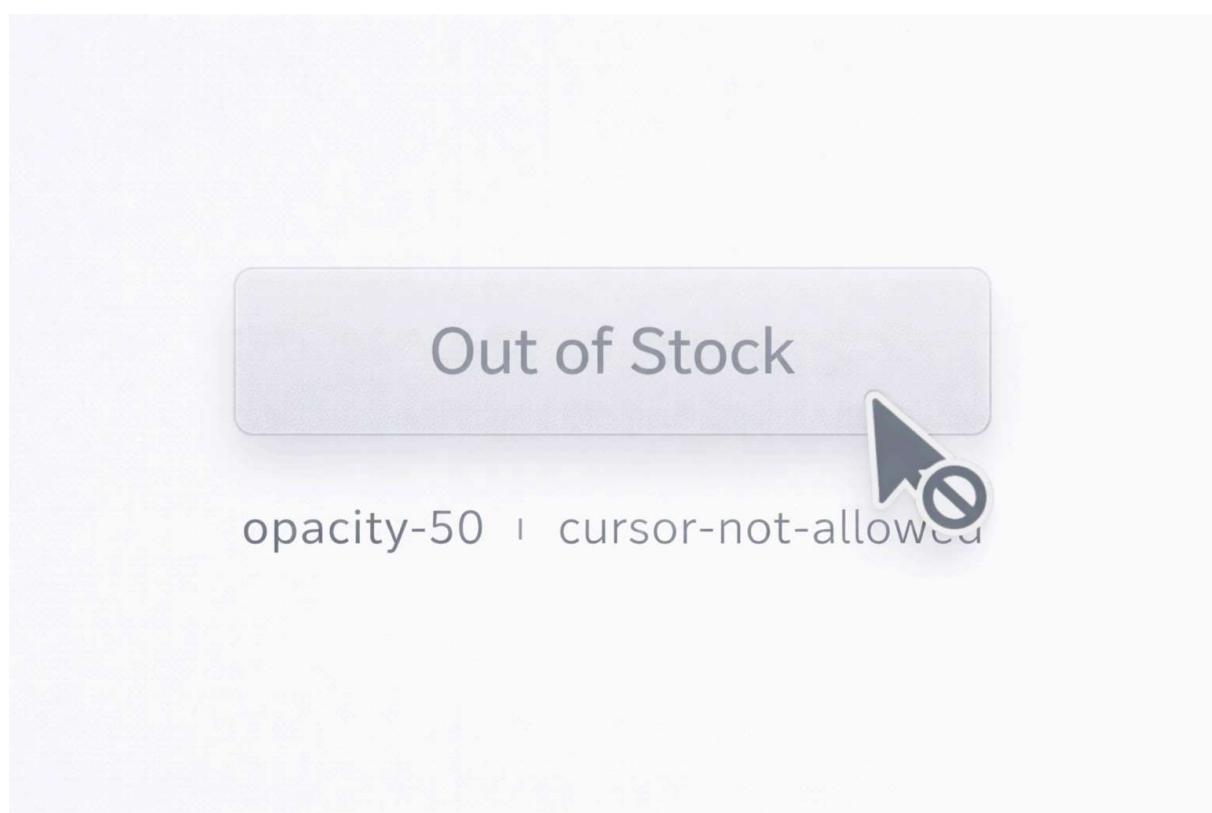
◆ Combined Effect

- **Opacity-50** → button ya element visually disabled lage
- **Cursor-not-allowed** → hover par cursor change ho → user ko feedback
- Often used with **disabled buttons / out-of-stock products**

UI Visual Example:

[Out of Stock Button]
opacity-50 → faded
cursor → not-allowed on hover

- Modern UI convention
- Clear, intuitive disabled state



flex-shrink-0

Logic:

Element **apni width shrink nahi kare** agar parent me space kam ho

CSS:

```
flex-shrink: 0;
```

- Useful: fixed width cards / sidebar
- Parent me flex items overlap ya compress nahi honge

UI Visual:

Parent flex container



- Even agar container small ho → Card 1 width **33% fixed**, shrink nahi hoti

◆ Combined Effect

Class	Purpose	Visual Outcome
lg:w-1/3	Desktop width = 33%	1 row me 3 cards aligned
flex-shrink-0	Prevent shrinking	Width fixed, responsive layout stable

◆ Use Case

- **Sidebar** in responsive layouts
 - **Product card** in horizontal scrolling carousels
 - **Dashboard widget** fixed width in flex container
-

Agar chaho, main iska **realistic visual mockup** bana sакта hoon jisme **desktop: 3 cards in a row with fixed width and no shrink** visible ho 😊

Kya main wo bana doon?

ticky top-20

Ah! 😊 Aap pooch rahe ho **sticky top-20** ka Tailwind logic aur visual effect. Chaliye step by step samjhte hain:

Code Snippet Example

```
<div class="sticky top-20 bg-white p-4 shadow">  
  Sidebar / Sticky Card  
</div>
```

◆ 1 **sticky**

Logic:

Element ko **scroll ke time parent ke boundary ke andar “stick” karne ke liye**

- CSS equivalent:

```
position: sticky;
```

- Element apni original position pe rehta hai **jab tak parent scroll hota hai**, phir stick ho jata hai top se offset ke saath

◆ 2 top-20

Logic:

Stick hone ke **top offset = 5rem (80px)**

- Tailwind spacing scale: **20 → 5rem**
- Element top se 80px gap me stick hoga

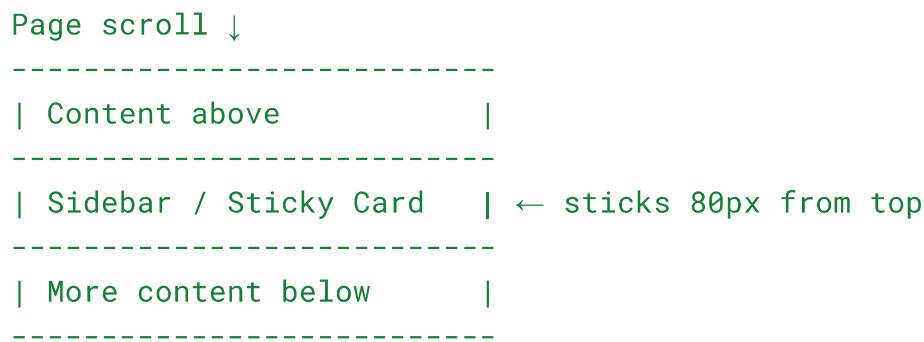
CSS equivalent:

```
top: 5rem;
```

◆ ✓ Visual UI

Scroll Behavior Example

Parent container scrollable height:



- Jab page scroll kare → card top se 80px gap me **chipka rahega**
 - Perfect for **filters, navigation, sticky buttons, or sidebars**
-

Use Case

- **Sidebar filters** in e-commerce
- **Sticky navbar / header**
- **Call-to-action buttons**

Mobile/Desktop responsive:

- Mobile → often full width sticky optional
- Desktop → sidebar sticky while scrolling conte