

## Notes For VR Studio Project:

We can target placeholder of form:

```
<div class="group">
  <a href="#">Vector</a>
  <div
    class="mx-2 mt-2 duration-500 border-b-2 opacity-0
    border-black group-hover:opacity-100"
  ></div>
```

Group Hover:

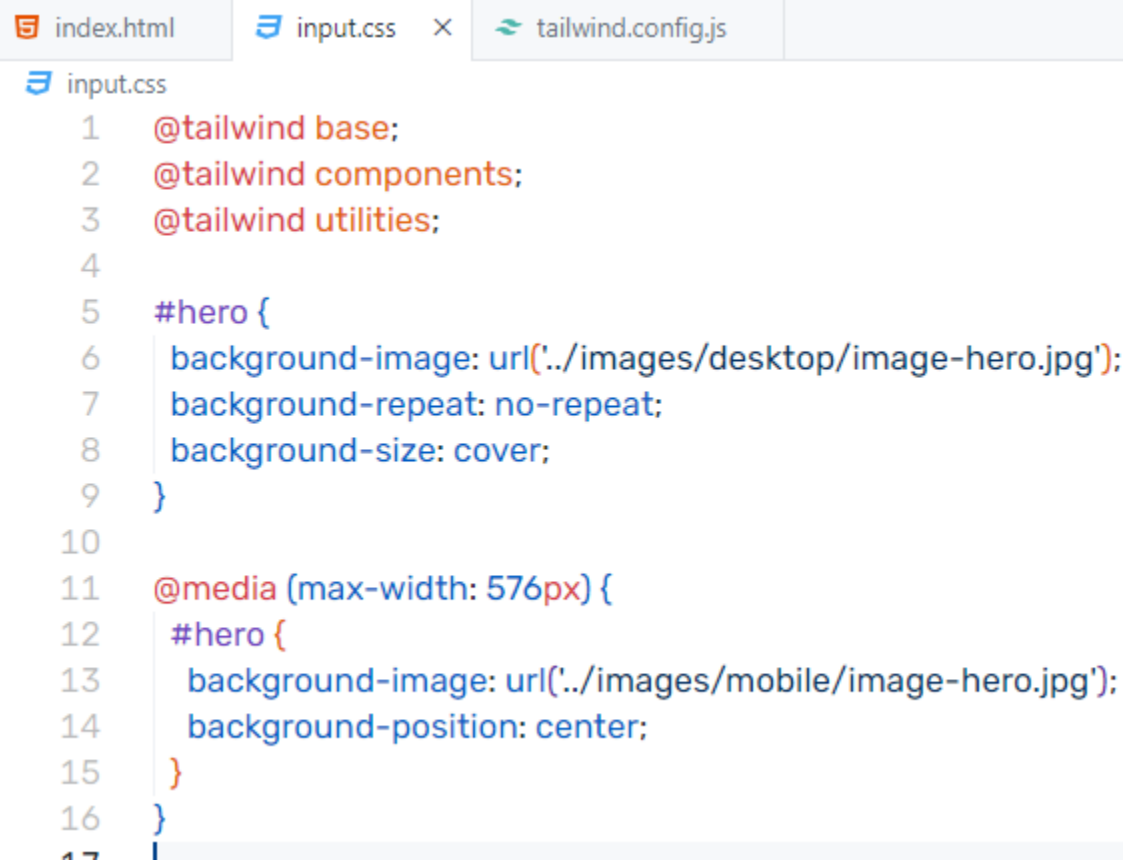
```
<!-- Menu Container -->
<div
  class="flex flex-col items-center justify-center space-y-3
  md:flex-row md:space-y-0 md:space-x-8 md:mb-24 md:justify-end"
>
  <!-- Menu Items -->
  <div class="group">
    <a href="#">Vector</a>
    <div
      class="mx-2 mt-2 duration-500 border-b-2 opacity-0
      border-black group-hover:opacity-100"
    ></div>
  </div>
```

```
<div class="group">
  <a href="#">Vector</a>
  <div
    class="mx-2 mt-2 duration-500 border-b-2 opacity-0
    border-black group-hover:opacity-100"
  ></div>
</div>
<div class="group">
  <a href="#">Vector</a>
  <div
    class="mx-2 mt-2 duration-500 border-b-2 opacity-0
    border-black group-hover:opacity-100"
  ></div>
</div>
<div class="group">
```

\*\*\*\*\*

```
<div class="hidden h-10 font-alata md:flex md:space-x-8">
|
</div>
```

By default hidden Rakha ha or md n above flex.



```
index.html  input.css  ×  tailwind.config.js
input.css
1  @tailwind base;
2  @tailwind components;
3  @tailwind utilities;
4
5  #hero {
6    background-image: url('../images/desktop/image-hero.jpg');
7    background-repeat: no-repeat;
8    background-size: cover;
9  }
10
11 @media (max-width: 576px) {
12   #hero {
13     background-image: url('../images/mobile/image-hero.jpg');
14     background-position: center;
15   }
16 }
```

url me link aesy pretend krkr do k ham css folder k ander wali css me likh rahy hen q k wo compil hogi or hamary project me lagay gi.

\*\*\*\*\*

Below we are over write default screen sizes, and putting our own font size .

```

tailwind.config.js > ...
1  module.exports = {
2    content: ['./*.html'],
3    theme: {
4      screens: {
5        sm: '480px',
6        md: '768px',
7        lg: '976px',
8        xl: '1440px',
9      },
10   extend: {
11     fontFamily: {
12       sans: ['Josefin Sans', 'sans-serif'],
13       alata: ['Alata'],
14     },
15     letterSpacing: {
16       widest: '.3em',
17     },

```

\*\*\*\*\*

First hero section:

Ek section banaya or usme ek container define kia , uski width k liye container ki class di , or max-w-6xl define kr di .....

Container responsive behaivur dega or max width se zuada uski width nahi hogi .

```

<div class="group">
  <a href="#">About</a>
  <div
    class="mx-2 group-hover:border-b [group-hover:border-blue-50"
  ></div>
</div>
<div class="group">
  <a href="#">Careers</a>
  <div
    class="mx-2 group-hover:border-b [group-hover:border-blue-50"

```

Yahan hamen group os liye dia k har ek item ko alag se hover na krna paray !

group class kaafi important hai! Iska maqsad **parent-child relationship** ke andar hover effects ko apply karna hai. Chaliye isko step-by-step samajhte hain:

## What is the group class in Tailwind CSS?

- group Tailwind CSS ki ek **utility class** hai jo parent element ke hover ya focus state ke basis par child elements ko style karne ki facility deti hai.
- Yeh ek **CSS pseudo-class** (:hover, :focus, etc.) ko propagate karta hai parent se child ke upar.

```
<div class="group">
  <a href="#">About</a>
  <div
    class="mx-2 group-hover:border-b group-hover:border-blue-50"
  ></div>
</div>
```

## Explanation:

### 1. group:

- Yeh parent <div> ko ek **group container** banata hai.
- Iska matlab hai ki parent ki hover state child elements ke liye available hogi.

### 2. group-hover in Child:

- **group-hover:border-b:**
  - Jab parent (group class wala <div>) par hover hoga, tab child <div> ke neeche ek **bottom border** add hoga.
- **group-hover:border-blue-50:**
  - Border ka color hover par **blue-50** hoga.

### 3. Without group:

- Agar group class na ho, to child element ko **parent ke hover** par target karna possible nahi hoga.
- For example, CSS me parent hover ke basis par child ko modify karne ke liye aapko manual CSS likhni padti:

```
.parent:hover .child {
  /* Styles */
}
```

- Tailwind me `group` yeh kaam aapke liye simplify karta hai.

## Rendered Behavior:

1. Jab aap kisi menu item par hover karenge (e.g., "About"), tab uske neeche:
  - Ek **bottom border** lagega.
  - Border ka color blue hoga.
2. Har <div> ke andar ek **link** (<a>) aur ek hidden **bottom border** (<div>) hai jo **hover state** par visible hota hai.

\*\*\*\*\*

Border k liye alag se div laga ha !

\*\*\*\*\*

```
class="hidden w-full duration-200 md:block
group-hover:scale-110"
/>
<!-- Mobile Image -->

<!-- Item Gradient -->
<div
  class="absolute top-0 bottom-0 right-0 left-0
  bg-gradient-to-b from-transparent to-gray-900
  group-hover:from-gray-50 group-hover:to-white
  group-hover:opacity-70"
></div>
<!-- Item Text -->
<h5
  class="absolute px-6 duration-200 w-52 bottom-4
  md:bottom-8 md:px-10 group-hover:scale-110
```

Image k upper gradient on hover....

\*\*\*\*\*

Key Note : Responsive classes sirf md k liye do baki container ki class khud responsiveness handle krlega !

\*\*\*\*\*

### **Hamburger button:**

```
<!-- Hamburger Button -->
<div class="md:hidden">
  <button
    id="menu-btn"
    type="button"
    class="z-40 block hamburger md:hidden focus:outline-none"
  >
    <span class="hamburger-top"></span>
    <span class="hamburger-middle"></span>
    <span class="hamburger-bottom"></span>
  </button>
</div>
```

Lines ko css ki help se banaya w & h

se, span ko target krke color, h, w de di .....

CSS For that:

```
/* Hamburger Menu */
.hamburger {
  cursor: pointer;
  width: 24px;
  height: 24px;
  transition: all 0.25s;
  position: relative;
}

.hamburger-top,
.hamburger-middle,
.hamburger-bottom {
  position: absolute;
  width: 24px;
  height: 2px;
  top: 0;
  left: 0;
  background: #fff;
  transform: rotate(0);
  transition: all 0.5s;
}

.hamburger-middle {
  transform: translateY(7px);
}

.hamburger-bottom {
  transform: translateY(14px);
}

.open .hamburger-top {
  transform: rotate(45deg) translateY(6px) translateX(6px);
}
```

Now we will control this behaviour with JS we will add these classes via JS:

```
.open {
  transform: rotate(90deg);
}

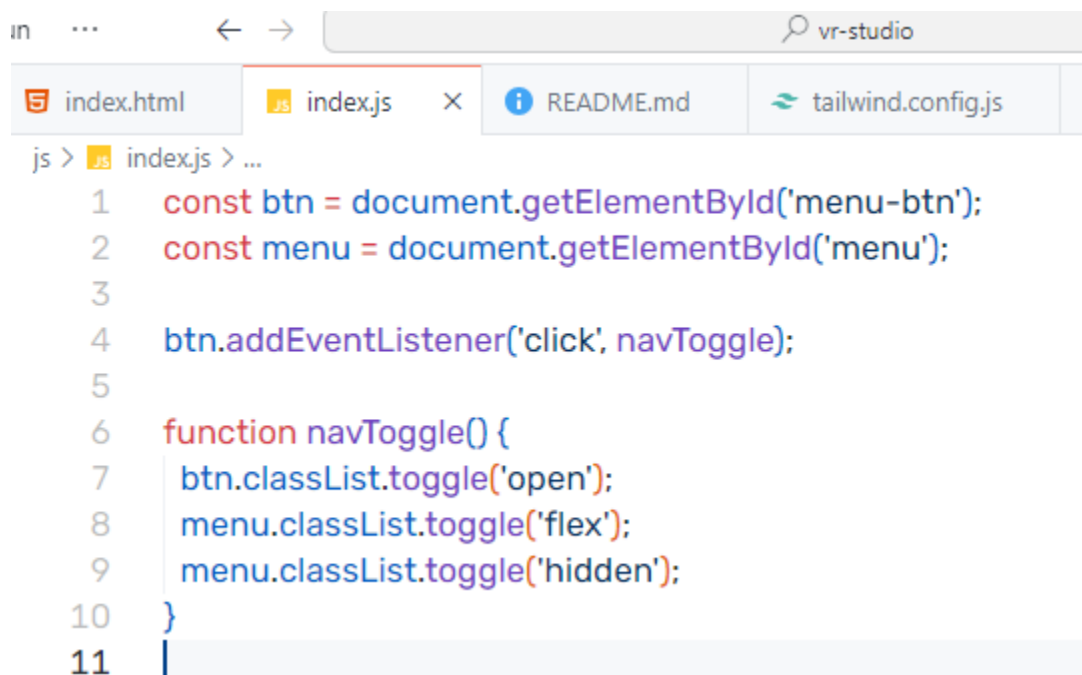
.open .hamburger-middle {
  display: none;
}

.open .hamburger-bottom {
  transform: rotate(-45deg) translateY(6px) translateX(-6px);
}
```

Now we will create a menu for this , when we click hamburger btn the links should appear :

```
<!-- Mobile Menu -->
<div
  id="menu"
  class="absolute top-0 bottom-0 left-0 hidden flex-col self-end w-full
  min-h-screen py-1 pt-40 pl-12 space-y-3 text-lg □ text-white uppercase ■ bg-black"
>
  <a href="#" class="□ hover:text-pink-500">About</a>
  <a href="#" class="□ hover:text-pink-500">Careers</a>
  <a href="#" class="□ hover:text-pink-500">Events</a>
  <a href="#" class="□ hover:text-pink-500">Products</a>
  <a href="#" class="□ hover:text-pink-500">Support</a>
</div>
```

JS :



The screenshot shows a code editor with a file explorer at the top displaying 'index.html', 'index.js', 'README.md', and 'tailwind.config.js'. The main editor area shows the following JavaScript code:

```
js > index.js > ...
1  const btn = document.getElementById('menu-btn');
2  const menu = document.getElementById('menu');
3
4  btn.addEventListener('click', navToggle);
5
6  function navToggle() {
7    btn.classList.toggle('open');
8    menu.classList.toggle('flex');
9    menu.classList.toggle('hidden');
10 }
11 |
```

\*\*\*\*\*