

Tailwind CSS Kullanımı

1- Kurulum

JIT desteği sayesinde artık çok daha hızlı projeler geliştirmemizi sağlayan Tailwind CSS'in mottosu HTML sayfasından CSS'e girmeden veya çok az giriş yaparak hızlı bir şekilde proje geliştirmemizi sağlayan framework. Bu CSS Framework sayesinde kullanmış olduğumuz CSS class'ların etiketlerini yazarak sadece onların çağırılmasını da sağlayabiliyoruz.

Kurulum için öncelikle <https://tailwindcss.com/docs/installation> linkinden **CLI**'i seçebiliriz. Bu yöntem en hızlı ve en olanaklı yöntemlerden bir tanesidir. İstek doğrultusunda **Bootstrap** gibi **CDN** olarak da çağırabiliriz. **CLI kurulumu ile Tailwind kullanabilmek için öncelikle node.js'in kurulu olması gerekiyor.**

Öncelikle **npm init -y** yaparak **package.json** dosyamızı oluşturuyoruz burada hangi eklentilerin kurulu olduğunu görebiliriz. Daha sonrasında **npm i -D tailwindcss** yaparak tailwind dosyamızı kullanıma hazır hale getiriyoruz. İşlemlerimiz gayet güzel gidiyorsa ayarlamalar yapılabilmesi için config dosyası oluşturulması gerekiyor bunun içinde dökümantasyonda yer alan terminalimizden **npm tailwindcss init** diyoruz ve config dosyamızın oluşturulmasına izin veriyoruz.

```
tailwind.config.js

module.exports = {
  content: ["/src/**/*.html"],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Dökümantasyonda dosyalarımızın hangi dizinde olduğunu gösteren bir açıklama var. Eğer işlemlerimiz **root** dizinde gerçekleştirmek istiyorsak content alanına **'./*html'** yapmamız yeterli.

Add the Tailwind directives to your CSS

Add the `@tailwind` directives for each of Tailwind's layers to your main CSS file.

```
src/input.css

@tailwind base;
@tailwind components;
@tailwind utilities;
```

Direktif ayarlarımızın verilebilmesi için ise dökümantasyon dosyamızda **input.css** dosyasını yaratıyoruz bu harici css dosyalarımız için base'imiz olacak.

Sıradaki kısım ise dosyalarımızın çalışabilir olmasını sağlayan path kısmı. Bunu aslında dokümantasyondan bağımsız olarak oluşturduğumuz **package.json** dosyamızda **Error** kısmını silerek belirlemesini yapıyorum. Çünkü, ilerleyen süreçte kontrolünü daha hızlı sağlayabilmem için bu bana ileride çok daha yararı dokunacak. Genel olarak ekran görüntüsü ekleyecek olursam izin ve yazılar alttaki şekilde olmalı.



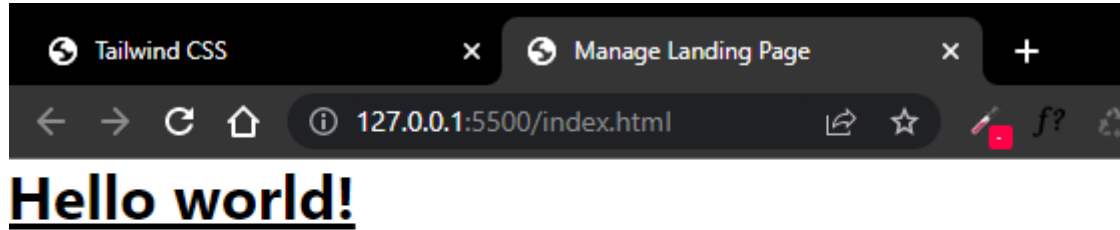
```
1 {
2   "name": "tailwindcss-demos",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "build": "tailwindcss -i ./input.css -o ./css/main.css",
8     "watch": "tailwindcss -i ./input.css -o ./css/main.css --watch"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "devDependencies": {
14    "tailwindcss": "^3.0.23"
15  }
16 }
```

Ve sonunda **npm run build** diyerek dosyalarımızı localimize çekiyoruz 🥳🎉
Oluşturulan css dosyasının içerisinde **main.css** ile anadizinimizin bu olduğunu görebiliriz.



```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta http-equiv="X-UA-Compatible" content="IE=edge" />
6     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7     <link rel="stylesheet" href="css/main.css" />
8     <title>Manage Landing Page</title>
9   </head>
10  <body>
11    <h1 class="text-3xl font-bold underline">Hello world!</h1>
12  </body>
13 </html>
```

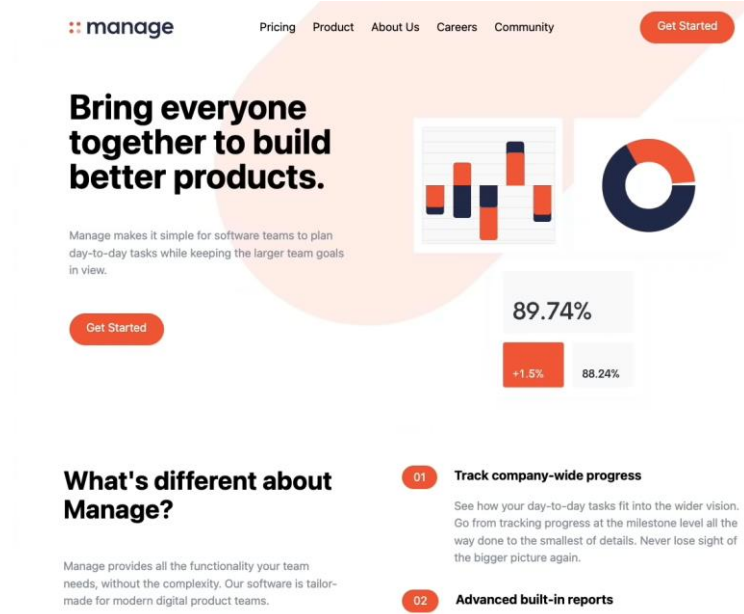
Hemen root dizinde **index.html** dosyasını oluşturalım ve css dosyamızı main dosyamızdan çağıralım. Aynı Bootstrap'te olduğu gibi class'a özelliklerimizi ekledik ve **npm run watch** diyerek görmek istiyoruz.



Her şey mükemmel görünüyor 😎 Kurulumumuz bu kadar. Devamında Özelleştirmeler ve Ayarlar kısmından söz edeceğim.

2- Özelleştirmeler & Ayarlar

Öncelikle, ilk defa kullanım için Visual Studio Code eklentisi olan **Tailwind CSS IntelliSense** topluluk tarafından önerilen bir eklenti. Bunun kurulumunu da gerçekleştiriyoruz. Bu olmazsa olmaz değil ama bana kalırsa olmazsa olmaz bir eklenti çünkü size kullanımda o kadar rahatlık sağlayacak.

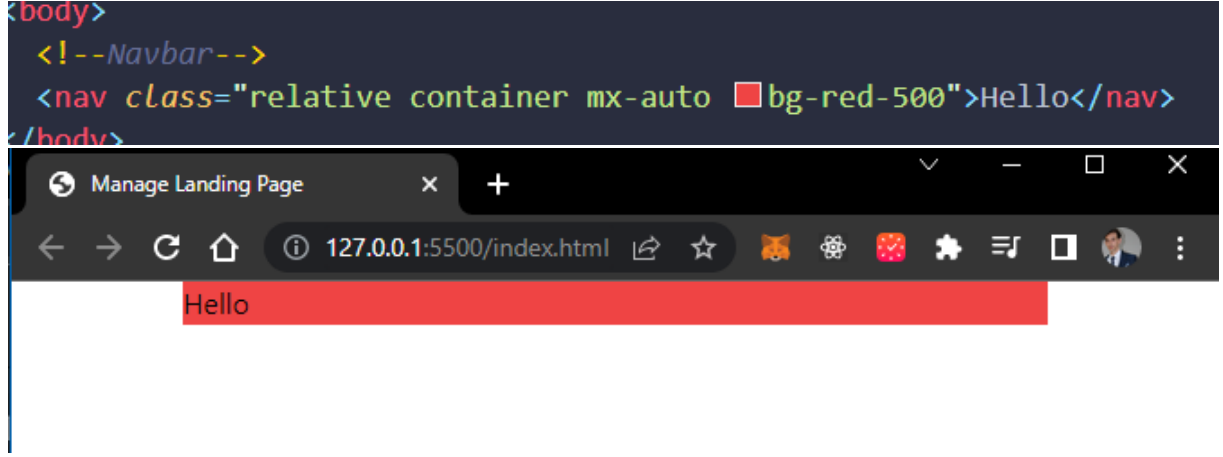


Proje üzerinden gitmek daha mantıklı olacağı için yukarıdaki tasarımı Tailwind sayesinde kullanılabilir hale çevireceğiz. Bunun için kendi breakpointlerimiz olması lazım. **tailwind.config.js** dosyasına gelerek theme'nin altına **screens** diye bölüm ayırıyorum. Sebebi ise basitçe breakpointler eklemek istiyorum. Bootstrap'te kullanılan small, medium, xl gibi. Daha sonrasında projemde kullanılacak olduğum renkleri ekliyorum. Bu sayede harici breakpointlerimizi bu şekilde kullanabileceğimizi öğreniyoruz.

```
File Edit Selection View Go Run Terminal Help
EXPLORER
  package.json
  main.css
  index.html
  tailwind.config.js
  TAILWINDCSS-DEMOS
    css
    main.css
    node_modules
    index.html
    input.css
    package-lock.json
    package.json
    tailwind.config.js
  package.json
  main.css
  index.html
  tailwind.config.js
  tailwind.config.js > ...
  module.exports = {
  content: ['./*.html'],
  theme: {
  screens: {
  sm: '480px',
  md: '768px',
  lg: '976px',
  xl: '1440px',
  },
  extend: {
  colors: {
  brightRed: 'hsl(12, 88%, 59%)',
  brightRedLight: 'hsl(12, 88%, 69%)',
  brightRedSupLight: 'hsl(12, 88%, 95%)',
  darkBlue: 'hsl(228, 39%, 23%)',
  darkGrayishBlue: 'hsl(227, 12%, 61%)',
  veryDarkBlue: 'hsl(233, 12%, 13%)',
  veryPaleRed: 'hsl(13, 100%, 96%)',
  veryLightGray: 'hsl(0, 0%, 98%)',
  },
  },
  },
  plugins: [],
  };
```

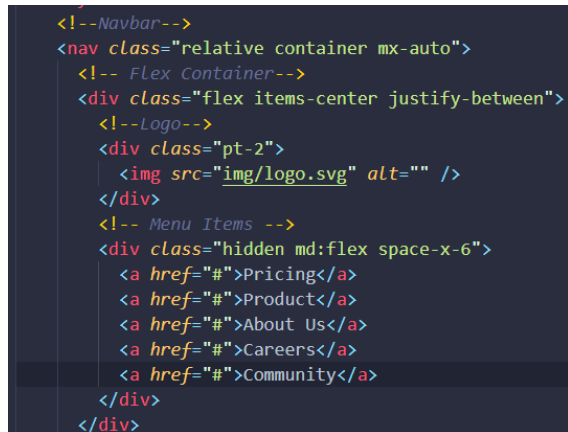
3- Menü Alanı, Spacing & Uyumluluk

Tasarımımızı, çalışabilir hale dökmenin zamanı geldi. Bunun için ilk olarak menü alanından başlayacağız. Birkaç test yapmak için navbar alanına bakıyorum.



Aynı nav class'ımızı oluşturduk ve ona relative özelliği verdik aynı zamanda container'ın içine alarak ortalamamızı sağladık. Testimizi görebilmek için ise arka plana renk verdik. Daha sonrasında ise istediğim ölçüde div'lerimi oluşturuyorum. Mantık olarak bir tane **Flex Container** olacak bunu da sağa sola yaslayarak yapacağım. Çünkü sol tarafta logoyu sağ tarafta ise menülerim olacak. **İsteğim, eğer mobil ölçülere geldiğinde menünün gözükmemesi.** Bu konuda Tailwind yardımımıza gerçekten jet hızıyla ulaşıyor. 🚀

Çünkü hidden class'ı sayesinde flex özelliğine görünürlüğünü ayarlayabiliyorum. Bu şekilde Responsive yapıyı çok rahat kurabiliyor ve mobilde aşağıya doğru uzanan bir menü yapacağım. Aşağıdaki görselden ne demek istediğimi daha iyi anlayabileceksiniz.





Tailwind sayesinde yapabildiğimiz güzel olaylardan bir tanesi ise **hover özelliği**. Bootstrap gibi kütüphanelerde olmayan bu özellik ile CSS'e gelmeden hızlıca hover ekleyebiliyoruz.

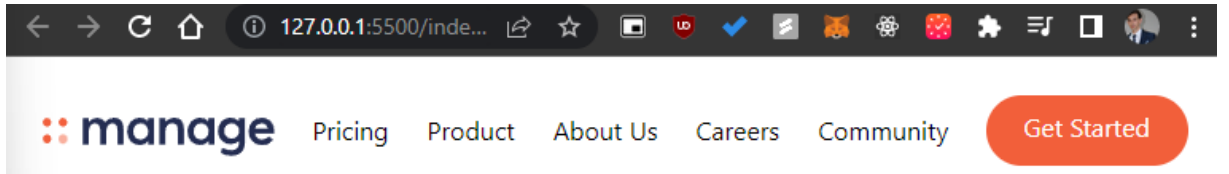
Pricing Product About Us Careers Community

```
<div class="hidden md:flex space-x-6">  
<a href="#" class="hover:text-darkGrayishBlue">Pricing</a>
```

Tasarımda bulunan sağ tarafta ki button'u eklemek istiyorum.

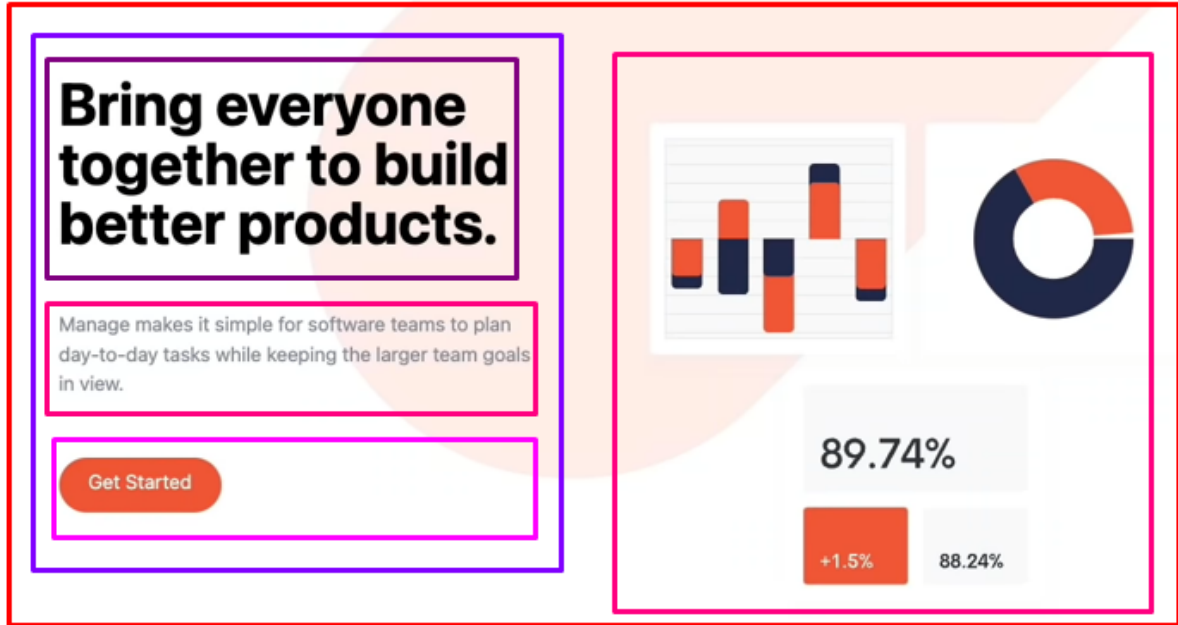
```
<!--Button-->  
<a  
  href="#"  
  class="hidden md:block p-3 px-6 pt-2 text-white bg-brightRed rounded-full baseline  
  hover:bg-brightRedLight"  
>Get Started</a>
```

Burayı açıklamak gerekirse, mobil olduğunda görüntülenmemesi, padding değerlerine 3 rem'lik boşluk bırakılmasını, yukarıdan aşağıdan 6 rem'lik alan ayarlamasını ve yukarıdan 2 remlik kısım belirlemesini istiyorum ve hover özelliği sayesinde üstüne gelindiğinde renk değiştirmesini istiyorum.



İstediğim gibi görünüyor şuanda, harika. ☒ Menü kısmını istediğimiz gibi tamamladığımız için artık **Hero Section**'a geçiş yapabiliriz.

4 – Hero Section



Yapmak istediğim kısım şuanda genellikle Hero Section olarak isimlendirilen bu bölüm. Şöyle bir baktığımızda bir tane h1, daha sonrasında paragraf flex olarak ikiye ayrılmış bir alan ve buton olduğunu görüyoruz ve küçülttüğünde mobil görünüm sağlandığında resmin yukarıda olmasını istiyorum. <nav> etiketlerinden sonra Hero Section kısmını yazmaya başlıyorum.

```
<!-- Hero Section -->

<section id="hero">
  <!-- Flex Container -->
  <div
    class="container flex flex-col-reverse md:flex-row items-center px-6 mx-auto mt-10 space-y-0 md:space-y-0"
  >
    <div class="flex flex-col mb-32 space-y-12 md:w-1/2">
      <h1
        class="max-w-md text-4xl font-bold text-center md:text-5xl md:text-left"
      >
        Bring everyone together to build better products.
      </h1>
      <p class="max-w-sm text-center text-darkGrayishBlue md:text-left">
        Manage makes it simple for software teams to plan day-to-day tasks
        while keeping the larger team goals in view.
      </p>
    </div>

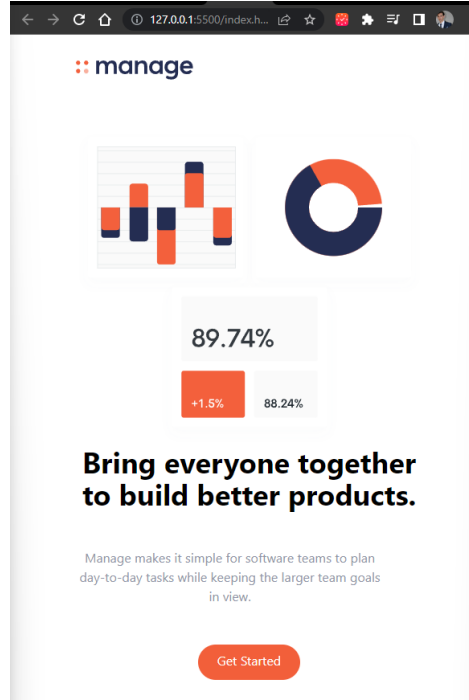
    <div>
      <!--Image-->
    </div>
  </div>
</section>
```

Id'si hero olan bir tane section açarak kodlarımı yazmaya başladım. Buraya hero alanında olacak tüm kodlarımı yazıyorum. Mantık olarak sol tarafta yazılar ve sağ tarafta yazılar olacağı için ikiye böldüm. Kodları açıklayacak olursak bunları birer container'in

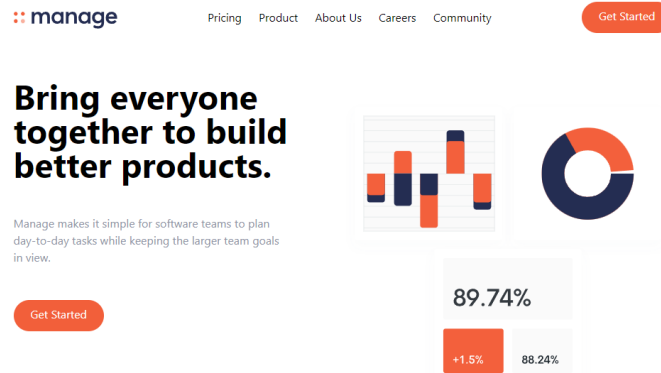
içerisine aldık ve flex özelliği verdik ve flex-col-reverse yaparak sondan başlamasını istedik. Sebebi ise resim kısmının çözünürlük düştüğünde yukarıda olmasını istiyorum. Mobil kısımda ise flex özelliğinin satır olmasını istedik ve her şeyi ortalayarak gerekli yükseklik ayarlarını verdik. Daha sonrasında yazı bölümüne gelerek **Tailwind**'ın güzel özelliklerinden bir tanesi olan yüzde ifadelerini vererek mobilde içeriklerin yüzde elli küçültülmesini istedik. Başlık ve Paragraf özelliklerimizi de ona göre düzenledik. Büyük ekranda metinler sola yaslı iken mobilde ortalı olması sağlandı.

```
<!--Image-->
<div class="md:w-1/2">
  
</div>
</div>
```

Artık metin işlemlerimiz bittiğine göre resim kısmını ekleme zamanı geldi ve class'ına mobilde yüzde elli küçültülerek eklenmesini istedik. Yukarıda bahsettiğim gibi flex-col-reverse özelliği sayesinde artık mobilde resim yukarıda görüntülenmesini sağladık. Çalışma mobilde aşağıda ki görünüme sahipken



Normal, bilgisayardan görünümü ise bu şekilde.



4 – Features Section

What's different about Manage?

Manage provides all the functionality your team needs, without the complexity. Our software is tailor-made for modern digital product teams.

01

Track company-wide progress

See how your day-to-day tasks fit into the wider vision. Go from tracking progress at the milestone level all the way done to the smallest of details. Never lose sight of the bigger picture again.

02

Advanced built-in reports

Set internal delivery estimates and track progress toward company goals. Our customisable dashboard helps you build out the reports you need to keep key stakeholders informed.

03

Everything you need in one place

Stop jumping from one service to another to communicate, store files, track tasks and share documents. Manage offers an all-in-one team productivity solution.

Features section için yukarıda ki gibi bir tasarım yapmam gerekiyor, yine Hero Sectionda ki gibi features id'li bir tane div oluştuyorum. İşlemler artık birbirlerine çok benzediği için yapılmak istenilen ve nasıl yapıldığına dair anlatmaya doğru geçiş yapıyorum. Yukarıda ki görselimize baktığımızda ikiye bölünmüş bir alan var ve bunlar container'ın içinde fakat mobil kısmında soldaki yazılar yukarıda ve ortalı sağda ki numaralı liste ise aşağıya doğru olmak zorunda.

```
<!-- Numbered List -->
<div class="flex flex-col space-y-8 md:w-1/2">
  <!-- List Item 1 -->
  <div
    class="flex flex-col space-y-3 md:space-y-0 md:space-x-6 md:flex-row"
  >
    <!-- Heading -->
    <div class="rounded-l-full bg-brightRedSupLight md:bg-transparent">
      <div class="flex items-center space-x-2">
        <div
          class="px-4 py-2 text-white rounded-full md:py-1 bg-brightRed"
        >
          01
        </div>
        <h3 class="text-base font-bold md:mb-4 md:hidden">
          Track company-wide progress
        </h3>
      </div>
    </div>
    <div>
      <h3 class="hidden mb-4 text-lg font-bold md:block">
        Track company-wide progress
      </h3>
      <p class="text-darkGrayishBlue">
        See how your day-to-day tasks fit into the wider vision. Go from
        tracking progress at the milestone level all the way done to the
        smallest of details. Never lose sight of the bigger picture
        again.
      </p>
    </div>
  </div>
</div>
```


Liste kısımlarını bu şekilde tamamlayabiliriz. Farklı işlemler yaptık burada çünkü, mobilde ki görünümde arka planda turuncu bir şekilde sıralı gelmesini istedik. Mobile first tasarım prensibinden dolayı normal arkaplan rengini kendi eklemiş olduğumuz açık kırmızı olduğunu fakat ekran genişliği yükseltildiğinde transparan olmasını sağladık ve numaraları yuvarlak içine aldık. Tasarım mobil kısımda aşağıda ki gibi görünüyor.

What's different about Manage?

Manage provides all the functionality your team needs, without the complexity. Our software is tailor-made for modern digital product teams.

01 Track company-wide progress

See how your day-to-day tasks fit into the wider vision. Go from tracking progress at the milestone level all the way down to the smallest of details. Never lose sight of the bigger picture again.

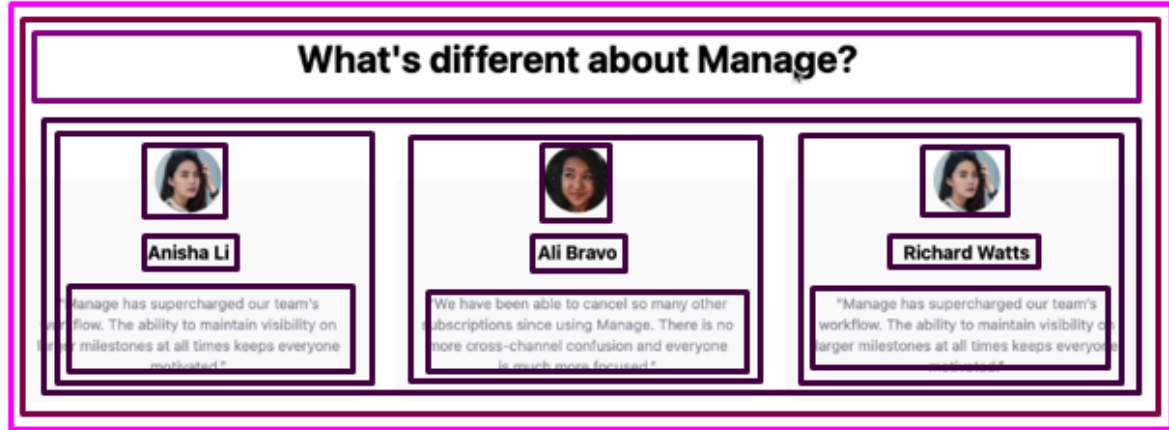
02 Advanced built-in reports

Set internal delivery estimates and track progress toward company goals. Our customisable dashboard helps you build out the reports you need to keep key stakeholders informed.

03 Everything you need in one place

Stop jumping from one service to another to communicate, store files, track tasks and share documents. Manage offers an all-in-one team productivity solution.

4 – Testimonial Section



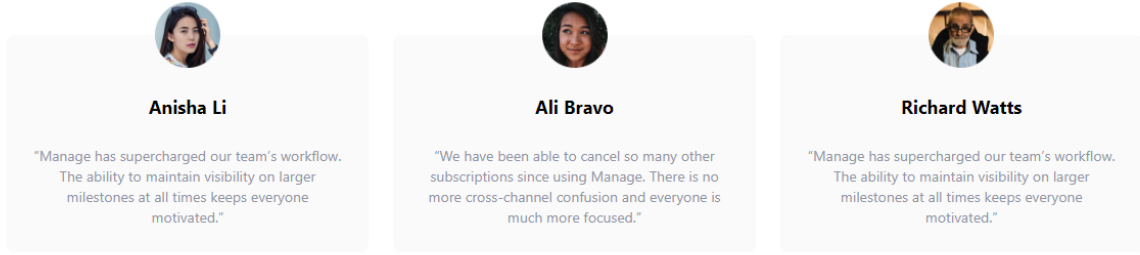
Tasarımımızda böyle bir testimonial alanı var. Yukarıda resim de gördüğünüz gibi bir yapıya bölmem gerekiyor. En dışarda genel bir id'si testimonial olan bir div, daha sonrasında hepsini kapsayan ayrı bir container, heading alanı. Sonra ki adımda ise yorumların hepsini bir içine alan bir alan ve onun alt başlıkları.

```
<!--Testimonials-->

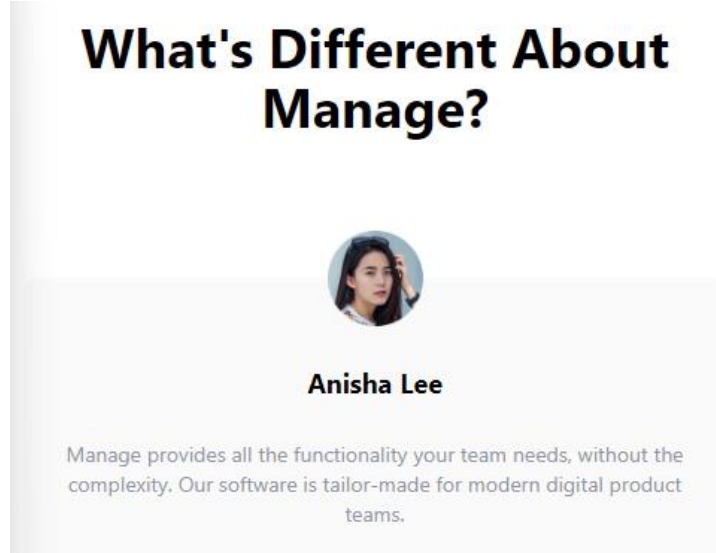
<section id="testimonial">
  <!--Container to heading and testm blocks-->
  <div class="max-w-6xl px-5 mx-auto mt-32 text-center">
    <!--Heading-->
    <h2 class="text-4xl font-bold text-center">
      What's Different About Manage?
    </h2>
    <!--Testimonial Container-->
    <div class="flex flex-col mt-24 md:flex-row md:space-x-6">
      <!--Testimonial 1-->
      <div
        class="flex flex-col items-center p-6 space-y-6 rounded-lg bg-veryLightGray md:w-1/3"
      >
        
        <h5 class="text-lg font-bold">Anisha Lee</h5>
        <p class="text-sm text-darkGrayishBlue">
          Manage provides all the functionality your team needs, without the
          complexity. Our software is tailor-made for modern digital product
          teams.
        </p>
      </div>
    </div>
  </div>
</section>
```

Kodlamasını bu şekilde yaptım ve negatif margin değeri ekleyerek fotoğraflarının daha yukarda olmasını sağladım. Mobilde ise sadece bir tanesini göstermek istediğim için görünürlüklerini kapattım.

```
<!--Testimonial 2-->
<div
  class="hidden flex flex-col items-center p-6 space-y-6 rounded-lg bg-veryLightGray md:flex md:w-1/3"
>
```

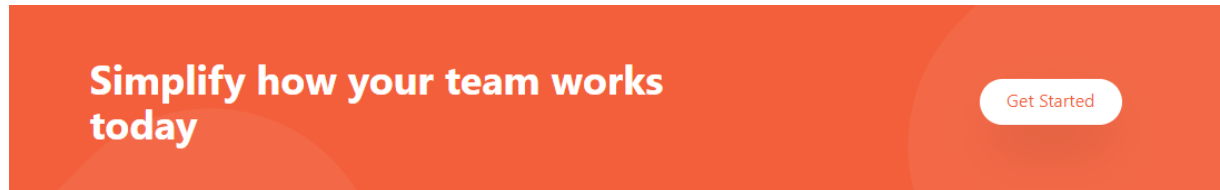


Bu görünümü elde ettim, mobilde ise istediğim gibi aşağıda ki görüntüye sahip.



Testimonials kısmı şimdilik bu kadar. İlerleyen süreçte aslında Carousel kısmı yapılabilir.

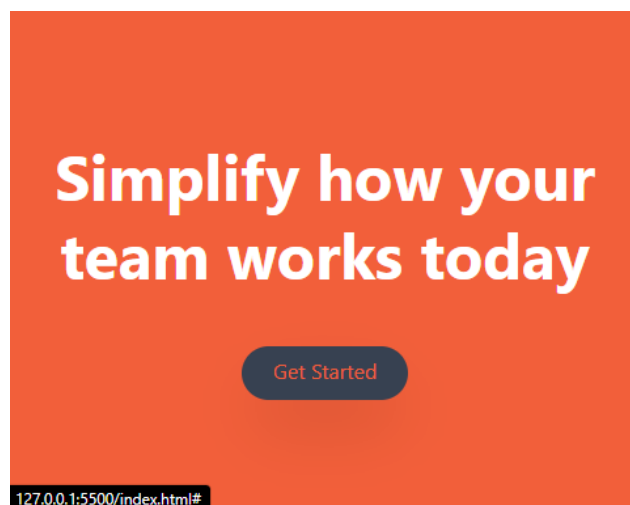
4 – CTA Section



Bu kısım aslında diğerlerinden çok daha basit ve uğraş gerektirmiyor. İki tane column'a bölünmüş, sağ tarafta buton ve sol tarafta yazı olacak bir şekilde bölüm ayarlayıp bu işi sonlandıracağım. Arkaplan görselleri ise en son tüm iskeletler bittikten sonra custom CSS ile eklemeler yapacağım.

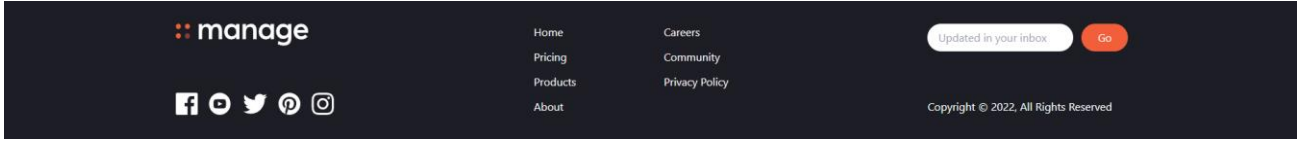
```
<section id="cta" class="bg-brightRed">
  <!--Flex Container-->
  <div
    class="container flex flex-col items-center justify-between px-6 py-24 mx-auto space-y-12 md:py-12 md:flex-row space-y-0"
  >
    <!--Heading-->
    <h2
      class="text-5xl font-bold leading-tight text-center text-white md:text-4xl md:max-w-xl md:text-left"
    >
      Simplify how your team works today
    </h2>
```

Yukarıda ki kodları yazdıktan sonra işlemim bitiyor. Uzun uzun class isimlerini yazdığımı farkındayım fakat burada ki asıl amacımızın sadece Tailwind CSS kullanarak bir tasarımı CSS'e dökmek olduğunu unutmayalım.



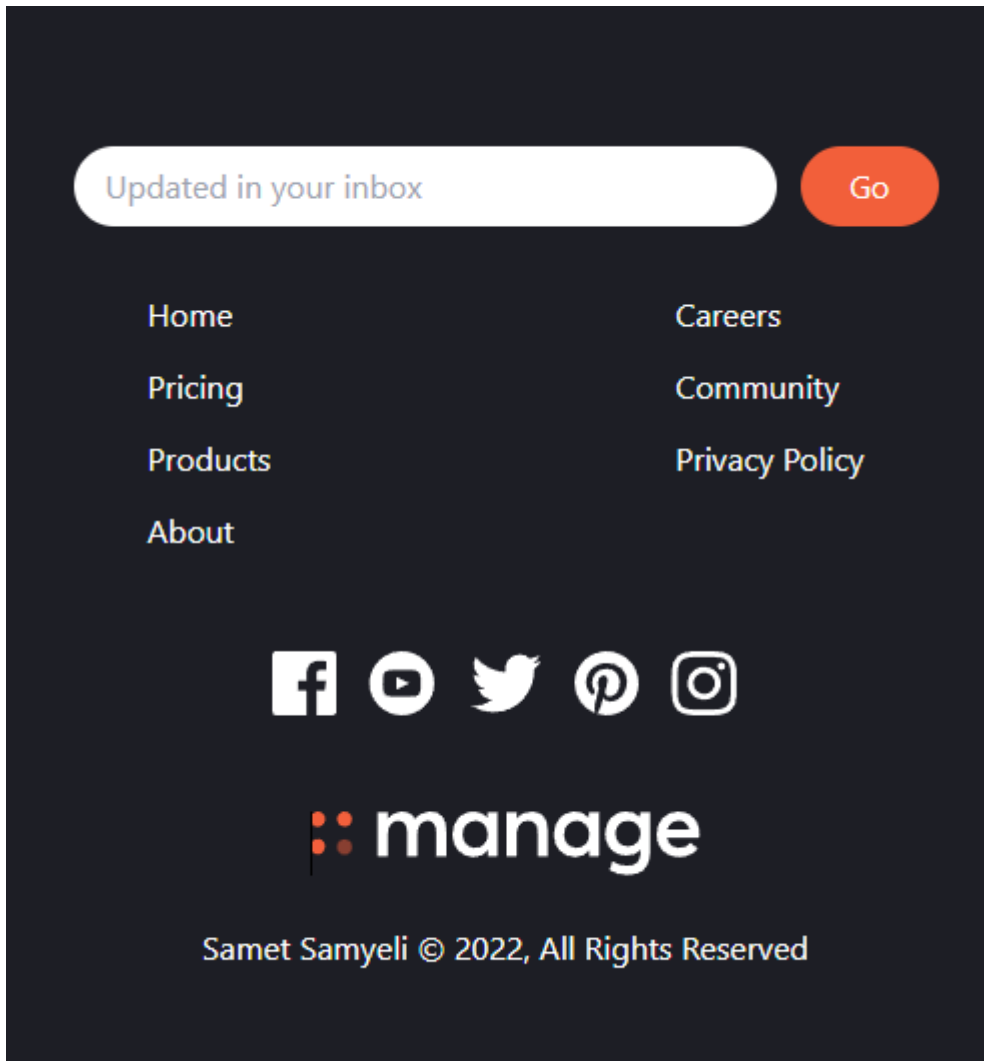
Mobil kısımda button'a hover olduğunda olan kısım bu şekilde görünüyor.

4 – Footer



Güzel bir alana geçiş yaptığımızı söylebilirim. 😊 Çünkü benzer işlemleri yapmak bir müddet sonra insanı sıkabiliyor. Burada yapmak istediğimiz dışarıda footer oluşturmak daha sonrasında her zaman ki gibi hepsini kapsayan bir div ve içerlerini 3'e bölmek. Sadece Tailwind Class'larını kullanmak için bu alan güzel bir pratik olabilir.

Hemen kodlarımı yazmaya başlıyorum. Aynı ilk örnekte olduğu gibi **flex-col-reverse** örneğini kullanarak logo'yu yukarıya taşımayı düşünüyorum.



Mobilde footer'ın böyle görünen bir yapı kurdum. Kodları Footer alanı için açıklamayacağım. Github üzerinden projenin bitmiş halinden inceleyebilirsiniz. Burada ki eksik kalan yapılar kaldı. Bunlara örnek olarak Menu mobilde görüntülenmemesi bunun için Hamburger Menu işlemine giriş yapacağım.

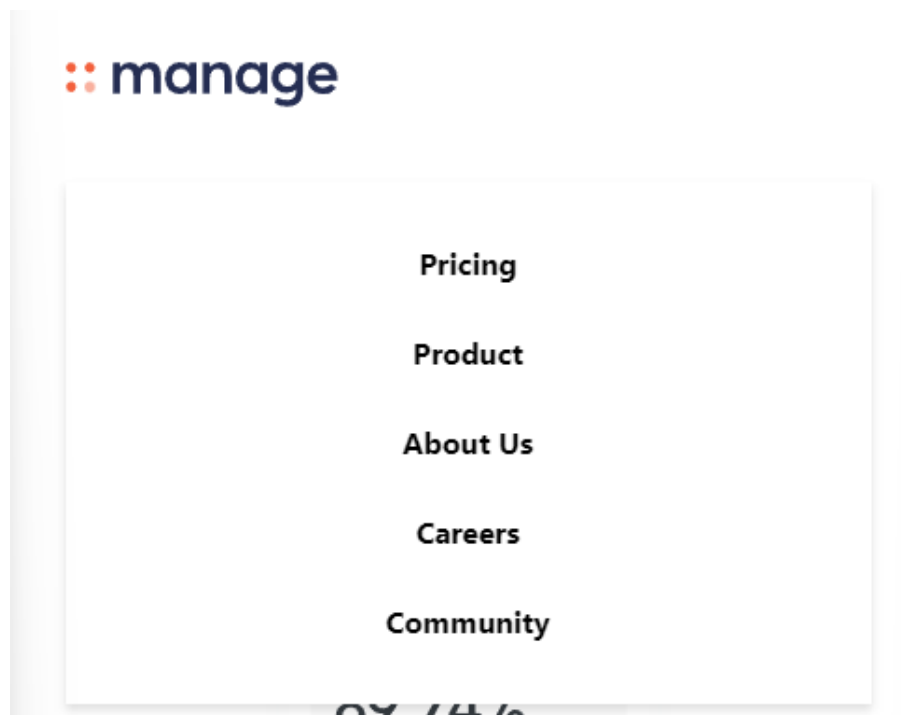
4 – Hamburger Menu HTML

Projenin eksik kalan kısımlarından bir tanesi Hamburger menüydü bunun için **<nav>** etiketlerimizin bitişinden önceki yerine gelip hemen yapıyı kurmaya başlıyorum. Yapı için öncelikle Hamburger Icon kısmını oluşturuyorum hamburger-top, hamburger-middle ve hamburger-bottom olarak ve görüntülenmesini hidden olarak ayarlıyorum.

```
<!-- Hamburger Icon -->

<button id="menu-btn block hamburger md:hidden focus:outline-none">
  <span class="hamburger-top"></span>
  <span class="hamburger-middle"></span>
  <span class="hamburger-bottom"></span>
</button>
```

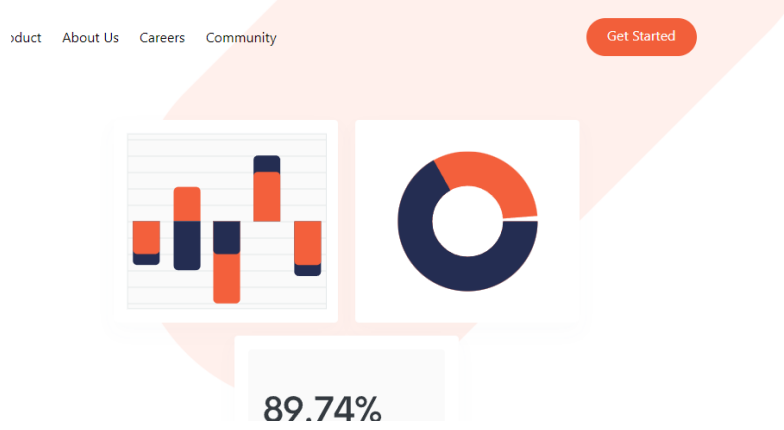
Daha sonrasında mobil için gösterilecek olan kısmı ayarladım. Bunun için absolute değerlerinde olacak ve biraz arkaplan gölgesi eklenmesi iyi olur. Genel olarak görünümü aşağıda ki gibi duruyor.



Yapmak istediğim tasarımı bitirdikten sonra **hidden** diyerek gizliyorum. Bu kısmı Javascript sayesinde daha iyi bir şekilde ayarlayacağım. Aslında Javascript kullanmadan düz CSS ile animations değerleri ile de yapabilirim fakat Javascript kullanmak daha kolay bir çözüm sunacak. 😊 Şuanda ise ilk başta icon değerlerini verdiğimiz alana Custom CSS kısımlarını ve daha sonra ekleyeceğim dediğim arkaplan görsellerini ekleme işlemini gerçekleştireceğim.

4 – Vanillia CSS ve Menü için Biraz Javascript

Evet, şuna kadar tasarlanmış olan tasarımın tamamına yakınına sadece ve sadece **Tailwind** teknolojisini kullanarak bu noktaya getirdik. HTML sayfasından başka hiçbir işlem yapmadık ve bu gerçekten inanılmaz. 🎉👏 Artık bi' nebze de olsa kendi CSS işlemlerimizi gerçekleştirmemiz gerekiyor. Olması gereken tasarımda aşağıda ki gibi alanlarda arkaplan SVG dosyaları bulunuyor.



Hemen tasarım işlemlerini tamamlamak için **input.css** dosyamıza gidiyoruz. Burada yapmamız gereken işlemler var. İlk olarak arkaplan SVG dosyalarımızı ayarlayarak başlıyorum.

```
@tailwind base;
@tailwind components;
@tailwind utilities;

body {
  background-image: url('../img/bg-tablet-pattern.svg');
  background-repeat: no-repeat;
  background-size: 800px;
  background-position: 90% -25%;
}

#cta {
  background-image: url('../img/bg-simplify-section-desktop.svg');
  background-repeat: no-repeat;
}

@media (max-width: 576px) {
  body {
    background-position: 50px -50px;
    background-size: 500px;
  }

  #cta {
    background-image: url('../img/bg-simplify-section-mobile.svg');
  }
}
```

Klasik Vanillia CSS yazarak gerekli arkaplan işlemlerimizi gerçekleştirdik ve **Responsive** özelliği için 576 px'ın altında olduğunda ona göre konumlandırmasını yaptık.

Hamburger Menu tasarımı için ise öncelikle dışarıda ki genel hamburger classına daha sonrasında ise top, middle ve bottom oluşturduğumuz div'lere gerekli özellikleri verdim.

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

Bu CSS dosyasını anlatacak olursam imlecimiz üstüne geldiğinde tıklama ile değiştirecek ve 24'e 24 piksel boyutunda bir alan gerektirdiğini ve geçişli bir işlem olduğunu belirtiyorum.

```
.hamburger-top,  
.hamburger-middle,  
.hamburger-bottom {  
  position: absolute;  
  top: 0;  
  left: 0;  
  width: 24px;  
  height: 2px;  
  background: black #000;  
  transform: rotate(0);  
  transition: all 0.5s;  
}  
  
.hamburger-middle {  
  transform: translateY(7px);  
}  
  
.hamburger-bottom {  
  transform: translateY(14px);  
}
```

Devamında ise 3'üne genel işlemler verirken top classımızı almıyorum çünkü istiyorum ki, tıklanıldığında döndürme işlemi olup çarpı butonuna dönüş yapsın. Devamını Javascript kullanarak devam edeceğim.


```
const btn = document.getElementById('menu-btn')
const nav = document.getElementById('menu')

btn.addEventListener('click', () => {
  btn.classList.toggle('open')
  nav.classList.toggle('flex')
  nav.classList.toggle('hidden')
})
```

Anadizinde **script.js** dosyamı açarak yukarıda ki kodlarımı yazdım ve kullanmış olduğum **HTML** sayfama gerekli eklemeleri yaptım. Yukarıda ki kod parçasında ise sabit btn ve nav değişkenleri oluşturarak menu-btn ve menu id'lerini yakaladım ve addEventListener click işlemi sayesinde open, flex ve hidden işlemlerini gerçekleştirdim. Daha sonrasında ise **input.css** dosyama geri giderek bu işlemlere CSS özellikleri ekledim.

```
.open {
  transform: rotate(90deg);
  transform: translateY(0px);
}

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

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

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

5 – Ve Sonuç 🎉🎊🥳

Desktop'ta bu şekilde, mobilde ise aşağıda ki gibi görünen tasarımı kullanılabilir hale getirdik ve bunların hepsini hemen hemen hiç Vanillia CSS kullanmadan sadece Tailwind CSS Framework'unu kullanarak gerçekleştirdik. Sonuç gerçekten muazzam.

