



in/poyrazavsever

→|

ATOMIC DESIGN: UI MİMARİSİ NASIL KURULUR?

**Atom → Molekül →  
Organizma → Şablon  
→ Sayfa**



in/poyrazavsever

→|

## Neden Atomic Design?

Tekrarlanabilirlik, tutarlılık ve ölçülebilirlik için UI'ı en küçük parçadan bütüne kurarız.

Bu model, ekip iletişimini de netleştirir: herkes aynı dilde konuşur (atom, molekül, organizma...).

in/poyrazavsever



## Atom = Tek Sorumluluk

Buton, ikon, input gibi bölünemez, tek sorumlu bileşenler.

Stiller tasarım token'larından gelir.



```
// atoms/Button.tsx
import React from "react";

type ButtonProps = React.ButtonHTMLAttributes<HTMLButtonElement> & {
  variant?: "primary" | "secondary";
};

export function Button({ variant = "primary", className = "", ...props }: ButtonProps) {
  const base = "inline-flex items-center justify-center rounded-lg px-4 py-2 text-sm font-medium";
  const variants = {
    primary: "bg-indigo-600 text-white hover:bg-indigo-700",
    secondary: "bg-stone-200 text-stone-900 hover:bg-stone-300",
  };
  return <button className={`${base} ${variants[variant]} ${className}`} {...props} />;
}
```

in/poyrazavsever



## Molekül = Bir Amaca Hizmet Eden Küçük Grup

Input + Button birleşip “Arama Çubuğu” gibi tek bir görevi yerine getirir.



```
// molecules/SearchBar.tsx
import React, { useState } from "react";
import { Button } from "../atoms/Button";

export function SearchBar({ onSearch }: { onSearch: (q: string) => void }) {
  const [q, setQ] = useState("");
  return (
    <div className="flex gap-2">
      <input
        value={q}
        onChange={(e) => setQ(e.target.value)}
        placeholder="Ara..."
        className="border rounded-lg px-3 py-2"
      />
      <Button onClick={() => onSearch(q)}>Ara</Button>
    </div>
  );
}
```

in/poyrazavsever



## Organizma = Bağımsız, Yeniden Kullanılabilir Bölüm

NavBar, ProductCardList, Hero gibi, sayfanın anlamlı bölümü.



```
// organisms/Header.tsx
import React from "react";
import { Button } from "../atoms/Button";
import Link from "next/link";

export function Header() {
  return (
    <header className="flex items-center justify-between py-4">
      <Link href="/" className="text-lg font-semibold">Brand</Link>
      <nav className="flex items-center gap-4">
        <Link href="/docs">Docs</Link>
        <Link href="/pricing">Pricing</Link>
        <Button>Giriş yap</Button>
      </nav>
    </header>
  );
}
```

in/poyrazavsever



## Template = İçeriksiz Sayfa İskeleti

Yerleşimi tanımlar, veri bağlamaz. Organizma yerleşimleri sabitlenir, içerik sonradan gelir.

```
// templates/MarketingTemplate.tsx
import React from "react";
import { Header } from "../organisms/Header";

export function MarketingTemplate({ hero, features, cta }: {
  hero: React.ReactNode; features: React.ReactNode; cta: React.ReactNode;
}) {
  return (
    <>
      <Header />
      <main className="container mx-auto px-4">
        <section className="py-12">{hero}</section>
        <section className="py-12">{features}</section>
        <section className="py-12">{cta}</section>
      </main>
    </>
  );
}
```

in/poyrazavsever



## Page = Template + Gerçek içerik

Veri bağlanır, template'in slot'lara içeriği yerleştirir,  
route/SEO gibi bağlantıları yönetir.

```
// app/marketing/page.jsx
import { MarketingTemplate } from "@/components/templates/MarketingTemplate";
import { Button } from "@/components/atoms/Button";

export default function Page() {
  const Hero = (
    <div>
      <h1 className="text-3xl font-bold">Hızlı başla</h1>
      <p className="text-stone-600">Modern Frontend ile daha az karmaşa.</p>
    </div>
  );

  const Features = (
    <ul className="grid grid-cols-1 md:grid-cols-3 gap-6">
      <li>Atomic Design</li><li>Performans</li><li>Erişilebilirlik</li>
    </ul>
  );

  const CTA = <Button variant="primary">Ücretsiz deneme</Button>

  return <MarketingTemplate hero={Hero} features={Features} cta={CTA} />;
}
```



in/poyrazavsever

→|

## Tutarlılık Token'larla Gelir

Renk, tipografi, spacing, gölge gibi stil kararlarını tokenlaştırır. Atomlar token tüketir, tüm hiyerarşi tutarlı kalır.

in/poyrazavsever



## Anlaşıllır Yapı = Hızlı Üretim

Bileşenleri katmanlarına göre ayır, isimleri tutarlı kullan.

```
/components
  /atoms
    Button.tsx
  /molecules
    SearchBar.tsx
  /organisms
    Header.tsx
  /templates
    MarketingTemplate.tsx
/pages (Next.js) veya /app
```

in/poyrazavsever



## Küçük Parça = Kolay Test

Atomlar birim test, organizmalar entegrasyon testleriyle güvence altına alınır.



```
// atoms/Button.test.tsx
import { render, screen, fireEvent } from "@testing-library/react";
import { Button } from "./Button";

test("renders and clicks", () => {
  const onClick = vi.fn();
  render(<Button onClick={onClick}>Kaydet</Button>);
  fireEvent.click(screen.getByText("Kaydet"));
  expect(onClick).toHaveBeenCalled();
});
```

in/poyrazavsever

→|

## Sık Yapılan Hatalar

- Atoma iş kuralı koymak ✗
- Molekülü sayfa bağımlı yapmak ✗
- Template içine veri çekmek ✗
- Organizmayı tek sayfaya gömmek ✗

Doğru: İş kuralı sayfa/serviste, sunum atom/molekül/  
organizmada.

**Küçükten Büyüye, Tutarlı Bir UI**  
Projende Atomic Design'ı nereden  
başlatacaksın?



in/poyrazavsever

