



BÖLÜM 1 – Giriş & Temeller

1.1) React Nedir? Tarihçe – Kullanım Alanları – Library vs Framework

1.1.1 React Nedir?

React, modern web uygulamalarında kullanıcı arayüzü (UI) oluşturmak için kullanılan **bileşen tabanlı bir JavaScript kütüphanesidir**.

Özetle:

- Arayüzü küçük, tekrar kullanılabilir **parçalara (component)** bölmemizi sağlar.
- Bu parçalar kendi içinde mantık + HTML görünümünü taşır.
- Veri değiştiğinde React sadece gerekli yerleri günceller. (Virtual DOM sayesinde.)

🧠 Basit Anlatım

React'i **lego parçaları** gibi düşünebilirsin.

- Bir web sitesi komple bir Lego yapıysa,
- React'ta her küçük parça (header, ürün kartı, buton, modal) bir Lego bloğudur.
- Bu blokları istediğin yerde tekrar kullanırsın.

Geleneksel yaklaşım:

- Sayfanın tamamını yönetmek zorunda kalırsın.

React yaklaşımı:

- “Sayfanın sadece değişen küçük kısımlarını yenileyeyim.”

💡 Günlük Hayattan Örnek

Bir telefonun ana ekranını düşün:

- Tüm ekranı değiştirmek yerine,
- Sadece hava durumu widget'ı güncellenir.

React'ın yaptığı tam olarak budur:

Ekranın sadece ihtiyaç duyulan kısmını günceller.

1.1.2 React'in Ortaya Çıkışı (Tarihçe)

2011 – Facebook İç Geliştirme

Facebook, özellikle haber akışındaki (news feed) güncellemeleri yönetmekte zorlanıyordu. Sayfa büyük olduğu için küçük değişikliklerde bile komple sayfa yenileniyor, performans sorunu çıkıyordu.

👉 Bu sorunu çözmek için React'ın ilk versiyonu geliştirildi.

2013 – Açık Kaynak Oldu

Facebook React'i herkese açtı.

O zamanlar pek kabul görmedi çünkü:

- JSX garip görünüyordu ("HTML içinde JavaScript, JavaScript içinde HTML?")
- Farklı bir mantık (component yapısı) vardı.

Ama kısa sürede, performans avantajı ve component yapısı sebebiyle hızla büyüdü.

2018 – Hooks Yayınlandı

React dünyasında bir devrim oldu.

Eskiden:

- class component yazmak zorunluydu.

Hooks ile:

- function component + özel hook mantığı geldi.
- Kod daha sade, daha anlaşılır oldu.

2020–2024 – Gelişmiş Ekosistem

React artık:

- Next.js
 - Remix
 - React Native
- gibi dev platformların temelini oluşturuyor.

Bugün dünyada en çok kullanılan UI teknolojilerinden biri.

1.1.3 React Nerelerde Kullanılır? (Gerçek Proje Örnekleriyle)

React'in gücünü anlamak için kullanım alanlarını detaylandıralım.

A) Web Uygulamaları

React'in en yaygın kullanım alanı.

Örnekler:

- Instagram Web
- Facebook Web
- Netflix kullanıcı arayüzü
- Medium arayüzü

Neden React?

Çünkü hızlı, esnek, tekrar kullanılabilir bileşenlerle büyük arayüzleri kolay yönetir.

Günlük Örnek:

Netflix'te film listesi aşağı kaydırıldığında yeni filmler yüklenir.
React burada sadece listeyi günceller, tüm sayfayı değil.

B) Mobil Uygulamalar (React Native)

React'in öğrettiği mantıkla direkt mobil uygulama yazabilirsin.

- 1 kez mantığı öğren → web yaz
- 2. kez aynı mantıkla → mobil yaz

Örnek:

Instagram, Facebook, Tesla uygulamasında React Native kullanılıyor.

C) Dashboard ve Admin Panelleri

E-ticaret yöneticisi panelleri, muhasebe sistemleri, CRM uygulamaları.

Neden React?

- Komponent yapısı çok ideal.
- Grafik, tablo, filtre vs. gibi karmaşık UI'lar hızlı yapılır.

D) E-Ticaret Siteleri / Ön Yüzü

- Hepsiburada tarzı kart yapıları
- Sepet işlemleri
- Filtreleme sistemleri

React'in Virtual DOM'u sadece ürün kartlarını günceller.

E) Startup Ürünleri

React hızlı geliştirme sağlar →
Startuplar MVP çıkarken en çok React tercih eder.

1.1.4 Library vs Framework (React Neden Library?)

🔍 **React = Library (Kütüphane)**

Bir uygulamanın sadece **UI kısmını çözen**, geri kalan her şeyi geliştiriciye bırakan yapıdır.

React şunları dikte etmez:

- Router kullanacaksın → ister React Router, ister Next.js Router
- State yönetimi → Redux, Zustand, Jotai, useState, Tanstack Query...
- Dosya yapısı → istediğini kullan

React sana esneklik sağlar.

“İstersen böyle yapabilirsin” der.

🔍 **Framework = Kuralları olan, her şeyi yöneten yapı**

Örneğin:

- Angular
- Next.js (artık framework sayılır)
- Vue’nun Nuxt yapısı

Frameworkler şunu der:

- Dosya yapın şöyle olacak.
- Routing şöyle olacak.
- API şu klasörde olacak.
- State şöyle yönetilecek.

Çok kural = Daha az özgürlük ama daha stabil yapı.

📌 Günlük Hayattan Örnek

React → Boş bir ofis düşün.

- Masayı sen seçersin
- Sandalyeyi sen seçersin
- Duvar rengini sen seçersin
- Düzeni sen kurarsın

Framework → Hazır döşenmiş bir ofis.

- Masalar yerleştirilmiş
- Sandalyeler hazır
- Düzen belirli

Sen sadece girer ve çalışırsın.

1.1.5 React Neden Bu Kadar Popüler?

- ✓ Öğrenmesi kolay
- ✓ Komponent yapısı kariyerde çok kritik
- ✓ Şirketlerde en çok aranan skill
- ✓ Ekosistemi inanılmaz büyük
- ✓ Performanslı
- ✓ Mobil + Web tek mantık
- ✓ Freelance işler için çok uygun

1.1.6 React Öğrenirken Bilinmesi Gereken Temel Kavramlar

(İlerleyen bölümlerde detaylı anlatılacak ama burada kısa rehber olsun.)

🧩 Component Mantığı

Sayfayı parçalara ayırma.

🔄 State

Bileşenin verisini yönetme.

🔄 Props

Bir bileşenden diğerine veri aktarma.

🧠 Virtual DOM

React'in hızlı olmasını sağlayan motor.

🪝 Hooks

Function componentlerin süper güçleri.

✓ Bölüm 1.1 Sonu – Mini Özet

Bu bölümde şunları öğrendin:

- React'in ne olduğu
- Neden component tabanlı mimarinin güçlü olduğu
- Tarihçesi
- Kullanım alanları

- Library vs Framework farkları
- React'in neden popüler olduđu