Giriş

World Wide Web (WWW)

World Wide Web (WWW) veya genellikle bilinen adıyla Web, 1989 yılında Tim Berners-Lee tarafından İsviçre'de bulunan CERN laboratuvarlarında bir bilgi paylaşım sistemi olarak tasarlandı. Berners-Lee, farklı bilgisayar sistemlerinde saklanan bilgilerin kolayca erişilebilir ve yönetilebilir hale getirilmesi amacıyla bir sistem geliştirmek istiyordu. Bu amaçla, hiper metin tabanlı bir sistem olan World Wide Web'i oluşturdu. Web, bilgiyi "web sayfaları" aracılığıyla sunar ve bu sayfalar "URL" (Uniform Resource Locator) kullanılarak adreslenebilir. İlk web tarayıcısı ve sunucusu da yine Berners-Lee tarafından geliştirildi. Bu sayede, kullanıcılar metin tabanlı bilgilere hiper bağlantılar üzerinden gezinerek ulaşabilir hale geldi.

Internet

Internet, 1960'ların sonlarında ABD Savunma Bakanlığı'nın desteklediği ARPANET projesiyle başlayan bir girişim olarak ortaya çıktı. Bu proje, bir nükleer saldırı durumunda bile iletişimin sürdürülebilmesi için tasarlanmış bir bilgisayar ağıydı. Zamanla, ARPANET akademik ve araştırma kurumları arasında bilgi paylaşımını kolaylaştıran bir platforma dönüştü ve sonrasında ticari kuruluşların da katılımıyla büyüyerek günümüzün global internet ağına evrildi. Internet, dünya genelinde milyarlarca cihazı birbirine bağlayan ve web sayfaları, e-posta, dosya aktarımı, anlık mesajlaşma gibi çeşitli hizmetleri sunan bir altyapıdır.

Internet ve Web Arasındaki Farklılıklar

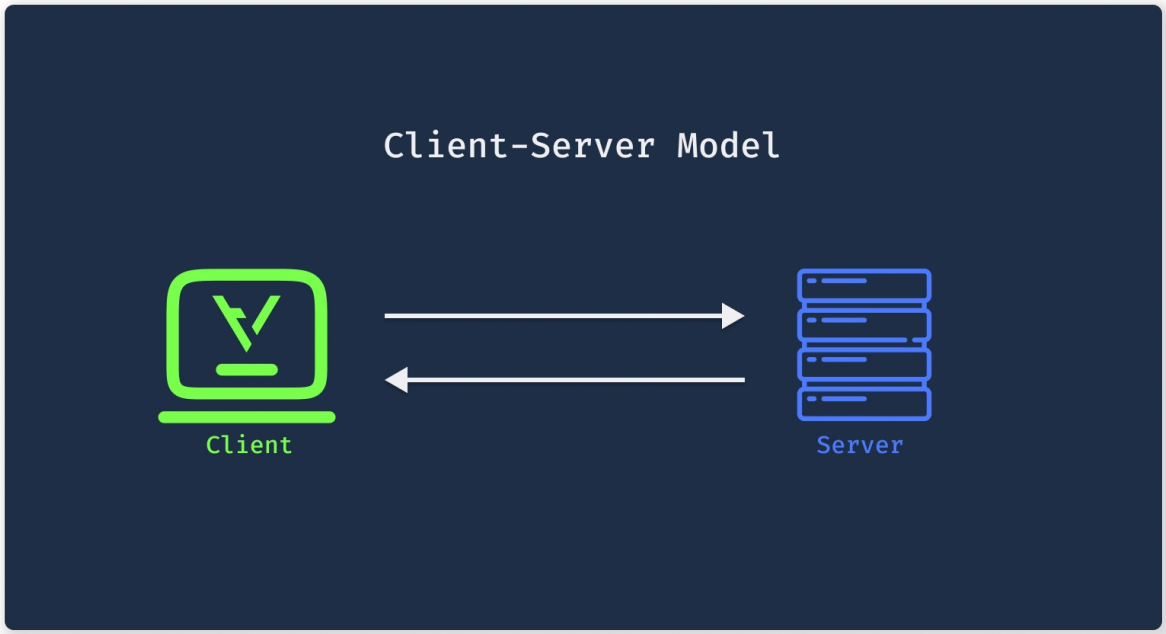
Internet ve Web sıkça karıştırılan iki terimdir, fakat aralarında belirgin farklar bulunmaktadır. Internet, bilgisayarların birbirine bağlanarak veri alışverişi yapabilmesini sağlayan küresel bir ağdır. Web ise, internet üzerinde çalışan, kullanıcıların web tarayıcıları aracılığıyla erişebildiği hiper metin dokümanları ve multimedya içeriklerinden oluşan bir hizmettir. Basitçe ifade etmek gerekirse, internet bir iletişim altyapısıdır; web ise bu altyapı üzerinde çalışan uygulamalardan biridir ve internet olmadan var olamaz.

Web Sitesi

Web siteleri, internetin temel taşlarından biridir ve HTML (HyperText Markup Language), CSS (Cascading Style Sheets), ve JavaScript gibi teknolojiler kullanılarak oluşturulan web sayfalarının bir araya gelmesiyle meydana gelir. Web siteleri, kullanıcıların bilgi edinmesi, alışveriş yapması, sosyal medya etkileşimleri ve daha birçok işlevi gerçekleştirebilmesi için tasarlanmıştır. İlk web sitesi 1991 yılında Tim Berners-Lee tarafından yayımlandı ve World Wide Web Projesi hakkında bilgi veriyordu.
Günümüzde, web siteleri çok daha karmaşık işlevlere ve dinamik içeriklere sahip olup, kullanıcı etkileşimini ve deneyimini zenginleştiren çeşitli teknolojilerle desteklenmektedir.
Modern web, bu tarihsel gelişimin bir sonucu olarak karşımıza çıkar. Berners-Lee'nin vizyonu, bilgiye erişimi demokratikleştirmiş, dünya çapında bilgi alışverişini kolaylaştırmış ve dijital ekonominin gelişimini sağlamıştır. Web'in evrimi, kullanıcıların ihtiyaçlarına yanıt vermek ve teknolojik yenilikleri entegre etmek için devam etmektedir. Bu süreçte, web teknolojileri sürekli olarak gelişirken, webin temelinde yatan amaç ve yapı taşları büyük ölçüde aynı kalmaktadır.

İstemci ve Sunucu Modeli

Web'in temelini oluşturan **istemci** (client) ve **sunucu** (server) modeli, internet üzerindeki bilgi alışverişinin nasıl gerçekleştiğini anlamak için önemlidir. Bu model, web sitelerinin nasıl çalıştığını, verilerin nasıl işlendiğini ve kullanıcıların isteklerine nasıl yanıt verildiğini açıklamaktadır. Bu modelde, bir istemci bir sunucudan (web sitelerini barındıran bilgisayar) bilgi veya hizmet talep eder. Sunucu, bu talebi işler ve gerekli yanıtı istemciye geri gönderir. Bu etkileşim, web'in temelini oluşturur.
İstemci-sunucu modeli, merkezi bir sistemle tüm verilerin tek bir yerde toplanması nedeniyle maliyet açısından büyük bir avantaj sağlar ve kaynaklar yalnızca gerektiğinde kullanılır.



İstemci

İstemciler, web sitelerini görüntüleyebilen ve sunucularla etkileşimde bulunabilen cihazlardır. Genellikle kişisel bilgisayarlar, akıllı telefonlar, tabletler gibi internete bağlanabilen her türlü cihazı kapsar. İstemciler, web tarayıcıları (örneğin, Google Chrome, Mozilla Firefox, Safari) aracılığıyla kullanıcıların web sayfalarıyla etkileşime girmesini sağlar. Kullanıcı, web tarayıcısında bir URL girerek veya bir bağlantıya tıklayarak bir web sayfası talep eder. Bu eylem, sunucuya bir HTTP/HTTPS isteği olarak iletilir.

Sunucu

Sunucu, istemcilerden gelen istekleri kabul eden, işleyen ve gerekli yanıtları (örneğin, bir web sayfası) geri gönderen bir bilgisayar veya yazılım

uygulamasıdır. Web siteleri, internet üzerinde bulunan sunucularda barındırılır.
Web sunucuları, web sayfalarını, resimleri, videoları ve diğer içerikleri barındırır. Apache, Nginx ve Microsoft IIS, yaygın web sunucu yazılımları arasındadır. Bu sunucuların ev veya iş için kullanılan bilgisayarlardan tek farkı, 7/24 çalışabilmek için özel tasarlanmış olmasıdır. İnternete bağlı herhangi bir bilgisayar ile de internet üzerinde web sayfası yayınlanabilmektedir.

İstemci-Sunucu İletişimi

İstemci ve sunucu arasındaki iletişim genellikle HTTP (Hypertext Transfer Protocol) veya HTTPS (HTTP Secure) protokolleri üzerinden gerçekleşir. İstemci, bir web sayfasına erişmek istediğinde, URL'yi içeren bir HTTP GET isteği gönderir. Sunucu, bu isteği alır, gerekli kaynağı bulur ve HTTP yanıtı ile birlikte istemciye geri gönderir.
Örneğin:
-Kullanıcı, tarayıcıda **www.example.com** adresini girer.
-Tarayıcı (istemci), "www.example.com" adresine bir HTTP GET isteği gönderir.
-"example.com" adresinde çalışan web sunucusu, isteği alır ve istenen ana sayfayı bulur.
-Sunucu, ana sayfanın HTML içeriğini içeren bir HTTP yanıtı hazırlar ve istemciye geri gönderir.
-Tarayıcı, gelen HTML içeriğini işler ve kullanıcıya görsel bir web sayfası olarak sunar

Dinamik İçerik Oluşturma

Modern web siteleri genellikle dinamik içerik içerir. Bu, kullanıcının eylemlerine veya veritabanındaki verilere bağlı olarak içeriğin değişebileceği anlamına gelir. Bu tür bir dinamik içerik, sunucu tarafında çalışan programlama dilleri (PHP, Python, Go, JavaScript/Node.js vb.) kullanılarak oluşturulur. Kullanıcı bir istek yaptığında, sunucu tarafında kod

çalışır, gerekli verileri bir veritabanından çeker veya işler ve sonuç olarak dinamik bir web sayfası üretir.

Domain Name System (DNS)

Alan Adı Sistemi (DNS), kullanıcıları World Wide Web'deki web sitelerine, hizmetlere ve kaynaklara bağlamada temel bir rol oynayan İnternet altyapısının kritik bir bileşenidir. Esasen internetin **telefon rehberi**dir ve kullanıcıların kolaylıkla hatırlayabileceği alan adlarını (örneğin, www.example.com) bilgisayarların anlayabileceği IP adreslerine (örneğin, 192.168.1.10) çevirir. Bu çeviri işlemi olmadan, kullanıcıların web sitelerine erişmek için doğrudan IP adreslerini kullanmaları gerekirdi, ki bu da pratik değildir

Tarihçe

DNS'nin gelişimi, internetin isminin ARPANET olduğu internetin ilk günlerinde başlamıştır. 1980'lerin başında ARPANET, ana bilgisayar adlarını IP adresleriyle eşleyen "hosts.txt" adlı merkezi olarak yönetilen bir dosyayı kullanıma sundu. Bilgisayarlar, alan adlarına ait IP adreslerini öğrenebilmek için, bu dosyada sorgu yapıyordu.
Ancak internet kullanıcı sayısının artmasıyla, bilgisayarların birbirlerine bağlanmasını sağlayan IP adreslerini yönetmek ve hatırlamak giderek zorlaşıyordu.
1983 yılında Paul Mockapetris ve Jon Postel, **RFC 882** ve **RFC 883** aracılığıyla bugün bildiğimiz DNS'yi tanıttı ve alan adı çözümlemesi için dağıtılmış ve hiyerarşik bir sistem sağladı. Bu yenilik, modern internetin temelini oluşturan ölçeklenebilir ve verimli DNS mimarisinin önünü açmıştır.

Alan Adı (Domain)

Alan adı (domain), internet üzerindeki belirli bir konumu veya kaynağı temsil eden, insan tarafından okunabilir bir etikettir. Alan adları, noktalarla (noktalar) ayrılmış seviyelerle bir hiyerarşi olarak yapılandırılmıştır.
Örneğin, **www.example.com** alan adı üç bölümden oluşur:
www - alt alan adı (subdomain)

example - ikinci düzey alan adı (second-level domain)
com - üst düzey alan adı (top-level domain)

DNS Yapısı

DNS, hiyerarşik ve dağıtık bir yapıdadır. En üst seviyede, "root" DNS sunucuları yer alır. Bu root sunucular, tüm üst düzey alan adlarını (.com, .net, .org, .edu gibi) ve bu alan adları için sorumlu olan yetkili DNS sunucularının adreslerini içerir. Bir kullanıcı bir web sitesini ziyaret etmek istediğinde, istek önce bir "resolver" aracılığıyla işlenir. Resolver, genellikle internet servis sağlayıcısı (ISP) tarafından sağlanır ve DNS sorgularını çözen bir tür DNS istemcisidir.

DNS Sorgulama Süreci

1-)Yerel DNS Çözümleyici: Kullanıcı bir web sitesini ziyaret etmek istediğinde, tarayıcı önce yerel DNS çözümleyicisine sorgu gönderir.
2-)Root DNS Sunucusu: Eğer yerel çözümleyici cevabı bilmiyorsa, sorgu bir root DNS sunucusuna yönlendirilir. Root sunucu, istenen üst düzey alan adı için yetkili sunucunun adresini döndürür.
3-)Yetkili DNS Sunucusu: Daha sonra sorgu, belirli alan adı için yetkili olan DNS sunucusuna iletilir. Bu sunucu, alan adının IP adresini içerir.
4-)Cevap: Yetkili DNS sunucusu, alan adının karşılık geldiği IP adresini çözümleyiciye gönderir. Çözümleyici, bu IP adresini tarayıcıya iletir ve tarayıcı sonunda kullanıcıyı hedef web sitesine yönlendirir.

Kayıt Türü	Açıklama
A	Bir alan adını bir IPv4 adresine çevirir.
AAAA	Bir alan adını bir IPv6 adresine çevirir.
CNAME	Bir alan adını başka bir alan adına yönlendirir (Canonical Name).
MX	Bir alan adı için e-posta sunucusunun adresini belirtir (Mail Exchange).
NS	Bir alan adı için ad sunucularını (name servers) belirtir, yani DNS kayıtlarının nerede olduğunu gösterir.

PTR	Bir IP adresini alan adına çevirir (ters DNS lookup için kullanılır).
SOA	DNS bölgesi hakkında genel bilgi verir (Start of Authority) ve DNS bölgesinin yetkili olduğu alan ile ilgili yönetimsel bilgiler içerir.
TXT	Alan adı ile ilişkili herhangi bir metni saklamak için kullanılır, genellikle doğrulama ve politika ifadeleri (örneğin, SPF kayıtları) için kullanılır.
SRV	Belirli bir hizmetin, protokolün ve domain'in hangi sunucular üzerinde çalıştığını tanımlar. Örneğin, SIP hizmetleri veya LDAP dizin erişimi için kullanılır.
SPF	Bir alan adından e-postaların hangi sunucular aracılığıyla gönderilebileceğini tanımlar, e-posta spoofing ve spam'i önlemek için kullanılır.

HTTPS

HTTPS (Hypertext Transfer Protocol Secure), internet üzerinden güvenli bilgi alışverişini sağlayan bir protokoldür. İlk olarak Netscape Communications tarafından 1994 yılında SSL protokolü ile birlikte tanıtıldı. Başlangıçta, özellikle çevrimiçi işlemlerde güvenliği artırmak amacıyla kullanılmaya başlandı. Zamanla, veri güvenliğinin öneminin artmasıyla birlikte, HTTPS kullanımı genişledi ve günümüzde web sitelerinin büyük bir çoğunluğu tarafından kullanılır hale geldi.
Temel olarak HTTP protokolünün, TLS (Transport Layer Security) veya SSL (Secure Sockets Layer) protokolleri ile güçlendirilmiş versiyonudur. HTTPS, kullanıcıların ve sunucuların arasındaki veri aktarımını şifreleyerek, üçüncü şahısların bu verilere erişimini veya verileri değiştirmesini önler.

SSL (Secure Sockets Layer): İnternet üzerindeki veri iletişiminin şifrelenerek güvenli bir şekilde gerçekleşmesini sağlayan bir protokoldür.

TLS (Transport Layer Security): SSL'in geliştirilmiş versiyonu olup, internet üzerinden güvenli iletişim sağlamak için daha güçlü şifreleme teknikleri ve güvenlik protokolleri sunar.

HTTPS İşleyişi

HTTPS, istemci ve sunucu arasında bir şifreleme katmanı ekleyerek çalışır. Bir kullanıcı HTTPS ile korunan bir web sitesini ziyaret ettiğinde, tarayıcı ve sunucu arasında bir "handshake" işlemi gerçekleşir. Bu işlem sırasında:

1-Sunucu Kimliğini Doğrulama: Sunucu, kendisine ait bir güvenlik sertifikası ve genel anahtar içeren bir sertifika sunar. Tarayıcı, sertifikanın güvenilir bir otorite tarafından imzalandığını doğrular.

2-Oturum Anahtarının Oluşturulması: Tarayıcı ve sunucu, iletişim için kullanılacak geçici bir oturum anahtarı oluşturmak üzere güvenli bir yöntem kullanır.

3-Veri Şifrelemesi: Oluşturulan oturum anahtarı, veri aktarımının şifrelenmesi için kullanılır. Bu sayede, verilerin güvenli bir şekilde iletilmesi sağlanır.

HTTP Yanıtları (Response)

1xx: Bilgilendirici Yanıtlar

Durum Kodu	Durum Adı	Açıklama
100	Continue	İstemci isteğini sürdürebilir.
101	Switching Protocols	Sunucu, istemcinin protokol değişim talebini kabul etti.
102	Processing	İstek işleniyor, ancak henüz tamamlanmadı (WebDAV).
103	Early Hints	Sunucunun istemciye erken başlık bilgileri gönderebileceği anlamına gelir.

2xx: Başarılı Yanıtlar

Durum Kodu	Durum Adı	Açıklama
200	OK	İstek başarılı.
201	Created	Kaynak başarıyla oluşturuldu.
202	Accepted	İstek kabul edildi, ancak işlenmedi.
204	No Content	İstek başarılı; döndürülecek içerik yok.
205	Reset Content	İstek başarılı; istemcinin belge görünümünü sıfırlaması gerekir.

3xx: Yönlendirme

Durum Kodu	Durum Adı	Açıklama
301	Moved Permanently	Kaynak kalıcı olarak başka bir URI'ya taşındı.
302	Found	Kaynak geçici olarak başka bir URI'ya taşındı.
303	See Other	Yanıt başka bir URI'da bulunabilir (GET metodunu kullanarak).
304	Not Modified	Kaynak değişmedi; önbellekteki sürüm kullanılabilir.
307	Temporary Redirect	Kaynak geçici olarak başka bir URI'ya taşındı; yöntem değişmedi.

5xx: Sunucu Hatası

Durum Kodu	Durum Adı	Açıklama
500	Internal Server Error	Sunucu tarafında genel bir hata meydana geldi.
501	Not Implemented	Sunucu isteği desteklemiyor veya tanımıyor.
502	Bad Gateway	Ara sunucu yanıt alamadı.
503	Service Unavailable	Sunucu geçici olarak hizmet dışı veya aşırı yüklenmiş.
504	Gateway Timeout	Ara sunucu, başka bir sunucudan zamanında yanıt alamadı.

Adım Adım Çalışma Prensibi

Tüm bu öğrendiklerimizi özetlemek için, bir web sitesine erişmek istediğimizde gerçekleşen adımların ne olduğuna bakalım:

1. Web Tarayıcısını Açma ve URL Girme:

İnternette gezinmeye başlamak için ilk adım, bir web tarayıcısı açarak ziyaret etmek istediğiniz web sitesinin adresini, yani URL'ini (Uniform Resource Locator) girerek başlar. Örneğin: https://google.com

2. DNS Sorgusu:

Girdiğiniz URL, insanlar için anlamlı ve hatırlaması kolay bir adrestir; fakat internet ağları ve bilgisayarlar bu adresler yerine sayısal IP adreslerini kullanır. Bu nedenle, tarayıcınız URL'yi, web sitesinin barındırıldığı sunucunun IP adresine çevirmek için bir Domain Name System (DNS) sorgusu yapar. DNS, internetin telefon rehberi gibidir; girdiğiniz web sitesi adını alır ve bunu karşılık gelen IP adresine dönüştürür.

3. Sunucuya İstek Gönderme:

DNS sorgusunun sonucunda elde edilen IP adresi ile tarayıcınız, web sitesinin barındırıldığı sunucuya bir HTTP isteği gönderir. Bu istek, sunucudan belirli bir sayfanın içeriğini talep eder.

4. Sunucudan Yanıt Alınması:
Web sunucusu, tarayıcınızın isteğini alır ve gerekli işlemleri yaparak istenilen web sayfasının içeriğini bir HTTP yanıtı ile geri gönderir. Bu yanıt, genellikle HTML (Hypertext Markup Language) kodlarından oluşur ve sayfanın nasıl görüneceğini tanımlar.

5. Web Sayfasının Görüntülenmesi:
Tarayıcınız, sunucudan aldığı HTML içeriğini işler ve kullanıcıya görsel bir web sayfası olarak sunar. Bu aşamada, sayfanın düzgün bir şekilde görüntülenebilmesi için ek kaynakların (resimler, CSS dosyaları, JavaScript dosyaları vb.) de sunucudan istenmesi ve alınması gerekebilir.

6. Bağlantının Sonlandırılması:
Web sayfası başarıyla yüklendikten sonra, HTTP/1.1 protokolünde keep-alive özelliği olmadığı takdirde, istemci ve sunucu arasındaki bağlantı otomatik olarak sonlandırılır. Eğer kullanıcı sayfa içinde başka bir linke tıklar veya yeni bir sayfa isteği yaparsa, bu süreç yeniden başlar ve yeni bir bağlantı kurulur.

Frontend ve Backend

Web geliştirme, internet üzerindeki sitelerin ve uygulamaların oluşturulması ve yönetilmesi sürecidir. Bu süreç, iki ana bileşene ayrılır: Frontend (İstemci tarafı) ve Backend (Sunucu tarafı). Her iki alan da web sitelerinin işlevsel ve estetik açıdan nasıl geliştirildiğini belirler.

Frontend

Frontend, kullanıcıların doğrudan etkileşime girdiği web sayfasının görünüm ve hissini oluşturmaktan sorumludur. HTML, CSS ve JavaScript gibi teknolojiler kullanılarak gerçekleştirilir.

Temel Teknolojiler

- **HTML (HyperText Markup Language)**: Web'in temel yapı taşıdır ve bir web sayfasının içeriğini ve yapısını tanımlar. HTML,

<tag>ler kullanılarak belirlenen bir dizi markup elementi içerir. Bu elementler, metinleri başlıklar, paragraflar, listeler olarak düzenlemeyi; resimler, videolar ve bağlantılar gibi multimedya içeriklerini eklemeyi; form elementleri aracılığıyla kullanıcı girdisini toplamayı sağlar.

CSS (Cascading Style Sheets): Web sayfalarının nasıl görüneceğini tanımlar. HTML ile oluşturulan yapıya stil ekleyerek sayfanın görsel tasarımını ve düzenini belirler. CSS, ayrı ayrı elementlere, belirli bir ID veya sınıfa sahip elementlere veya belirli bir HTML tipine sahip tüm elementlere stil uygulayabilir. Renkler, fontlar, aralıklar gibi görsel öğeleri kontrol eder.

JavaScript: Web sayfalarını dinamik ve interaktif hale getiren bir programlama dilidir. Kullanıcı etkileşimlerine yanıt verebilir, veri alışverişinde bulunabilir ve sayfa içeriğini programlı bir şekilde değiştirebilir. Dinamik içeriklerin yönetimi JavaScript ile yapılır.

Frontend Framework ve Kütüphaneleri

Geliştiriciler, süreci hızlandırmak ve daha gelişmiş özellikler eklemek için Angular, React, Vue.js gibi JavaScript frameworklerini ve kütüphanelerini kullanır.

React: Facebook tarafından geliştirilen bir JavaScript kütüphanesidir. Kullanıcı arayüzleri inşa etmek için kullanılır. Tek Sayfa Uygulamaları (SPA) geliştirmek için popüler bir seçenektir. Reusable (yeniden kullanılabilir) komponentler üzerine kuruludur.

Angular: Google tarafından geliştirilen bir JavaScript framework'üdür. Dinamik web uygulamaları oluşturmak için kullanılır. Model-View-Controller (MVC) mimarisiyle çalışır ve geniş çaplı uygulamalar için uygun bir yapı sunar.

Vue.js: Hafif ve esnek bir JavaScript framework'üdür. Küçük ve orta ölçekli projeler için idealdir. Kolay öğrenilir ve uygulanır; detaylı dokümantasyona sahiptir.

Backend

Backend, bir web sitesinin "arka planında" çalışan ve veritabanı işlemleri, kullanıcı yönetimi, sunucu mantığı gibi işlevleri yöneten bölümdür. Backend, kullanıcı isteklerini alır, gerekli işlemleri yapar (veritabanı sorgulamaları, hesaplamalar vb.) ve sonuçları frontend'e gönderir.

Temel Teknolojiler

Sunucu Dilleri: Go, Node.js, Python, Ruby, PHP gibi programlama dilleri ve kendi içerisinde bulunan frameworkler kullanılır.

Veritabanı Yönetim Sistemleri: MySQL, PostgreSQL, MongoDB gibi sistemler verilerin saklanması ve yönetilmesi için kullanılır.

Sunucu Altyapısı: Apache, Nginx gibi web sunucuları, istemcilerden gelen isteklere yanıt verir.

Backend Dilleri ve Frameworkler

Go (Golang)

Google tarafından geliştirilen Go, açık kaynaklı, statik olarak tiplemeli, derlenmiş ve eşzamanlılık (concurrency) destekleyen bir programlama dilidir. Yüksek performanslı ve ölçeklenebilir web uygulamaları, mikroservisler ve veri işleme görevleri için idealdir. Go, basit, hızlı ve güvenilir yazılım geliştirmeyi hedefler.

Go HTTP Web Kütüphanesi: Go'nun standart kütüphanesi içerisinde, HTTP sunucuları ve istemcileri oluşturmak için güçlü araçlar sunan bir HTTP paketi bulunur. Bu paket, Go'nun web uygulamaları geliştirmek için doğrudan kullanılabilecek kapsamlı bir çözüm sunduğunu gösterir.

Fiber Framework: Go için geliştirilmiş, Express.js'ten esinlenmiş minimalist bir web frameworkü olan Fiber, yüksek performans ve kullanım kolaylığı ile öne çıkar. Fiber, Go'nun hızından ve düşük bellek kullanımından faydalanırken, geliştiricilere daha tanıdık ve kolay bir API sunar.

Echo Framework: Go için yüksek performanslı ve minimalist bir web frameworküdür. RESTful API'lerin hızlı ve kolay bir şekilde geliştirilmesine olanak tanır. Echo, basitlik ve performansı bir arada sunarak Go geliştiricileri arasında popüler bir seçenektir.

Node.js

JavaScript'i sunucu tarafında çalıştırmak için kullanılan Node.js, geleneksel olarak yalnızca tarayıcıda çalışan JavaScript'in, backend geliştirmede de kullanılmasını sağlar. Olay tabanlı, non-blocking (engelsiz) I/O modeli sayesinde hafif ve verimlidir.

Python ve Django/Flask

Python: Okunabilirliği ve esnekliği ile popüler bir programlama dilidir. Web geliştirme, veri bilimi, yapay zeka gibi birçok alanda kullanılır.

Django Framework: Büyük ölçekli web uygulamaları geliştirmek için kullanılan yüksek seviyeli bir Python web framework'üdür. Güçlü ve hızlı geliştirme imkanı sunar.

Flask Framework: Daha minimalist bir yaklaşım sunan hafif bir Python web framework'üdür. Küçük ve orta ölçekli projeler için uygundur.

Ruby on Rails

Ruby programlama dilini kullanarak web uygulamaları geliştirmek için tasarlanmış bir framework'tür. Convention over Configuration (Yapılandırmadan ziyade Anlaşma) ve Don't Repeat Yourself (Kendini Tekrar Etme) prensipleriyle kod tekrarını minimize eder ve geliştirme sürecini hızlandırır.

PHP ve Laravel**PHP**: dinamik web içeriği oluşturmak için yaygın olarak kullanılan bir server-side scripting dilidir. WordPress gibi popüler içerik yönetim sistemlerinin temelini oluşturur.

Laravel Framework: PHP için modern, hızlı ve güvenli bir web framework'üdür. MVC mimarisiyle çalışır ve gelişmiş özellikler sunar.

Web Geliştirme Süreci

Web geliştirme, teknolojik yeniliklere ve kullanıcı ihtiyaçlarına hızlı bir şekilde adapte olabilen, dinamik bir alandır. Frontend ve backend geliştirme, web

sitelerinin ve uygulamalarının başarılı bir şekilde oluşturulması ve yönetilmesi için birlikte çalışır.

Application Programming Interface (API)

Application Programming Interface (API), yazılımlar arasında iletişim kurmak için kullanılan bir arayüzdür. API'ler, farklı yazılım bileşenlerinin birbirleriyle etkileşimde bulunabilmesi için standart protokoller ve araçlar sağlar. Backend geliştirmede, API'ler genellikle veritabanı işlemleri, kullanıcı yönetimi ve diğer sunucu tarafı görevleri için frontend ile sunucu arasındaki köprü görevi görür.

API Çeşitleri

REST (Representational State Transfer)
REST, web üzerindeki sistemler arasında iletişim kurmak için kullanılan bir API tasarım modelidir. HTTP protokolünü kullanarak çalışır ve kaynak tabanlı bir yaklaşım sunar. REST API'ler, CRUD (Create, Read, Update, Delete) işlemlerini destekler ve genellikle JSON veya XML formatında veri alışverişi yapar.
Örnek REST API isteği ve uç noktası: ID değeri "123" olan kullanıcının bilgilerini getirmektedir.

```
GET /users/123 HTTP/1.1
Host: api.example.com
```

Uç nokta: https://api.example.com/users/123

SOAP (Simple Object Access Protocol)
SOAP, HTTP ve XML tabanlı bir protokol kullanarak ağ üzerindeki nesneler arasında iletişim kurmak için tasarlanmıştır. Güvenlik, işlem yönetimi gibi özellikler sunar ve genellikle kurumsal uygulamalarda tercih edilir.
Örnek SOAP isteği ve uç noktası:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ser="http://example.com/service">
   <soapenv:Header/>
   <soapenv:Body>
      <ser:GetUser>
         <userId>123</userId>
      </ser:GetUser>
   </soapenv:Body>
</soapenv:Envelope>
```

Uç nokta: https://api.example.com/soap

GraphQL

GraphQL, Facebook tarafından geliştirilmiş bir sorgulama dilidir ve bir API tasarımıdır. Kullanıcılara, istedikleri verileri tek bir istekle alabilme yeteneği sunar. Bu, veri alışverişinin daha verimli olmasını sağlar ve aşırı/eksik fetching sorunlarını azaltır.

Örnek GraphQL sorgusu ve uç noktası:

```
query {
  user(id: "123") {
    name
    email
  }
}
```

Uç nokta: https://api.example.com/graphql

WebSocket

WebSocket, istemci ve sunucu arasında iki yönlü, sürekli bir bağlantı sağlayan bir protokoldür. Gerçek zamanlı uygulamalar için idealdir çünkü HTTP'nin aç-kapa bağlantı modeline kıyasla daha az gecikme süresi sunar.

WebSocket bağlantısı için JavaScript örneği:

```
const socket = new WebSocket('ws://api.example.com/socket');

socket.onopen = function(event) {
    console.log('WebSocket connection successful.');

    socket.send('Hello server!');
};

socket.onmessage = function(event) {
    console.log('Message from the server: ', event.data);
```

```
};

socket.onclose = function(event) {
    console.log('WebSocket connection closed.');
};

socket.onerror = function(error) {
    console.error('WebSocket error: ', error);
};
```

Uç nokta:

ws://api.example.com/socket

API Kullanım Alanları ve Örnek API'ler

Aşağıda, çeşitli kullanım alanlarına ve bu alanlarda kullanılan popüler API'lere yer verilmiştir.

Sosyal Medya Entegrasyonu

Sosyal Medya Entegrasyonu

API	Kullanım Alanı
Facebook Graph API	Facebook verilerine erişim sağlar, gönderileri, yorumları yönetebilir ve Facebook analitik verilerini alabilir.
Twitter API	Tweet gönderme, silme, kullanıcı bilgilerini çekme ve tweet akışını analiz etme gibi işlevleri destekler.
Instagram Graph API	Instagram hesapları üzerindeki etkileşimleri yönetir, medya yayınlama ve hikaye oluşturma işlemlerini yapabilir.

Harita ve Konum Servisleri

API	Kullanım Alanı
Google Maps API	Haritalar ekler, konum bazlı aramalar yapar ve yol tarifleri sağlar.
OpenStreetMap API	Ücretsiz harita verilerine erişim sağlar ve harita üzerinde özelleştirmeler yapılmasına olanak tanır.
Mapbox API	Zengin harita özellikleri ve özelleştirilebilir harita çözümleri sunar.

Hava Durumu Bilgileri

API	Kullanım Alanı
OpenWeather API	Güncel hava durumu, tahminler ve tarihsel hava verilerine erişim sağlar.
Weatherstack API	Hava durumu bilgilerini gerçek zamanlı olarak sorgular ve çeşitli hava durumu verileri sunar.
AccuWeather API	Kapsamlı hava durumu verileri ve tahminleri sunar, global erişime sahiptir.

API Güvenliği

API güvenliği, uygulamanın güvenliğinin sağlanmasında kritik bir rol oynar. Yetkilendirme protokolleri (OAuth, API anahtarları vb.), veri şifreleme ve sınırlama oranları (rate limiting), API'leri kötü niyetli kullanımlara karşı korumak için kullanılan yöntemler arasındadır.

Veri Tabanı Sunucusu

Veri tabanı sunucusu, veri tabanı yönetim sistemlerini (DBMS) barındıran ve kullanıcılara veya uygulamalara veri depolama, sorgulama, güncelleme ve yönetim hizmetleri sağlayan bir sunucudur. Veri tabanı sunucuları, web uygulamalarından büyük kurumsal sistemlere kadar geniş bir yelpazede kullanılır.

Veri tabanı sunucularının tarihçesi, 1960'lara, bilgisayar teknolojilerinin ve veri depolama ihtiyaçlarının hızla gelişmeye başladığı döneme dayanır. İlk veri tabanı yönetim sistemleri (DBMS) basit ve düz dosya tabanlı sistemlerken, zamanla ilişkisel veri tabanı yönetim sistemleri (RDBMS) ve

NoSQL veri tabanları gibi daha karmaşık ve ölçeklenebilir sistemler geliştirilmiştir.

Veri tabanı sunucusu yönetimi (DMBS): Veri tabanı performansını optimize etme, veri bütünlüğünü koruma, yedekleme ve felaketten kurtarma operasyonlarını içerir. Bu yönetim süreci, genellikle veri tabanı yöneticileri (DBA) tarafından gerçekleştirilir.

Temel İşlevler

- **Veri Depolama ve İyileştirme**: Verilerin etkili bir şekilde saklanması ve sorguların hızlı bir şekilde işlenmesi için yapılandırılır.
- **Güvenlik**: Yetkilendirme, kimlik doğrulama ve veri şifreleme ile veri güvenliğini sağlar.
- **Yedekleme ve Kurtarma**: Veri kaybını önlemek için düzenli yedeklemeler yapar ve gerektiğinde verileri kurtarır.

Popüler Veri Tabanı Sunucuları

Her bir veri tabanı sunucusunun benzersiz özellikleri, kullanım alanlarına ve projenin gereksinimlerine göre seçimi etkiler. SQL ve NoSQL veri tabanları arasındaki seçim, uygulamanın veri modeli, ölçeklenebilirlik ihtiyaçları ve geliştirme ekibinin uzmanlığı gibi faktörlere bağlı olarak değişiklik gösterir.

SQL (Structured Query Language) Veri Tabanları

SQL (Structured Query Language) veri tabanları, verileri tablolar halinde düzenleyen ilişkisel bir model kullanır. Her tablo, benzer veri türlerini depolayan sütunlara (veya alanlara) bölünmüştür ve tablolar arası ilişkiler, veri bütünlüğünü ve tutarlılığını sağlar.

Özellikler:

- **ACID Uyumluluğu**: Atomiklik, Tutarsızlık, İzolasyon ve Dayanıklılık ilkelerini destekler, bu da veri tabanı işlemlerinin güvenilirliğini ve tutarlılığını artırır.
- **Sıkı Şema**: Veri tabanı yapısının önceden tanımlanması gerekir ve veriler bu yapıya uygun olarak saklanır.
- **Sorgulama Gücü**: SQL dili, karmaşık sorgular ve veri analizi için güçlü bir araçtır.

Popüler SQL Veri Tabanları: MySQL, PostgreSQL, Oracle, Microsoft SQL Server

MySQL

MySQL, LAMP (Linux, Apache, MySQL, PHP/Python/Perl) yığınının bir parçası olarak popülerdir. Replication, sharding ve yüksek erişilebilirlik sunar.

- **Port Numarası**: 3306
- **Örnek Sorgu**: SELECT * FROM users WHERE age > 18;
- **Kullanım Alanları**: Web siteleri, forumlar, CMS sistemleri ve küçük-orta ölçekli uygulamalar.

PostgreSQL

Nesne-ilişkisel bir DBMS olup, gelişmiş özellikler (örneğin, kompleks sorgular, dış anahtarlar, transactional integrity, multiversion concurrency control) sunar. Genişletilebilirlik ve uyumluluk özellikleri ile bilinir.

- **Port Numarası**: 5432
- **Örnek Sorgu**: INSERT INTO books (name, author) VALUES ('Animal Farm', 'George Orwell');
- **Kullanım Alanları**: Finansal işlemler, büyük ölçekli CRM ve ERP sistemleri, coğrafi bilgi sistemleri.

Oracle

Kapsamlı özellikleri, yüksek güvenilirlik ve genişletilebilirlik seçenekleri ile dünya çapında kurumsal düzeyde tercih edilen bir ilişkisel veritabanı yönetim sistemidir (RDBMS). Oracle, özellikle büyük ölçekli işletmelerin ve kritik iş uygulamalarının ihtiyaç duyduğu performans, güvenilirlik ve güvenlik standartlarını karşılamak üzere tasarlanmıştır.

- **Port Numarası**: 1521
- **Örnek Sorgu**: UPDATE employees SET salary = salary * 1.05 WHERE department_id = 10;
- **Kullanım Alanları**: Finansal sistemler, ERP ve CRM sistemleri, e-ticaret, veri ambarları ve büyük veri analizi.

Microsoft SQL Server

Güçlü analiz araçları, kapsamlı güvenlik özellikleri ve kullanıcı dostu yönetim araçları ile kurumsal düzeyde tercih edilir. Özellikle .NET ekosistemiyle entegrasyon için tasarlanmıştır.

- **Port Numarası**: 1433
- **Örnek Sorgu**: UPDATE orders SET status = 'delivered' WHERE order_no = 100;
- **Kullanım Alanları**: .NET tabanlı uygulamalar, büyük veri depolama, iş zekası ve analiz uygulamaları.

NoSQL Veri Tabanları

NoSQL veri tabanları, ilişkisel olmayan, esnek şemalı veri modelleri sunar. Bu veri tabanları, büyük veri setlerinin ve yapılandırılmamış veya yarı yapılandırılmış verilerin depolanması ve yönetilmesi için daha uygun olabilir.

Özellikler:

- **Şema Esnekliği**: Verilerin önceden tanımlanmış bir şemaya uyması gerekmez, bu da uygulama geliştirmeyi hızlandırabilir.
- **Ölçeklenebilirlik**: Yatay ölçeklenebilirlik (diğer sunuculara genişletme) ile büyük veri hacimlerini ve yüksek trafikli uygulamaları destekler.
- **Veri Modelleri**: Anahtar-değer çiftleri, belgeler, grafikler veya sütunlar gibi çeşitli veri modellerini destekler.

Popüler NoSQL Veri Tabanları: MongoDB (Belge tabanlı), Cassandra (Sütun tabanlı), Redis (Anahtar-değer tabanlı), Neo4j (Grafik tabanlı)

MongoDB

JSON benzeri formatındaki belgelerle çalışır. Schemaless yapısı, uygulama geliştirme sürecini hızlandırır ve esneklik sağlar.

- **Port Numarası**: 27017
- **Örnek Sorgu**: `db.users.find({name: "John"});`
- **Kullanım Alanları**: Büyük ölçekli veri setleri, doküman odaklı depolama gerektiren uygulamalar, gerçek zamanlı analitik.

Cassandra

Lineer ölçeklenebilirlik ve yüksek erişilebilirlik sunan sütun tabanlı bir depolama sistemidir. Hata toleransı yüksektir.

- **Port Numarası**: CQL (Cassandra Query Language) için 9042
- **Örnek Sorgu**: `SELECT * FROM user_status WHERE status = 'active';`
- **Kullanım Alanları**: Büyük ölçekli veri yazma ve okuma işlemleri, zaman serisi verileri, dağıtık sistemler.

Redis

Anahtar-değer (key-value) tabanlı bir veri tabanıdır. Verileri RAM'de saklayarak yüksek performans sunar, disk tabanlı depolama seçeneği de mevcuttur.

- **Port Numarası**: 6379
- **Örnek Sorgu**: `SET session_536fb1022357d8c410 { "id":"d32d5b69-03a3-4783-a519-b55bdaea5e88", "name": "John", "role": "admin" }`

- **Kullanım Alanları**: Oturum yönetimi, önbellekleme, mesaj kuyrukları, gerçek zamanlı uygulamalar.

SQL ve NoSQL Arasındaki Temel Farklar

- **Veri Yapısı**: SQL sıkı bir şemaya sahipken, NoSQL daha esnek veri modelleri sunar.
- **Ölçeklenebilirlik**: SQL genellikle dikey ölçeklenebilirlik sunarken, NoSQL çözümleri yatay ölçeklenebilirliği kolaylaştırır.
- **İşlem Güvenilirliği**: SQL veritabanları ACID prensiplerine sıkı sıkıya bağlıdır, NoSQL veritabanları ise genellikle eventual consistency modelini benimser.

Veri tabanı Sunucusu Kurulumu ve Kullanımı

Veri tabanı sunucusu kurulumu, genellikle sunucu donanımının veya bulut hizmetlerinin konfigürasyonunu, DBMS yazılımının yüklenmesini ve başlangıç yapılandırmasını içerir. Veri tabanı sunucusu kullanımı, SQL veya NoSQL sorgulama dilleri aracılığıyla veri ekleme, güncelleme, silme ve sorgulama işlemlerini kapsar.

Web Sunucusu

Web sunucusu, internet üzerinden kullanıcıların web tarayıcıları aracılığıyla erişebileceği web sayfalarını ve diğer içerikleri barındıran ve dağıtan bir sunucu türüdür. Temel işlevi, HTTP protokolü üzerinden gelen istekleri işlemek ve bu isteklere HTML sayfaları, görüntüler, stil dosyaları (CSS) ve JavaScript dosyaları gibi içeriklerle yanıt vermek olan web sunucuları, internetin temel taşlarından biridir.
İlk web sunucusu, Tim Berners-Lee tarafından CERN'de 1990 yılında geliştirildi ve World Wide Web projesinin bir parçasıydı. Bu ilk web sunucusu, basit HTML dokümanlarını sunacak şekilde tasarlanmıştı. Zamanla, web sunucuları daha karmaşık web uygulamalarını ve dinamik içerikleri destekleyecek şekilde evrildi.

Web Sunucusu Yazılımları

Modern web sunucuları, statik (değişmeyen) içeriklerin yanı sıra dinamik içerikleri de işleyebilir. Bu işlevsellik, çeşitli web sunucu yazılımları aracılığıyla sağlanır:

Web Sunucusu Konfigürasyonu: Web sunucusu yazılımları, genellikle yapılandırma dosyaları aracılığıyla özelleştirilir. Bu yapılandırma dosyalarında, sunucunun hangi portu dinleyeceği, hangi dizindeki dosyaların sunulacağı, güvenlik ayarları ve erişim kontrol kuralları gibi önemli ayarlar bulunur.

Apache HTTP Server

Apache, açık kaynaklı ve platformlar arası çalışabilen bir web sunucusudur. Esnek yapılandırma seçenekleri ve geniş modül desteğiyle öne çıkar.

Özellikleri:

- Modüler yapı, dinamik yüklemeyi destekler.
- .htaccess dosyası üzerinden dizin bazında yapılandırma sağlar.
- SSL/TLS desteği ile güvenli bağlantılar kurulabilir.

Yapılandırma Örneği:

httpd.conf dosyasına basit bir Virtual Host eklemek.

```
<VirtualHost *:80>
    ServerName www.example.com
    DocumentRoot "/www/example"
    <Directory "/www/example">
        AllowOverride All
        Require all granted
    </Directory>
</VirtualHost>
```

Nginx

Yüksek performanslı bir web sunucusu, ters proxy ve e-posta proxy'si olarak işlev görür. Asenkron olay yönetimi ile yüksek eşzamanlı bağlantıları destekler.

Özellikleri:

- Düşük bellek kullanımı ve yüksek eşzamanlılık.
- Yük dengeleme ve ters proxy özellikleri.
- HTTP2, WebSocket ve SSL/TLS desteği.

Yapılandırma Örneği: Basit bir ters proxy yapılandırması.

```
server {
    listen 80;
    server_name example.com;
```

```
    location / {
        proxy_pass http://localhost:3000;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

Microsoft IIS (Internet Information Services)

Windows sunucular için bir web sunucusu ve bir dizi internet hizmeti sunar. ASP.NET uygulamaları için derin entegrasyon sağlar.

Özellikleri:

- Grafik kullanıcı arayüzü üzerinden kolay yönetim.
- Güçlü güvenlik özellikleri ve kimlik doğrulama mekanizmaları.
- ASP.NET, Windows Authentication, URL Rewrite gibi modüllerle zenginleştirilmiş.

Yapılandırma Örneği: IIS, genellikle grafik arayüz üzerinden yönetilir, ancak komut satırı araçları veya PowerShell script'leri kullanılarak da yapılandırılabilir. Bir site eklemek için appcmd kullanımı:

```
appcmd add site /name:example /bindings:http/*:80:example.com
/physicalPath:C:\inetpub\example
```

Tomcat

Java servletlerini ve JSP sayfalarını destekleyen bir web konteyneridir. Genellikle Java web uygulamalarını çalıştırmak için kullanılır.

Özellikleri:

- Hafif ve modüler yapı.
- Java EE özelliklerinin bir kısmını destekler.
- Java servlet ve JSP teknolojileri için tam destek.

Yapılandırma Örneği:

server.xml dosyasında bir connector yapılandırması.

```
<Connector port="8080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

Her bir web sunucusu, farklı ihtiyaçlara ve teknolojilere odaklanır. Apache ve Nginx, statik ve dinamik içerik sunumu için geniş kullanım alanları sunarken, IIS .NET ekosistemiyle, Tomcat ise Java tabanlı uygulamalarla yakın entegrasyon sağlar. Bu örnekler, her sunucunun temel yapılandırma yaklaşımlarını göstermektedir ve gerçek dünya uygulamalarında daha karmaşık yapılandırmalar gerekebilir.

İstek-Yanıt Döngüsü

1. **İstek**: Kullanıcı, web tarayıcısı aracılığıyla bir URL girer. Tarayıcı, bu URL'yi temsil eden bir HTTP GET isteği oluşturur ve ilgili web sunucusuna gönderir.
2. **İşleme**: Web sunucusu isteği alır ve URL'yi analiz eder. İstenen kaynağın bir HTML dosyası, resim, CSS veya JavaScript dosyası olup olmadığını belirler.
3. **Yanıt**: Sunucu, istenen kaynağı bulur ve HTTP yanıtı ile birlikte kullanıcıya (tarayıcıya) geri gönderir. Eğer kaynak bulunamazsa, bir hata mesajı (örneğin, 404 Not Found) döndürülür.

Güvenlik ve Performans

Web sunucuları, DDoS saldırıları, SQL enjeksiyonu ve XSS gibi güvenlik tehditlerine karşı savunmasız olabilir. Bu nedenle, güvenlik duvarları, SSL/TLS şifrelemesi ve düzenli yazılım güncellemeleri gibi güvenlik önlemlerinin uygulanması kritiktir. Ayrıca, yüksek trafik durumlarında performansı optimize etmek için caching, yük dengeleme ve içerik dağıtım ağları (CDN) gibi teknikler kullanılır.

Güvenlik ve Performans

Modern web uygulamalarının iki temel önceliği güvenlik ve performanstır. Bu iki öncelik, kullanıcı deneyimini doğrudan etkiler ve uygulamanın başarısı için kritik öneme sahiptir.

WAF (Web Application Firewall)

WAF (Web Uygulama Güvenlik Duvarı), web uygulamalarını çeşitli güvenlik tehditlerine karşı korumak için tasarlanmış özel bir güvenlik duvarıdır. Bu

tehditler arasında SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF) ve özellikle DDoS (Distributed Denial of Service) saldırıları bulunur. DDoS saldırıları, hedeflenen web kaynaklarını kullanılamaz hale getirmeyi amaçlayan kötü niyetli girişimlerdir. Bu saldırılar, çok sayıda bilgisayarın kontrol edilerek hedef web uygulamasına veya sunucusuna aşırı miktarda sahte trafik göndermesiyle gerçekleştirilir. WAF, bu tür saldırıları tespit etmek için gelen trafiği sürekli olarak izler, analiz eder ve potansiyel tehditleri filtreler. DDoS saldırıları söz konusu olduğunda, WAF, anormal trafik artışlarını belirleyebilir ve bu trafikleri yönetmek için önlemler alır. Bu önlemler arasında sahte trafik isteklerinin otomatik olarak engellenmesi, trafiğin düzenlenmesi ve meşru kullanıcı isteklerinin önceliklendirilmesi yer alır.

DDoS Saldırılarına Karşı WAF Stratejileri:

- **Trafik Analizi**: Gelen trafikteki anormallikleri, örneğin aşırı miktarda istek yapısını veya belirli bir kaynaktan gelen yoğun trafiği analiz eder.
- **Oran Limiti**: Belirli bir süre içinde izin verilen maksimum istek sayısını sınırlar. Bu, normalden çok daha yüksek istek oranlarına ulaşıldığında saldırıyı hafifletebilir.
- **Coğrafi Bloklama**: Saldırı trafiğinin belirli coğrafi bölgelerden geldiği tespit edilirse, bu bölgelerden gelen trafiği engelleyebilir.
- **Davranışsal Bloklama**: İnsan kullanıcıların davranışlarını taklit etmeyen trafik modellerini tespit edip bloke eder.

Cloudflare, geniş bir WAF hizmeti sunar. Kullanıcıların web uygulamalarını karmaşık saldırılara karşı korumak için önceden tanımlanmış kurallar seti ve özelleştirilebilir kurallar sağlar. Bu hizmet, web trafiğini sürekli olarak analiz eder ve şüpheli davranışları otomatik olarak engeller.

Load Balancer

Load Balancer, gelen istekleri birden fazla sunucu arasında dağıtarak, uygulamanın yükünü dengeler ve böylece tek bir sunucunun aşırı yüklenmesini önler. Aynı zamanda bir tür Reverse Proxy olarak da işlev görerek, istemci ve sunucular arasında bir ara katman oluşturur ve uygulama sunucularının doğrudan erişime maruz kalmasını engeller.

Kullanım Alanları:

- Yüksek trafikli web siteleri
- Büyük ölçekli web uygulamaları

- E-ticaret platformları

Connection Pooling

Connection Pooling, veritabanı bağlantılarının yönetimini optimize eden bir tekniktir. Uygulama tarafından yapılan veritabanı bağlantılarını bir havuzda saklar ve tekrar kullanılabilir hale getirir. Bu sayede, her istek için yeni bir bağlantı açma ve kapatma maliyetinden kaçınılarak performans artışı sağlanır.

Bir web uygulaması, kullanıcıdan gelen her sorgu için veritabanına bağlantı açarsa, bu yüksek maliyet ve gecikmelere neden olabilir. Connection Pooling kullanılarak, önceden açılmış ve havuzda tutulan bağlantılar yeniden kullanılır, böylece veritabanı operasyonları daha hızlı gerçekleşir.

CDN (Content Delivery Network)

CDN, içeriği kullanıcılara coğrafi olarak daha yakın sunuculardan teslim etmek için kullanılan bir ağdır. Bu yaklaşım, içerik yükleme sürelerini azaltır ve kullanıcı deneyimini iyileştirir. Aynı zamanda orijinal sunucular üzerindeki yükü azaltır.

Bir video paylaşım sitesi, dünya çapında kullanıcılara hizmet veriyorsa, CDN kullanarak videoların farklı coğrafi bölgelerdeki sunucularda önbelleğe alınmasını sağlayabilir. Kullanıcı bir videoyu izlemek istediğinde, CDN en yakın sunucudan videoyu sunar, böylece daha hızlı yükleme süreleri elde edilir.

Caching

Caching, sık erişilen verilerin veya hesaplamaların sonuçlarının geçici olarak saklanması işlemidir. Bu teknik, aynı bilgilere tekrar erişim gerektiğinde hızlı bir şekilde yanıt vermek ve sistem kaynaklarını daha verimli kullanmak için kullanılır.

Bir haber sitesi, ana sayfada yer alan popüler haberlerin içeriğini önbelleğe alabilir. Böylece, kullanıcılar ana sayfayı ziyaret ettiğinde, haber içerikleri doğrudan önbellekten servis edilir, bu da veritabanı sorgularını ve sunucu işlem yükünü azaltır.

Log Sunucu

Log sunucuları, sistemlerin, uygulamaların ve ağ cihazlarının ürettiği log verilerini toplar, saklar ve analiz eder. Güvenlik, hata ayıklama ve sistem performansı izleme gibi amaçlarla kullanılır. Güvenlik ihlallerinin erken tespiti, sistem hatalarının analizi ve kullanıcı davranışlarının incelenmesi bu sunucuların başlıca kullanım alanlarıdır.
Genellikle Syslog, Windows Event Log gibi standart log protokolleri aracılığıyla veri alır. Bu veriler zaman damgası, kaynak, olay türü gibi bilgiler içerir. Toplanan loglar, arama ve analiz için merkezi bir veritabanında saklanır.

- **ELK Stack (Elasticsearch, Logstash, Kibana):** Log verilerini toplama, işleme ve görselleştirme için kullanılır.
- **Splunk:** Güçlü bir veri toplama ve analiz platformudur, özellikle karmaşık sorgulama ve veri görselleştirme konusunda yeteneklidir.

Mail Sunucusu

Mail sunucuları, kurum içi ve dışı e-posta iletişimini sağlar. SMTP protokolü ile e-posta gönderimi, POP3 veya IMAP protokolleri ile e-posta alımı ve yönetimi yapılır. Kurumsal iletişim, müşteri destek hizmetleri, pazarlama ve bilgilendirme e-postaları bu sunucular üzerinden yönetilir.
Gönderilen e-postaları alıcıların posta sunucularına iletmek için SMTP kullanır. Alıcılar, IMAP veya POP3 protokolleri üzerinden kendi mail sunucularından e-postalarını alır. IMAP, postayı sunucuda tutarken, POP3 e-postaları indirir.

- **Microsoft Exchange:** Entegre e-posta, takvim ve kişiler yönetimi sunar.
- **Postfix:** Açık kaynaklı ve yüksek performanslı bir SMTP sunucusudur.

Session Servisi

Session servisleri, kullanıcıların web uygulamalarıyla etkileşim durumlarını yönetir. Web alışveriş sepetleri, kullanıcı tercihleri, oturum bilgileri gibi verileri saklar. E-ticaret siteleri, çevrimiçi bankacılık ve kişiselleştirilmiş içerik sunumu bu servislerin kullanım alanları arasındadır.
Kullanıcı girişi yapıldığında, session servisi bir session ID oluşturur ve bu ID'yi kullanıcıya ve sunucuya bildirir. Bu ID, kullanıcının sunucu ile olan tüm

etkileşimlerinde kullanılır ve kullanıcıya özel verilerin saklanmasını ve takip edilmesini sağlar.

- **Redis Session Store:** Hızlı veri erişimi için kullanılan popüler bir anahtar-değer deposudur.
- **JWT (JSON Web Tokens):** Kimlik doğrulama ve session yönetiminde kullanılan bir standarttır.

Yedekleme ve Felaket Kurtarma Sistemleri

Yedekleme ve felaket kurtarma sistemleri, veri kaybını önlemek ve sistem çökmesi gibi durumlarda verilerin ve hizmetlerin hızla geri yüklenmesini sağlamak için kritik öneme sahiptir. Bu sistemler, verilerin düzenli olarak yedeklenmesini ve acil durumlarda hızlı bir şekilde kurtarılmasını sağlar.

Kullanım Alanları:

- Veri merkezleri
- Kurumsal IT altyapıları
- Online hizmet sağlayıcılar

Bulut Mimarisi

Bulut mimarisi, veri depolama, işleme ve yönetimi için internet üzerinden erişilebilen kaynakları ve hizmetleri kapsayan, dağıtık bilgi işlem teknolojilerinin genel bir yapısını ifade eder. Son yıllarda, işletmelerin ve bireylerin veri erişimini, uygulama barındırmayı ve çeşitli IT kaynaklarını esnek, ölçeklenebilir ve maliyet etkin bir şekilde kullanmalarını sağlamıştır. Bulut bilişim kavramı 1960'larda ortaya çıktı, ancak gerçek potansiyeli 2000'lerin başında internet teknolojilerinin gelişmesiyle anlaşıldı. Amazon Web Services (AWS) 2006'da Elastic Compute Cloud (EC2) servisini başlatarak bulut bilişim hizmetlerinde bir devrim yarattı ve ardından Google Cloud ve Microsoft Azure gibi diğer büyük oyuncular piyasaya girdi. Günümüzde teknolojinin hemen her alanında devrim yaratmıştır. İşletmeler artık fiziksel altyapı yatırımlarının kısıtlamalarından kurtulmuş, esneklik, ölçeklenebilirlik ve maliyet etkinliği avantajlarından faydalanmaktadır.

Bulut bilişim, dijital dönüşümün itici gücü olarak kabul edilir ve gelecekte teknolojinin nasıl şekilleneceğinde önemli bir rol oynar.
Karmaşık bilgi işlem kaynaklarının yönetimini basitleştirir. Kaynakların otomatik dağıtımı, yük dengeleme, otomatik ölçeklendirme ve hata toleransı gibi özellikler, yüksek kullanılabilirlik ve güvenilirlik sağlar. Bulut hizmet sağlayıcıları, güvenlik, yedekleme ve felaket kurtarma gibi hizmetleri de müşterilerine sunarak, veri güvenliğini ve sürekliliğini garanti eder.
Bulut mimarisi, çeşitli bileşen ve servislerden oluşur:

IaaS (Infrastructure as a Service)

IaaS, sanal sunucular, ağ, depolama gibi temel bilgi işlem kaynaklarının kullanıcıya internet üzerinden sunulduğu bir modeldir. Kullanıcılar, fiziksel donanımı yönetme zahmetinden kurtulurken, kaynakları ihtiyaç duydukça kolayca ölçekleyebilirler.

- AWS EC2
- Microsoft Azure Virtual Machines
- Google Compute Engine

PaaS (Platform as a Service)

PaaS, geliştiricilere uygulamalarını geliştirmeleri, test etmeleri ve dağıtmaları için ihtiyaç duydukları platform ve araçları sunar. Bu, yazılım geliştirme süreçlerini hızlandırır ve altyapı yönetimiyle ilgili karmaşıklıkları ortadan kaldırır.

- Heroku
- Google App Engine
- Azure App Services

SaaS (Software as a Service)

SaaS, kullanıcılara internet üzerinden erişilebilen, tamamen işlevsel yazılım uygulamaları sunar. Kullanıcılar, yazılımı barındırmak veya yönetmek zorunda kalmadan uygulamaları kullanabilirler.

- Google Workspace
- Salesforce
- Microsoft Office 365

Bulut Mimarisi Modelleri

- **Kamu Bulutu**: Genel olarak internet üzerinden erişilebilen bulut kaynaklarını ifade eder. Kaynaklar, birden fazla kiracı arasında paylaşılır.

- **Özel Bulut**: Bir kuruluşun kendi iç kullanımı için ayrılmış bulut kaynaklarını ifade eder. Kaynaklar yalnızca bu kuruluş tarafından kullanılır ve genellikle güvenlik ve uyumluluk gereksinimleri nedeniyle tercih edilir.
- **Hibrit Bulut**: Kamu ve özel bulut kaynaklarının birleştirilmesidir. Kuruluşlar, veri ve uygulamalarını ihtiyaçlarına göre en uygun bulut türü arasında taşıyabilir.

Bulutun Web ile Olan Bağlantısı ve Modern Teknolojiler

Bulut bilişim, web teknolojilerinin gelişiminde ve dağıtımında önemli bir dönüm noktası oluşturmuştur. Docker, Kubernetes ve mikroservisler gibi modern teknolojiler, bulut bilişimin sağladığı esneklik ve ölçeklenebilirlik avantajlarından yararlanarak, uygulama geliştirme ve dağıtım süreçlerini kökten değiştirmiştir.

Docker

Docker, uygulamaları ve bağımlılıklarını hafif, taşınabilir, kendine yeterli konteynerler içinde paketlemek için kullanılan bir platformdur. Bu konteynerler, herhangi bir Linux veya Windows makinesinde sorunsuz bir şekilde çalışabilir, bu da uygulama geliştirmeden dağıtıma kadar süreçleri standartlaştırır ve basitleştirir.

Uygulamalar bir ya da birden fazla konteyner içinde çalıştırılır. Her konteyner, uygulamanın çalışması için gerekli tüm kodu, çalışma zamanını, sistem araçlarını ve kütüphaneleri içerir. Konteynerler, işletim sistemi çekirdeğini paylaşır ancak birbirlerinden izole edilmişlerdir, bu da kaynak kullanımını optimize eder ve güvenliği artırır.

Kullanım Alanları:

- Uygulama izolasyonu ve mikroservis mimarileri
- DevOps ve sürekli entegrasyon/devamlı dağıtım (CI/CD) süreçleri
- Çoklu platform ve bulut hizmetleri üzerinde tutarlı uygulama dağıtımı

Kubernetes

Kubernetes, konteynerize uygulamaların otomatik dağıtımı, ölçeklendirilmesi ve yönetilmesi için kullanılan açık kaynaklı bir sistemdir. Docker konteynerlerinin yanı sıra diğer konteyner teknolojilerini de destekler. Kubernetes, uygulama dağıtımlarını otomatize eder ve yüksek kullanılabilirlik, ölçeklenebilirlik ve kaynak kullanımını optimize eder.

Kubernetes, bir ya da birden fazla çalışma düğümünden (node) oluşan bir kümeyi (cluster) yönetir. Her düğüm, uygulama konteynerlerinin çalıştırıldığı bir makinedir. Kubernetes, uygulamaları bu düğümler arasında otomatik olarak dağıtır, ölçeklendirir ve gerektiğinde yeniden başlatır.

Kullanım Alanları:

- Mikroservis tabanlı uygulama dağıtımı ve yönetimi
- Otomatik ölçeklendirme ve yük dengeleme
- Kendi kendine iyileşme yetenekleri ile uygulama sürekliliği

Mikroservisler

Mikroservis mimarisi, bir uygulamanın daha küçük, bağımsız hizmetlerden oluşacak şekilde tasarlanmasını ifade eder. Her mikroservis, belirli bir işlevi gerçekleştirir ve genellikle bir API üzerinden diğer servislerle iletişim kurar. Bu yaklaşım, uygulama geliştirme, test ve dağıtım süreçlerini basitleştirir ve uygulamanın ölçeklenebilirliğini artırır.

Mikroservisler, belirli bir işlevi yerine getiren bağımsız hizmetlerdir. Bu hizmetler, genellikle API'ler aracılığıyla birbirleriyle iletişim kurarlar. Her mikroservis, kendi veritabanını, bağımlılıklarını ve kaynaklarını yönetir. Bu, hizmetlerin bağımsız olarak geliştirilmesini, test edilmesini ve ölçeklendirilmesini sağlar.

Bu teknolojiler, modern web ve bulut altyapılarının temel taşlarındandır ve gelişmiş otomasyon, ölçeklenebilirlik ve hizmet yönetimi sağlayarak işletmelerin ve geliştiricilerin ihtiyaçlarına cevap verir.

Kullanım Alanları:

- Büyük ve karmaşık uygulamaların geliştirilmesi ve yönetimi
- Bağımsız ekipler tarafından geliştirilebilen ve dağıtılabilen uygulama bileşenleri
- Bulut tabanlı uygulama dağıtımları

DevOps ve Sürekli Entegrasyon/Devamlı Dağıtım (CI/CD) Süreçleri

DevOps

DevOps, yazılım geliştirme (Dev) ve IT operasyonları (Ops) arasındaki işbirliğini artırmayı, yazılımın daha hızlı ve daha güvenilir bir şekilde geliştirilip dağıtılmasını sağlamayı hedefleyen bir kültür ve pratikler bütünüdür. DevOps yaklaşımının temel amacı, geliştirme ve operasyon

ekipleri arasındaki bariyerleri kaldırarak, ürün geliştirme sürecinin hızlanmasını ve operasyonel verimliliğin artırılmasını sağlamaktır. DevOps, otomasyon, sürekli entegrasyon (CI), sürekli dağıtım (CD), mikro hizmetler ve bulut bilişim gibi teknolojilerden yararlanır. Bu yaklaşımda, kodun hızla ve güvenilir bir şekilde üretim ortamına taşınması için otomatikleştirilmiş iş akışları kullanılır. Ekipler, altyapıyı kod olarak yönetir ve sürekli geri bildirim aracılığıyla hizmetlerin kalitesini ve güvenilirliğini artırır.

Kullanım Alanları:

- Sürekli yazılım geliştirme ve dağıtım süreçleri
- Otomatikleştirilmiş test ve dağıtım
- Hızlı geri bildirim döngüleri ve sürekli iyileştirme

Sürekli Entegrasyon ve Sürekli Dağıtım (CI/CD)

Sürekli Entegrasyon (Continuous Integration - CI), geliştiricilerin yazdıkları kodu düzenli aralıklarla merkezi bir repoya birleştirmeleri (entegre etmeleri) pratiklerini ifade eder. Bu yaklaşım, hataların erken tespitini ve kolay düzeltilmesini sağlar. Sürekli Dağıtım (Continuous Deployment - CD), her kod değişikliğinin otomatik testlerden geçirilerek üretim ortamına güvenle otomatik olarak dağıtılmasını sağlayan bir süreçtir.

CI/CD pipeline'ı, kodun geliştirilmesinden test edilmesine, konteynerizasyona ve üretim ortamına dağıtılmasına kadar olan süreci kapsar. CI sürecinde, kod değişiklikleri repoya her entegre edildiğinde otomatik olarak test edilir. CD sürecinde ise, testlerden geçen kod değişiklikleri otomatik olarak üretim ortamına dağıtılır, böylece manuel müdahale gereksinimi minimize edilir.

DevOps ve CI/CD, modern yazılım geliştirme süreçlerinde önemli roller oynar. Bu yaklaşımlar sayesinde, yazılım geliştirme ekipleri daha hızlı, esnek ve verimli çalışır, kod kalitesi artar ve yazılımın müşterilere sunulma süresi kısalır. Bu metodolojiler, bugünün dinamik ve rekabetçi teknoloji ortamında işletmelerin başarılı olmaları için kritik öneme sahiptir.

Kullanım Alanları:

- Agile yazılım geliştirme süreçleri
- Mikroservis tabanlı uygulama mimarileri
- Bulut tabanlı uygulama dağıtımları

SORULAR

1-) World Wide Web (WWW) ilk olarak hangi yıl ve kim tarafından tasarlanmıştır?

1980, Bill Gates
1989, Tim Berners-Lee
1991, Linus Torvalds
1985, Steve Jobs

2-) Soru 2

HTTP ve HTTPS arasındaki fark nedir?
HTTP daha hızlıdır, HTTPS ise daha yavaştır.
HTTP şifreli bir bağlantı sağlar, HTTPS sağlamaz.
HTTPS şifreli bir bağlantı sağlar, HTTP sağlamaz.
Her ikisi de tamamen aynıdır.

3-) DNS'in temel işlevi nedir?

Web sayfalarını hızlandırmak
Alan adlarını IP adreslerine çevirmek
E-posta adreslerini yönetmek
Güvenli internet bağlantısı sağlamak

4-) Soru 4

Bir web sayfasının içeriğini almak için bir web sunucusuna hangi HTTP isteği gönderilir?

GET

Soru 5

REST API'ler genellikle hangi veri formatını kullanır?

JSON

Soru 6

Load Balancer'ın ana işlevi nedir?

Veri şifreleme

İstekleri birden fazla sunucu arasında dağıtmak

İçerik önbelleğe almak

Kullanıcı oturumlarını yönetmek

Soru 7

WAF (Web Uygulama Güvenlik Duvarı) neyi amaçlar?

Web uygulamalarını güvenlik tehditlerine karşı korumayı

Trafik hızlandırmayı

Veritabanı optimizasyonu

API hızlandırmayı

Soru 8

WebSocket protokolünün kullanıldığı bir durum nedir?

Statik web sayfası sunumu

E-posta gönderimi

Güvenli dosya transferi

Gerçek zamanlı veri iletimi

Soru 9

Bir web sayfasının adresini tarayıcıya girdiğinizde ilk olarak hangi sistem devreye girer?

WAF

CDN

DNS

API

Soru 10

CDN (İçerik Dağıtım Ağı) kullanmanın ana avantajı nedir?

Sunucu maliyetlerini azaltmak

İçerik yükleme sürelerini azaltmak

Web sitesi güvenliğini artırmak

Veritabanı performansını iyileştirmek

EMİRHAN KELLECİGİL

25.12.2025 Tarihinde tamamlamış olduğum Modern Web Nasıl Çalışır adlı

Akademik çalışmam.

Kaynak:Hackwiser