

# DGH BİTİRME PROJESİ

ARAŞTIRMA ÖDEVLERİ

# 1) C# NEDİR?

**C#** yani diğer bir adıyla **C Sharp**, Microsoft tarafından geliştirilen sunucu ve gömülü sistemleri çalıştırmak için tasarlanmış programlama dilidir. **.NET Framework** ortamında kullanılmak üzere geliştirilmiş nesne tabanlı programlama dillerinden birisidir.

**C# programlama dili** ile web programlama konusunda güçlü projeler geliştirilir. Kullanımı kolay modern bir dildir. C# ile uygulama geliştirilirken kullanılan platform **Visual Studio IDE**'dir. Visual Studio birçok işletim sistemlerinde kullanılmaktadır.

C# programlama dili **C** ve **C++** programlama dillerinden türemiştir. C, **Java** veya **C++** dillerinden biri ile ilgilenmiş yazılım geliştiriciler C#'ın kurallarına kolayca uyum sağlayabilirler. **Debugger** özelliği yani gelişmiş hata giderme özelliği ile yazılımcıların işlerini kolaylaştırır. Bu özellik ile **"syntax"** hatalarını kolayca tespit eder. Dizinlerdeki hataları yazılımcıya ayrı bir ekranda göstererek yazılımcının bu hataları kolayca tespit etmesini sağlar. Bu özellikleri sonucunda **Backend Developer**'lar tarafından yaygın kullanılan bir programlama dili haline gelmiştir.

# C# Programlama Dili İle Neler Yapılabilir?

C# ile de farklı uygulamalar ve programlar oluşturulabilir. Masaüstü, mobil, web, ve oyun geliştirmede kullanılır.

Ayrıca IoT, bulut ve API'ler gibi her türlü uygulamayı C# kullanarak oluşturabiliriz.

## 2) Data Structures Nedir?

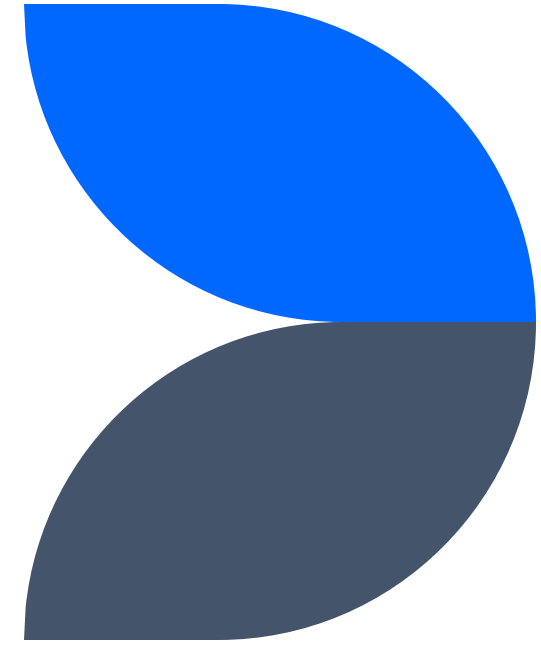
*Veri yapıları; organize edilmiş bir veriler koleksiyonudur. İnt, float, double gibi türlerde olabilir. Veri yapılarına ihtiyaç duyma sebebimiz ise kodumuzun daha hızlı çalışmasını, kodumuzu daha rahat oluşturabilmemizi sağlamasıdır.*

*Veri yapısı; C, C++, Java gibi bir programlama dili değildir. Verileri bellekte yapılandırmak için herhangi bir programlama dilinde kullanabileceğimiz bir dizi algoritmadır.*

Veri türlerinde yaygın olarak kullanılan bazı işlemler şunlardır:

- 1- Searching (Arama)
- 2- Sorting (Sıralama)
- 3- Insertion (Ekleme)
- 5- Updation (Güncelleme)
- 5- Deletion (Silme)

Özetle; veri yapıları, veri öğelerini mantıklı bir şekilde bir araya getirir ve verilerin etkin kullanımını, kalıcılığını ve paylaşımını kolaylaştırır.



### 3) Git ve GitHub Nedir?

**Git**, yazılım geliştirme süreçlerinde kullanılan bir **versiyon kontrol sistemidir**. **Git** sayesinde yapacağınız projelerin adım adım versiyonlarının kopyalarını alarak daha sonra ihtiyaç duyduğunuzda aldığınız kopyalara yani versiyonlara kolayca dönebiliyorsunuz. İlk sürümü Linux çekirdeği'nin geliştirilmesinde kullanılmak üzere 2005 yılında **Linus Torvalds** tarafından tasarlanıp geliştirilmiş, 2019 yılı itibarıyla %70 pazar payına ulaşmıştır. Açık kaynaklı özgür bir yazılım ürünü olan Git'i istediğiniz gibi kullanabilirsiniz.



# GitHub

**Github** ise projelerimizin saklandığı (depolandığı) uzak sunucudur. **Github**'a projelerinizi ekleyebilir aynı zamanda istediğiniz public olan farklı projelere **Github** üzerinden erişerek projeyi bilgisayarınıza indirebilirsiniz. Hatta istediğinizde bu projeler üzerinde değişiklikler yaparak **Pull Request** gönderebilirsiniz.

# 4) Tree nedir?

**Tree**, bilgisayar bilimlerinde kullanılan bir veri yapısıdır. Ağaç şeklindeki bu veri yapısı, birbirine bağlı düğümlerden oluşur. Her düğüm, bir veya daha fazla alt düğüme sahip olabilir. Ağaç yapısında, her düğümün yalnızca bir üst düğümü vardır ve yalnızca bir düğüm kök düğüm olarak belirlenir. Bilgisayar bilimlerinde ağaçlar, verilerin hiyerarşik bir şekilde saklanmasına olanak tanır. Ağaçlar, birçok algoritmanın temelini oluşturur ve birçok programlama dili tarafından desteklenir. Ağaçlar, birçok veri yapısının temelini oluşturur. Örneğin, ağaçlar, dosya sistemlerinde, XML belgelerinde, HTML sayfalarında, veritabanlarında ve daha birçok yerde kullanılır. Ağaçların birçok türü vardır

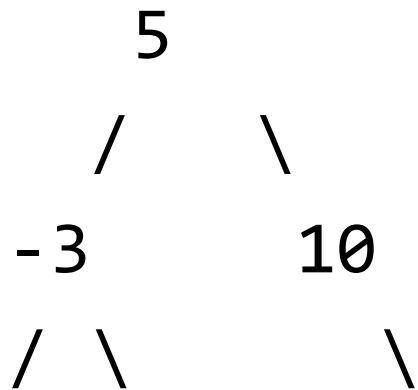


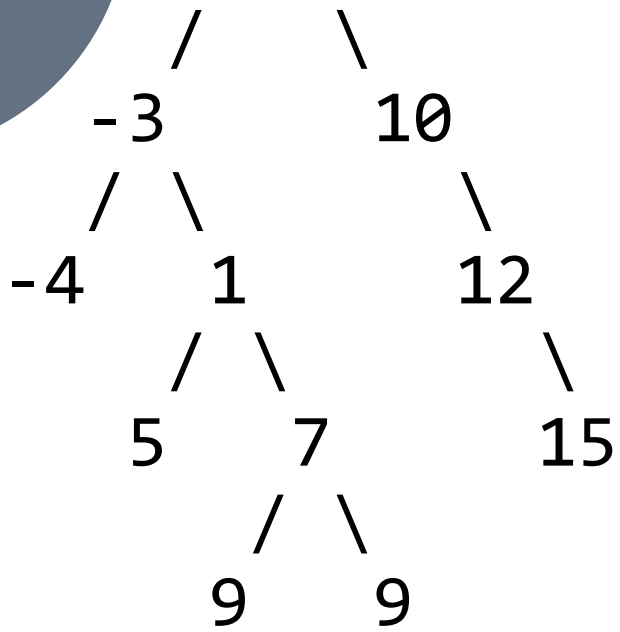
# Tree(ağaç) Çeşitleri

- İkili ağaçlar (Binary Tree)
- İkili arama ağaçları (Binary Search Tree)
- Trie (Metin Ağacı)
- AVL Ağacı (AVL Tree)
- Yığın Ağacı (Heap)

# Binary Tree Örnek

Binary Search Tree (İkili Arama Ağacı) ağaç şeklindeki bir veri yapısıdır. Ağaç, birbirine bağlı düğümlerden oluşur ve her düğüm, bir veya daha fazla alt düğüme sahip olabilir. İkili arama ağacı, her düğümün yalnızca bir üst düğümü olduğu ve her düğümün en fazla iki alt düğümü olduğu bir ağaçtır. Aşağıdaki örnekte, 5, -3, 10, 5, 7, 12, 1, -4, 15, -3, 10, 9, 9, 7, 8 sayıları Binary Search Tree ile gösterilmiştir:





Yukarıdaki örnekte, ağacın seviyesi 7'dir. Toplamda 5 tane düğümlenmiş yaprak vardır. 8 sayısına ulaşmak için genişlik yöntemiyle 4 adım, derinlik yöntemiyle ise 3 adım gereklidir. -4 sayısına ulaşmak için genişlik yöntemiyle 5 adım, derinlik yöntemiyle ise 3 adım gereklidir

# 5)Domain, Hosting, SSL Nedir?

Domain, Türkçe karşılığı "alan adı" olan bir terimdir. Herhangi bir web sitesine girerken o web sitesinin IP adresi ile ilgili alana bağlanmanız gerekir. Söz konusu IP adreslerinin herkes tarafından bilinmesi ve akılda tutulması mümkün değildir. Dolayısıyla domain kavramı geliştirilmiş olup, insanların IP adresi yerine domain ismi ile web sitesine bağlanması sağlanmıştır.

Domain, en genel ifade ile web sitesinin adı ve adresidir. Kullanıcılar web sitesini bu alan adı ile tanırlar. Seçilen alan adının; "alanadi" olduğunu düşünürseniz, domain de "alanadi.com" ya da "alanadi.net" gibi web sitesine yönlendiren adreslerdir.

# Hosting Nedir?

İnternet siteleri de tıpkı gerçek hayattaki siteler gibi bir arazi üzerine kurulurlar. Bir internet sitesini yayınlamak için gerekli olan sanal araziye “hosting” adı verilir. İngilizce bir terim olan hosting, dilimize “web barındırma alanı” ya da “barındırma alanı” olarak çevrilebilir.

Hosting; web sayfanıza ait tüm dosyaların, yılın 365 günü bir bilgisayarda saklanması ve erişime açık tutulması hizmetidir. Bu bilgisayarlara ise “server” veya “web sunucusu” adı verilir.

Web sunucuları, evinizde ya da ofisinizde bulunan bilgisayarlardan donanımsal olarak daha özellikli makinalardır. Serverlar; “datacenter” veya “veri merkezi” olarak adlandırılan yerlerde barındırılırlar. Veri merkezleri; çok yüksek hızlı internet bağlantısına sahiptir, elektrik kesintileri yaşanmaz, sel ve deprem benzeri doğal afetlere karşı korunaklıdır. Bu sebeple, siteleriniz her koşulda ziyaretçilerinin erişimine açık olur.

# SSL Nedir?

SSL internet ortamında çok karşılaşılan kavramlardan biridir. Genellikle alışveriş sitelerinde oldukça sık rastlanmaktadır. SSL'in açılımı **Secure Socket Layer**'dir. Türkçe anlamıysa **Güvenli Giriş Katmanı**'dir. SSL kişisel gizlilik ve güvenilirlik sağlayan, network üzerindeki bilgi transferi sırasında bilginin bütünlüğü ve gizliliği (*data protection*) için sunucu ile istemci arasındaki iletişimin şifrelenmiş şekilde yapılabilmesine imkan veren bu sayede gizliliğinin ve bütünlüğün korunmasını sağlayan **Netscape** tarafından geliştirilmiş bir güvenlik protokolüdür diye tanımlayabiliriz.

SSL protokolü bütün yaygın **web sunucuları** (server) ve **tarayıcıları** (browser) tarafından desteklenen bir protokoldür. SSL, standart bir algoritmadır. Milyonlarca web sitesinde güvenli veri iletişimi için kullanılmaktadır. SSL fonksiyonun çalışabilmesi için sunucu tarafında bir **anahtar** (private key) ve istemci tarafında çalışacak bir **sertifikaya** (Public Key) ihtiyaç duyulmaktadır.

## 6) MVC Nedir?

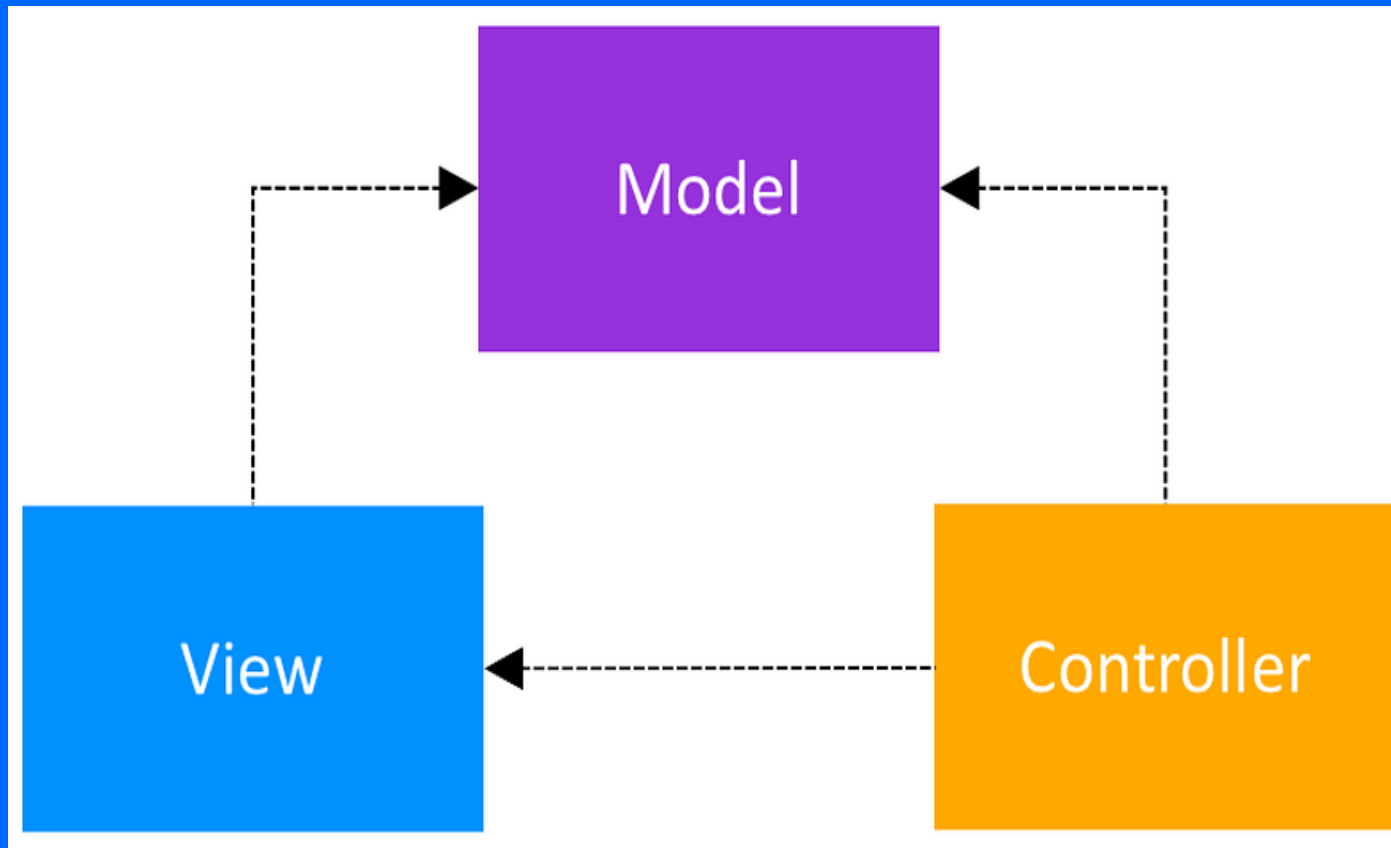
MVC, Yazılım Mühendisliği'nde önemli bir yere sahip architectural patterns (yazılım mimari desenleri)'ın bir parçasıdır. Model, View ve Controller kelimelerinin baş harflerinden oluşan MVC (Model-View-Controller), 1979 yılında Tygve Reeskaug tarafından oluşturulmuş ve yazılım geliştirmede bir çok projede kullanılmıştır. Son dönemlerde Microsoft'un MVC desenini Asp.Net teknolojisi ile birleştirmesi ile popülaritesi daha da artmıştır.

# MVC Nedir?

MVC, Yazılım Mühendisliği'nde önemli bir yere sahip architectural patterns (yazılım mimari desenleri)'in bir parçasıdır. Model, View ve Controller kelimelerinin baş harflerinden oluşan MVC (Model-View-Controller), 1979 yılında Tygve Reeskaug tarafından oluşturulmuş ve yazılım geliştirmede bir çok projede kullanılmıştır. Son dönemlerde Microsoft'un MVC desenini Asp.Net teknolojisi ile birleştirmesi ile popüleritesi daha da artmıştır.



# MVC



# MVC

## Model Nedir?

Model, MVC’de projenin iş mantığının (business logic) oluşturulduğu bölümdür. İş mantığıyla beraber doğrulama (validation) ve veri erişim (data access) işlemleri de bu bölümde gerçekleştirilmektedir.

Model tek katmandan oluşabileceği gibi kendi içinde birden fazla katmandan da oluşabilir. İç yapılandırma projenin büyüklüğü ile yazılım geliştiricinin planlamasına kalmış bir durumdur. Eğer proje büyük çaplı ise modeli birden çok katmana ayırmak projenin yönetimi açısından faydalı olacaktır.

# MVC

## View Nedir?

View, MVC’de projenin arayüzlerinin oluşturulduğu bölümdür. Bu bölümde projenin

kullanıcılara sunulacak olan HTML dosyaları yer almaktadır. Projenin geliştirildiği yazılım dillerine göre dosya uzantıları da değişebilmektedir. Projelerin büyüklüğüne göre dikkat edilmesi gereken bir nokta ise, klasörlemedir.

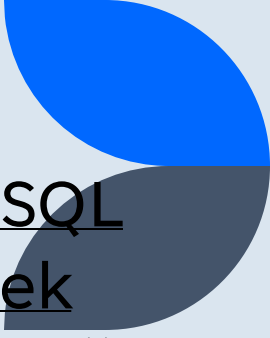
Eğer bir web projesi geliştiriyorsanız, projenin View’larının yer aldığı klasörlerinin hiyerarşisi, ilerleyen dönemlerde karmaşıklığa sebep olmaması için dikkatli yapılmalıdır. Kimi yazılım geliştiriciler web projelerinde HTML dosyaları ile Javascript, CSS ve resim dosyalarını aynı klasör içinde barındırmaktadır. Ufak bir ayrıntı gibi görünse de projenin ilerleyen dönemlerinde ciddi problemler oluşturmaktadır.

# MVC

## Controller Nedir?

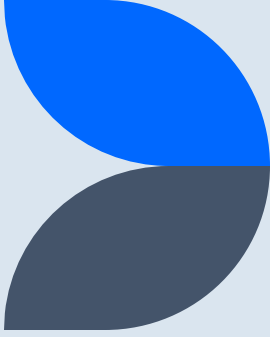
Controller, MVC’de projenin iç süreçlerini kontrol eden bölümdür. Bu bölümde View ile Model arasındaki bağlantı kurulur. Kullanıcılardan gelen istekler (request) Controller’larda değerlendirilir, isteğin detayına göre hangi işlemlerin yapılacağı ve kullanıcıya hangi View’ın döneceği (response) belirtilir.

# 7) PostgreSQL, Docker Nedir?



PostgreSQL, açık kaynak kodlu, ilişkisel veritabanı yönetim sistemidir. SQL dilini, karmaşık veri iş yüklerini yüksek verimlilikle işlemesini sağlayan ek özelliklerle kullanır ve genişletir . PostgreSQL, güvenilirliği, ölçeklenebilirliği, kararlılığı ve güvenliği ile en gelişmiş ve profesyonelce kullanılan açık kaynaklı bir veri tabanı yönetim sistemidir . PostgreSQL, çoğu işletme tarafından özellikle backend geliştirmeler için kullanılan en popüler ve güvenilir veri tabanı sistemlerinden biridir . PostgreSQL, hem ilişkisel (SQL) hem de ilişkisel olmayan (JSON) sorgulamayı destekleyen açık kaynaklı, kurumsal sınıf ve gelişmiş bir nesne-ilişkisel veri tabanı sistemidir . PostgreSQL, Windows, macOS, Linux, UNIX vb. gibi önde gelen tüm işletim sistemleriyle uyumludur . PostgreSQL, belirli bir noktadan sonra kurtarma, eşzamansız çoğaltma, Çok Sürümlü Eşzamanlılık Kontrolü – Multi-Version Concurrency Control (MVCC), iç içe işlemler, tablo bölümleme ve daha fazlasını içeren tonlarca özellik sunar

# Docker Nedir?



Docker, uygulamaların hızla derlenmesini, test edilmesini ve dağıtılmasını sağlayan bir yazılım platformudur . Docker, yazılımları kitaplıklar, sistem araçları, kod ve çalışma zamanı dahil olmak üzere yazılımların çalışması için gerekli her şeyi içeren **container** adlı standartlaştırılmış birimler halinde paketler . Docker, bir **konteynerleştirme teknolojisi** olarak adlandırılır. Konteynerler, uygulamaları ve onların bağımlılıklarını bağımsız ve izole bir ortamda çalıştırmak için kullanılır . Bir Docker konteyneri, tüm uygulama kodunu, kütüphaneleri ve konfigürasyon dosyalarını içerir . Docker, yazılım geliştiricilerin uygulamalarını hızlı bir şekilde dağıtmalarına ve çalıştırmalarına olanak tanır

## 8) Algoritma Nedir?

**Algoritma**, belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yol anlamına gelir, matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesine denir.

**Algoritma**, bir işin hangi aşamalardan geçilerek yapılacağını gösteren çalışma planıdır. Algoritma bir programlama dili değildir. Programlama dillerine yol gösteren bir yöntemler dizisidir. Her dilde algoritma yazılıp uygulanabilir.

## 9) Cross Platform Nedir?

**Cross-platform** mobil uygulama, birden fazla mobil alt yapı platformunda çalışabilecek şekilde geliştirilen mobil uygulama türleridir. Bu uygulamalar hem İOS hem de Android gibi birden fazla işletim sistemiyle uyumlu şekilde çalışabilirler. Çift platform olarak da tabir edilen bu uygulamalarla tek bir kod sistemi üzerinden farklı platformlarda çalışma özelliği olan uygulamalara sahip olabilirler.

Bir işletmenin ürün ve hizmetlerini daha hızlı bir şekilde piyasaya sunabilmesi adına çok elverişli sistemlerdir. Bu uygulamalar birçok mobil işletim sistemine entegre olabildiği için geniş kitlelere ulaşmada etkili seçimlerdir.

Birçok işletme mobil uygulamaya sahip olmaya karar verdiğinde tercihini yerel uygulamalardan yana kullanmaya odaklanır. Ancak native mobil uygulamalar sadece Android ya da IOS sistemlerinde çalışabilir. Bu yönüyle yerel uygulamalar çok yüksek maliyetli olabilmektedir.



# 10) API Nedir?

**Uygulama Programlama Arabirimi** anlamına gelen **API**, iki uygulamanın belirli tanımlar aracılığıyla birbiriyle **iletişim** kurmasına olanak tanıyan bir **yazılım aracıdır**.

API nedir sorusuna kısaca; bir uygulamanın **verilere, sunucu yazılımına** veya diğer programlara ulaşabilmek için kullandığı **bağlantı arayüzüdür** denilebilir.

Birçok veri, tekrar altyapı oluşturmaya gerek kalmadan web üzerinde veya akıllı telefonlarla API'ler sayesinde kolayca alınabiliyor. Bu sayede yeni bir programa ihtiyaç olmuyor, iletişim **pratik** ve **hızlı** bir şekilde hallediliyor.

API terimi, ilk defa 1974 yılında **Christopher J. Date** tarafından yayımlanan "**The Relational and Network Approaches: Comparison of the Application Programming Interface**" adlı makalede geçmiştir.

# 11)Front-End, Back-End, Full-Stack Nedir?

**Frontend:** Frontend'in Türkçe karşılığı "Önyüz"dür. Yapılma aşamasındaki bir web sitesinin ön yüzünü (client-side) HTML, CSS ve JavaScript gibi teknolojileri kullanarak web sitesinin görsel tarafını oluşturan kişilere ise front-end developer ( Ön yüz geliştirici ) denir. Web sitesinde kullanılacak renkler, içeriklerin yerleşimi, yazı tipinin seçilmesi ve uygulanması gibi birçok görevi bünyesinde barındırır.

# Back-End Nedir?

Türkçeye “Arkayüz” olarak çevrilmiştir. Kullanıcıların görmediği kısımların ve ana sistemin arkayüzün geliştirilme işlemidir. Bir programın hangi programlama diliyle yazılacağını bulmak, bu yazılımın programlama dilleriyle meydana getirilmesi ve bir veritabanına bağlanması gibi işlevlerin tümü Backend yazılım tasarımına girer.

Sistemin mimarisini oluşturulması, veri tabanı yönetiminin planlanması, sunucu ayarlamalarının yapılması, sistemin maksimum seviyede verimli ve hızlı çalışmasını sağlamak vb. işlevler gerçekleştirilir. Python, PHP, Ruby, Java, C#, ASP.NET, MySQL, MS Sql, MongoDB teknolojileri kullanılır. Bu alanda çalışan kişiler “Back-end

Developer” ( Arka yüz geliştirici ) olarak isimlendirilir.

# Full-Stack Nedir?

Full Stack Developer, web sitesi projesinin tüm aşamalarında yetkin şekilde çalışabilen, kullanıcı arayüzü tasarımı ve kullanıcı deneyimi konularında bilgi sahibi olan, sunucu ve veritabanı işlevleri konusunda uzmanlaşmış çok yönlü profesyoneldir. Bu kişiler hem ön yüz (front-end) hem de arka yüz (back-end) teknolojilerine hakimdir. Modern web teknolojilerini, çeşitli programlama dillerini ve veritabanı sistemlerini kullanarak dinamik web siteleri veya uygulamalar geliştirebilir. Ayrıca sürekli değişen teknoloji trendlerine ayak uydurarak yenilikçi çözümler sunma yeteneğine sahiptir.

# 12)Div, Html, Css, Responsive Nedir?

Div, HTML dilinde web sayfalarını bölümlere ayırmak ve bu bölümlere CSS ile stil değerleri vermemizi sağlayan bir HTML etiketidir . Div etiketi, block bir etikettir. Yani bulunduğu satırı tamamen kaplar . Div etiketi, sayfamızdaki kodları anlamsal olarak gruplamak için de kullanılabilir . Div etiketi, HTML sayfasında bir bölüm oluşturmak istediğimizde kullanılır . Div etiketi, içinde bulunan öğelerin sahip olduğu yükseklik kadar bir yüksekliğe sahip olur . Div etiketi, CSS ile birlikte kullanıldığında, web sayfalarının tasarımını daha düzenli ve anlaşılır hale getirir

# HTML Nedir?

Web tasarım konusunda araştırma yapan hemen herkesin karşına çıkan temel kavram **HTML**dir. Web sayfalarını oluşturma aşamasında kullanılan standart bir metin işaret dili olan **HTML açılımı** “Hyper Text Markup Language” olarak bilinir. Genel bilinen yanlış kanının aksine HTML bir programlama dili değildir. Daha açık anlatmak gerekirse, Chrome, Firefox, Yandex gibi tarayıcıların okuyup anlamlandırdığı dil HTML dilidir.

**HTML, web tasarımcılarına sayfalar ve uygulamalar için yapı profilleri, bağlantılar, blok alıntılar, paragraflar ve başlıklar oluşturmalarında yardımcıdır.** Bu konuda basit kod yapıları olan etiketler ve nitelikler kullanılarak web sayfaları şekillendirilebilir. HTML için aslında bir web sitesinin iskeleti denilebilir. Yani HTML kodları olmadan web sitesi kodlanamaz.

# CSS Nedir?

İngilizcesi “**Cascading Style Sheets**” olan CSS açılımı, “**Basamaklanmış Stil Katmanları**” anlamına gelmektedir. “**CSS nedir?**” sorusunun cevabı ise web sitelerinin görsel olarak şekillendirilmesine olanak tanıyan ve kendine has kuralları olan bir tanım dilidir şeklinde ifade edilebilir. **HTML ve JavaScript ile birlikte en temel web teknolojileri** arasında bulunan CSS, web sayfaları üzerinde oldukça fazla görsel denetim sunar.

CSS sayesinde web sayfalarının mizanpajı, renkleri, kullanılan fontlar, ara başlıklar, görsel efektler ve diğer görsel unsurlar üzerinde etkili ve fonksiyonel bir kontrol sağlayabilirsiniz. Kullanıcı dostu olan ve görsel açıdan etkileyici web siteleri için HTML ve JavaScript’in yanı sıra yazılan CSS kodları son derece önem taşır. **Etkili bir CSS kodlaması, web sitelerine yüksek bir kullanılabilirlik suna**

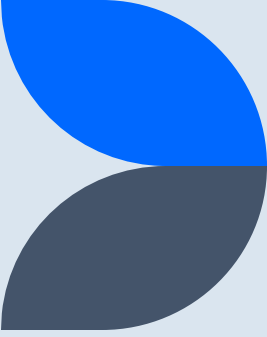


# 13) Şifreleme Yöntemleri(Kriptografi)

**Base64 Nedir:**Base64 Encoding, binary verileri metne dönüştürerek saklamaya veya iletmeye yarayan tekniklerden en yaygın olarak kullanılanıdır . Bu teknik ile temel olarak 8 bitlik baytlardan oluşan binary veri 6 bitlik ( $2^6 = 64$ ) parçalara bölünür. 6 bit ile ifade edilen 64 farklı sayı, ASCII karakter kümesinde Printable Character olarak ifade edilen ve aşağıda gösterilen 64 farklı karakterle eşleştirilmiştir. Eldeki 6 bitlik verinin bu tablo ile eşleştirilmesi ile Base64 Encoding yapılmış olur. Base64 Encoding'in bir kullanım alanı da modern tarayıcılarda imaj ve diğer dosyaların Data URLs formatı kullanılarak HTML ve CSS dokümanlarına Base64 ile kodlanarak eklenebilmesidir . Base64 Encoding'in bir diğer kullanım alanı, metin iletmek üzere kurgulanmış ve binary uyumu olmayan mail, HTTP, FTP gibi protokoller ile binary veri iletmektir . Base64 kodlama ile kodlaması yapılan her 3 bayt için 1 bayt harcadığı ve veri boyutunun yaklaşık olarak 3'te 1 oranında arttığı görülmektedir .



# Decoder-Encoder Nedir?



Kodlayıcı (Encoder) ve kod çözücü (Decoder) birbirinden farklı iki kavramdır. Encoder, verileri bir biçimden başka bir biçime dönüştürmek için kullanılırken, decoder ise kodlanmış verinin ilk halinin yeniden elde edilmesi adına kullanılır. Encoder ve decoder arasındaki farklar şunlardır:

- Encoder, verileri bir biçimden başka bir biçime dönüştürmek için kullanılırken, decoder, kodlanmış verinin ilk halinin yeniden elde edilmesi adına kullanılır.
- Encoder, fiziksel bir çözücüdür ve belli bir mekanik yapısı bulunmaktadır. Decoder ise daha çok bir şifre çözücü olarak ele alınır ve değerlendirilir.

MD5, SHA ve Breyted, verilerin bütünlüğünü test etmek için kullanılan tek yönlü şifreleme algoritmalarıdır .

- **MD5 (Message-Digest 5):** Ron Rivest tarafından geliştirilen ve bugün çeşitli İnternet uygulamalarında kullanılan bir karma algoritmasıdır. İsteğe bağlı bir uzunluk dizesinden 128 bitlik bir dize değeri oluşturmak için kullanılabilen bir şifreleme karma algoritmasıdır .
- **SHA (Secure Hash Algorithm):** ABD Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından geliştirilen Güvenli Karma Algoritma, belirtilen bir şifreleme hash fonksiyonları ailesidir. Federal Bilgi İşleme Standardı (FIPS 180-2) dört adet güvenli karma algoritmasını (SHA-1, SHA-256, SHA-384 ve SHA-512) belirtir. Tek yönlü bir mesaj fonksiyonunu kullanır. Bir mesaj özeti olarak adlandırılan 160 ile 512 bit yoğunlaştırılmış gösterimi üretmek için maksimum uzunluğu 264 – 2128 – bit arasındadır .
- **Breyted (Blowfish):** Bruce Schneier tarafından tasarlanan bir blok şifreleme algoritmasıdır. 64 bit ile 448 bit arasında anahtar uzunluğu ile çalışabilir. Blowfish, birçok uygulama tarafından kullanılan bir şifreleme algoritmasıdır .

# 14) Web server , Application server nedir?

## Web Server

Web server ya da ağ sunucusu, internet üzerinde bir web sitesinin yayınından sorumlu olan sunucudur. Web server, Hosting ya da “barındırma” işlemini internet protokolü üzerinden sunan bir sunucudur. Barındırma ya da hosting, Web sayfalarını internette yayınlamak için gerekli alanın kiralınmasıdır. Diğer bir ifade ile hosting, bir Web sitesinde yayınlanmak istenen sayfaların, resimlerin veya dokümanların internet kullanıcıları tarafından erişebileceği bir bilgisayarda tutulmasıdır.

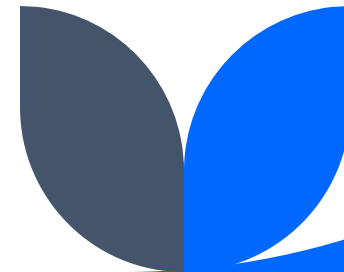
# Application Server

Uygulama sunucusu, dağıtık yapıdaki bir ağda bulunan bir bilgisayarda çalıştırılan sunucu yazılımıdır. Üç katmanlı uygulamaların bir parçasıdır. Bu üç katman: Kullanıcı arayüzü (GUI), uygulama sunucusu ve veritabanı sunucusudur.

Uygulama sunucusu, internet üzerinde bir web sitesinin yayınından sorumlu olan sunucudan farklıdır. Web sunucusu, HTTP protokolünü kullanarak internet siteleri üzerinde yer almakta olan dosyaları kullanıcılara sunan, kullanıcıların her bir talep isteğini yanıtlayan ve bilgisayarların otomatik mekanizmaları tarafından yürütülen yazılımlardır

# 15) Monolitik Uygulama Nedir?

Monolitik uygulama, tüm ilgili modüllerin tek bir dağıtılabilir yürütme birimi olarak paketlenildiği bir uygulama sistemidir. Bu uygulama geliştirmeleri yapıldıkça uygulama paketi büyür. Uygulamayı bölme ya da parçalama gibi bir özelliği bulunmadığından büyük kod yapısı ortaya çıkar. Yönetmesi ve bakımı zorlaşır. Aynı projeyi birden fazla kişi tarafından geliştirilmesi dependency sorunları ile karşılaşmamıza sebep olur. Object Oriented olarak tasarlanmış bir projede aynı class üzerinde birden fazla kişinin çalışması zorlaşır. Conflict çıkma olasılığı artar. Conflict'lerin yönetilmesi zorlaşır. CD/CI süreçleri bu stratejiden olumsuz olarak etkilenir.



# 16) Microservisler Nelerdir?

Microservice, bir uygulamanın farklı işlevlerinin, birbirinden bağımsız ve küçük yazılım parçalarına bölünmesiyle oluşan bir yazılım mimarisidir. Bu yaklaşım, uygulamanın daha modüler ve ölçeklenebilir olmasını sağlar. Her bir microservice, kendine özgü bir işlevi yerine getirir ve diğer microservislerle belirli bir protokol üzerinden haberleşir. Bu sayede, bir microservisdeki değişiklikler diğer microservisleri etkilemez ve uygulamanın bakımı daha kolay hale gelir. Microservice mimarisi, uygulamanın daha esnek, ölçeklenebilir ve dağıtık olmasını sağlar. Ancak, bu mimarinin uygulanması, uygulamanın daha karmaşık hale gelmesine neden olabilir.

# 17) Linux Nedir?

Linux, açık kaynak kodlu bir işletim sistemi ailesidir . İlk olarak Linus Torvalds tarafından geliştirilmiştir ve günümüzde birçok farklı dağıtımı bulunmaktadır. Linux, Unix benzeri bir işletim sistemi olup, ücretsiz ve açık kaynak kodlu bir yapıya sahiptir. Bu sayede, kullanıcılar işletim sistemi kodlarını inceleyebilir, değiştirebilir ve geliştirebilirler. Linux, birçok farklı cihazda kullanılabilir ve geniş bir donanım desteğine sahiptir. Ayrıca, Linux işletim sistemi, diğer işletim sistemlerine göre daha güvenli ve daha az hata verir .

# 18) Modern bilgisayar katmanları

Bilgisayar ağlarındaki veri iletimi, OSI modeli olarak bilinen bir çerçeve içinde gerçekleşir. Bu model, ağ sisteminin işlevlerini tanımlayan bir çerçevedir ve ağ oluşturan yazılım ve donanımın birlikte çalışabilirliğini desteklemek için gereken kural ve gereksinimleri özetleyerek, farklı ağ bileşenlerinin bilgi işlem işlevlerini kategorilere ayırır. OSI modeli, yedi katmandan oluşur ve her katmanın belirli bir işlevi vardır.



# Modern bilgisayar katmanları

Aşağıdaki tablo, OSI modelinin katmanlarını ve işlevlerini özetler:

7. Uygulama Katmanı	Uygulama programlarına hizmet verir
6. Sunum Katmanı	Verilerin sunumunu sağlar
5. Oturum Katmanı	İletişim oturumlarını yönetir
4. Taşıma Katmanı	Verilerin taşınmasını sağlar
3. Ağ Katmanı	Verilerin yönlendirilmesini sağlar
2. Veri Bağlantı Katmanı	Verilerin iletimini sağlar
1. Fiziksel Katman	Verilerin fiziksel iletimini sağlar

Her katman, altındaki katmanın hizmetlerini kullanır ve üstündeki katmana hizmet verir. Bu sayede, verilerin iletimi sırasında her katmanın belirli bir işlevi yerine getirilir ve verilerin doğru bir şekilde iletilmesi sağlanır.

# 19) Abstraction, Encapsulation, Information Hiding Nedir?

Abstraction, yazılım geliştirme sürecinde önemli bir rol oynayan bir kavramdır. Soyutlama, bir işin ne yaptığını biliriz ancak nasıl yapıldığı ile ilgilenmeyiz, yani işimize yarayacak kadar bilgi sahibi oluruz ve problemlerimizi çözer geçeriz . C# programlama dilinde, soyutlama kavramı, karmaşıklığı yönetmek için kullanılır ve nesnenin diğer tüm nesne türlerinden ayıran temel özelliklerini belirtir, böylece izleyicinin bakış açısından açıkça tanımlanmış bir kavramsal sınır sağlar . Soyutlama, Abstract Class ve Interface gibi yapılardan yararlanarak gerçekleştirilir . Abstract Class, soyut sınıfların bir türüdür ve içinde soyut metotlar barındırabilir. Soyut metotlar, yalnızca imzadan ibaret olan metotlardır . Interface ise, bir sınıfın hangi metotları içermesi gerektiğini belirleyen bir yapıdır .

# Encapsulation

“Encapsulation” (kapsülleme), nesne yönelimli programlama (OOP) kavramlarından biridir. Bir sınıfın, içerisinde bulunan metotların ve değişkenlerin korunması işlemidir. Programlarımızı oluştururken bazı bilgilerin her yerden ulaşılmamasını ve değiştirilmemesini isteyebiliriz. Bu durumda bilgilerimizi bir kasa içerisinde saklamamıza olanak sağlayan “Encapsulation” kavramı, programlarımız üzerinde koruma ve esneklik sağlamak için kullanılır.

“Encapsulation” kavramını anlamlandırmak için örnek olarak, bir bilgisayarın açılış işlemini düşünebiliriz. Bilgisayarımızı açmak için güç tuşuna basmamız gerektiğini biliyoruz fakat bilgisayar açılırken arka plan üzerinde hangi işlemleri gerçekleştirdiğini bilmiyoruz. Sadece bu işlemlerin nasıl yapılacağını bilebiliriz. Bu işleme “Encapsulation” diyoruz.

# INFORMATION HIDING

“Information hiding” (bilgi gizleme), bilgisayar bilimlerinde bir programın tasarım kararlarının en çok değişebilecek olan kısımlarının diğer kısımlardan korunması prensibidir. Bu sayede, tasarım kararı değiştirildiğinde programın diğer kısımlarının geniş çaplı değişikliklerden korunması sağlanır.

Bir yazılım bileşeninin iç mekanizmalarının diğer bileşenlerden gizlenmesi, “Information hiding” prensibinin bir uygulamasıdır. Bu sayede bileşenin iç mekanizmaları değiştirildiğinde, bileşenin diğer kısımlarının etkilenmesi engellenir.

“Information hiding” kavramı, “encapsulation” (kapsülleme) kavramı ile sıklıkla birbirinin yerine kullanılır. İkisi arasındaki farklar konusunda herkes aynı fikirde değildir. Bazıları “information hiding” prensibinin bir prensip, “encapsulation” kavramının ise bir teknik olduğunu düşünür.

# 19.1) Inheritance, Polymorphism

## Nedir

### Inheritance

C# programlama dilinde, **Inheritance (Kalıtım)**, bir sınıfın başka bir sınıftan özelliklerini ve davranışlarını miras almasını sağlayan bir mekanizmadır. Bu sayede kod tekrarından kaçınabilir, kodun okunabilirliğini ve bakımını kolaylaştırabilir, ortak işlevselliği tek bir yerde toplayabilir ve kodun genişletilebilirliğini artırabiliriz. Inheritance, C# programlama dilinde karmaşıklığı yönetmek için kullanılır ve nesnenin diğer tüm nesne türlerinden ayıran temel özelliklerini belirtir, böylece izleyicinin bakış açısından açıkça tanımlanmış bir kavramsal sınır sağlar. Inheritance, Abstract Class ve Interface gibi yapılardan yararlanarak gerçekleştirilir. Abstract Class, soyut sınıfların bir türüdür ve içinde soyut metotlar barındırabilir. Soyut metotlar, yalnızca imzadan ibaret olan metotlardır. Interface ise, bir sınıfın hangi metotları içermesi gerektiğini belirleyen bir yapıdır.

# Polymorphism

C# programlama dilinde, **Polymorphism (Çokbiçimlilik)**, bir nesnenin farklı şekillerde davranabilmesini sağlayan bir yapıdır . Polymorphism, nesne yönelimli programlamanın temel yapı taşlarından biridir ve C# programlama dilinde karmaşıklığı yönetmek için kullanılır . Polymorphism, Inheritance (Kalıtım) ve Interface gibi yapılardan yararlanarak gerçekleştirilir .

Polymorphism, bir sınıfın başka bir sınıftan özelliklerini ve davranışlarını miras almasını sağlayan Inheritance yapısı ile birlikte kullanıldığında, türetilmiş sınıfların, taban sınıfın metotlarını kendi ihtiyaçlarına göre değiştirebilmesine olanak tanır . Bu sayede, aynı isme sahip farklı sınıflardan türetilmiş nesneler, aynı arayüzü kullanarak farklı işlevler gerçekleştirebilirler .

Polymorphism, Abstract Class ve Interface gibi yapılardan yararlanarak da gerçekleştirilebilir. Abstract Class, soyut sınıfların bir türüdür ve içinde soyut metotlar barındırabilir. Soyut metotlar, yalnızca imzadan ibaret olan metotlardır . Interface ise, bir sınıfın hangi metotları içermesi gerektiğini belirleyen bir yapıdır .

## 20) D.R.Y Nedir?

**D.R.Y. (Don't Repeat Yourself)**, yazılım geliştirme sürecinde sıkça kullanılan bir prensiptir. Bu prensibe göre, aynı kodun tekrar tekrar yazılması yerine, kodun bir kere yazılıp, gerektiği yerlerde çağırılması gerektiği savunulur . Bu sayede, kodun okunabilirliği ve bakımı kolaylaşır, kod tekrarından kaçınılır ve kodun genişletilebilirliği artar.

## 21) S.O.L.I.D

S.O.L.I.D. prensipleri, bir yazılım geliştiricinin Nesne Yönelimli Programlama (OOP) ile yazılım geliştirirken, geliştirdiği yazılımın esnek ve geliştirilmeye uygun olması için uyması gereken kurallar bütünüdür. Bu prensipler kodun daha temiz, yeniden kullanılabilir, sürdürülebilir ve daha önemlisi anlaşılabilir olmasını sağlar. S.O.L.I.D. prensipleri, beş tasarım ilkesinin baş harflerinden oluşur:



- Single Responsibility Principle (Tek Sorumluluk Prensibi): Bir sınıfın yalnızca bir işlevi olmalıdır.
- Open-Closed Principle (Açık Kapalı Prensibi): Bir sınıfın davranışı, değişiklik yapmadan genişletilebilir olmalıdır.
- Liskov Substitution Principle (Liskov Yerine Geçme Prensibi): Alt sınıflar, üst sınıfların yerine geçebilmelidir.
- Interface Segregation Principle (Arayüz Ayrımı Prensibi): Kullanılmayan arayüzler, uygulamalara yük olmamalıdır.
- Dependency Inversion Principle (Bağımlılık Tersine Çevirme Prensibi): Bağımlılıklar, soyutlamalara bağlı olmalıdır.

Bu prensipler, yazılımın daha esnek, yeniden kullanılabilir, sürdürülebilir ve daha anlaşılır olmasını sağlar.

## 22) CRUD Nedir?

CRUD (Create, Read, Update, Delete), veritabanı yönetiminde temel işlemleri ifade eden bir kısaltmadır . CRUD, veri depolamada kullanılan dört temel fonksiyondur ve programlamada oluşturma, okuma, güncelleme ve silme işlemlerini ifade eder . Bu işlemler, veritabanındaki kayıtların yönetiminde sıkça kullanılır .

## 23) .NET Framework , .NET Core Nedir?

.NET Framework, Microsoft tarafından geliştirilen bir yazılım geliştirme platformudur . .NET Framework, Windows işletim sistemi üzerinde çalışır ve masaüstü ve sunucu uygulamaları oluşturmak için kullanılır . .NET Framework, Common Language Runtime (CLR) ve .NET Framework Class Library olmak üzere iki ana bileşenden oluşur . CLR, uygulamaların çalıştırılmasını sağlar ve iş parçacığı yönetimi, bellek yönetimi, tür güvenliği, özel durum yönetimi ve daha fazlası gibi hizmetler sağlar . Class Library, ortak işlevsellik için bir dizi API ve tür sağlar. Dizeler, tarihler, sayılar vb. için türler sağlar. Class Library, dosya okuma ve yazma, veritabanlarına bağlanma, çizim yapma ve daha fazlası için API'ler sağlar. .NET Framework uygulamaları, C#, F# veya Visual Basic programlama dillerinde yazılır. Kod, dil bağımsız Common Intermediate Language (CIL) haline getirilir. Derlenmiş kod, .dll veya .exe dosya uzantısına sahip dosyalarda saklanır. Bir uygulama çalıştırıldığında, CLR derlenmiş kodu alır ve JIT (just-in-time) derleyici kullanarak, kodu çalıştırmak için özgün bilgisayar mimarisine uygun makine koduna dönüştürür.

# .NET Core Nedir?

**NET Core**, Microsoft tarafından geliştirilen açık kaynak kodlu bir yazılım geliştirme platformudur . .NET Core, Windows, Linux ve macOS gibi farklı işletim sistemlerinde çalışabilir . .NET Core, sunucu uygulamaları oluşturmak için kullanılır ve Docker gibi konteyner teknolojileriyle birlikte kullanıldığında, mikro hizmet tabanlı uygulamalar için daha hafif bir alternatif sunar . .NET Core, .NET Framework ile uyumlu değildir ve .NET Framework'ta bulunan bazı özellikleri desteklemez . .NET Core, C#, F# veya Visual Basic programlama dillerinde yazılır ve kod, dil bağımsız Common Intermediate Language (CIL) haline getirilir. Derlenmiş kod, .dll veya .exe dosya uzantısına sahip dosyalarda saklanır. Bir uygulama çalıştırıldığında, .NET Core derlenmiş kodu alır ve JIT (just-in-time) derleyici kullanarak, kodu çalıştırmak için özgün bilgisayar mimarisine uygun makine koduna dönüştürür .

# .NET Framework ve .NET Core Farkları

.NET Framework ve .NET Core, Microsoft tarafından geliştirilen iki farklı yazılım geliştirme platformudur. .NET Framework, yalnızca Windows işletim sistemi üzerinde çalışırken, .NET Core, Windows, Linux ve macOS gibi farklı işletim sistemlerinde çalışabilir. .NET Core, açık kaynak kodlu bir platformdur ve Docker gibi konteyner teknolojileriyle birlikte kullanıldığında, mikro hizmet tabanlı uygulamalar için daha hafif bir alternatif sunar. .NET Framework, Windows masaüstü ve sunucu uygulamaları oluşturmak için kullanılırken, .NET Core, sunucu uygulamaları oluşturmak için kullanılır.

# 24) C# Veri Tipleri Ve Örnekleri

C# programlama dilinde kullanılan veri tiplerinden bazıları şunlardır:

- **int**: Tam sayıları temsil eder. Örneğin, `int sayi = 5;` ifadesi, `sayi` adlı bir değişken oluşturur ve bu değişkene 5 değerini atar.
- **float**: Ondalıklı sayıları temsil eder. Örneğin, `float ondalikSayi = 3.14f;` ifadesi, `ondalikSayi` adlı bir değişken oluşturur ve bu değişkene 3.14 değerini atar.
- **double**: Çift hassasiyetli ondalıklı sayıları temsil eder. Örneğin, `double pi = 3.14159265359;` ifadesi, `pi` adlı bir değişken oluşturur ve bu değişkene 3.14159265359 değerini atar.
- **bool**: Mantıksal değerleri temsil eder. Örneğin, `bool dogruMu = true;` ifadesi, `dogruMu` adlı bir değişken oluşturur ve bu değişkene `true` değerini atar.
- **string**: Metinleri temsil eder. Örneğin, `string isim = "Ahmet";` ifadesi, `isim` adlı bir değişken oluşturur ve bu değişkene "Ahmet" değerini atar.

# **25) Recursive, Overloading, Public, Private, ArrayList st Nedir**



# Recursive

Recursive, bir fonksiyonun kendisini çağırmasıdır . Bu, bir problemin alt problemlere ayrılması ve her alt problemin aynı işlemi yaparak çözülmesi için kullanılır . Özyinelemeli fonksiyonlar, bir dizi işlemi tekrar tekrar yapmak yerine, bu işlemleri daha küçük parçalara ayırarak çözerler. Özyinelemeli fonksiyonlar, özellikle ağaç yapıları, grafikler ve matematiksel problemler gibi tekrar eden yapılarla çalışırken kullanışlıdır.

Özyinelemeli fonksiyonlar, bir çıkış koşulu belirleyerek ve her çağrıda bu koşulu kontrol ederek sonlanır. Bu, fonksiyonun sonsuz döngüye girmesini önler. Özyinelemeli fonksiyonlar, kodun daha okunaklı ve anlaşılır olmasını sağlar, ancak aynı zamanda daha fazla bellek kullanımına neden olabilir.

# Recursive Örnek

Özyinelemeli fonksiyonlar, C# programlama dilinde kullanılabilir. Örneğin, aşağıdaki özyinelemeli fonksiyon, bir sayının faktöriyelini hesaplar:

```
static int Faktoriyel(int sayi)
{
    if (sayi == 0)
    {
        return 1;
    }
    else
    {
        return sayi * Faktoriyel(sayi - 1);
    }
}
```

# Overloading

Overloading, C# programlama dilinde aynı isme sahip birden fazla fonksiyonun tanımlanmasıdır . Bu, aynı işlevi farklı parametrelerle çağırmak için kullanılır . Overloading, kodun daha okunaklı ve anlaşılır olmasını sağlar ve aynı zamanda kodun yeniden kullanılabilirliğini artırır .

# Overloading Örnek

Örneğin, aşağıdaki örnek, iki farklı Topla fonksiyonu tanımlar:

```
static int Topla(int sayi1, int sayi2)
{
    return sayi1 + sayi2;
}
```

```
static double Topla(double sayi1, double sayi2)
{
    return sayi1 + sayi2;
}
```

# Public

**public**, C# programlama dilinde bir erişim belirleyicisidir. Public erişimi, en geniş erişim seviyesidir ve herhangi bir sınıf veya nesne tarafından erişilebilir . Public erişim belirleyicisi, bir sınıfın veya sınıf üyesinin herhangi bir yerden erişilebilir olmasını sağlar . Public erişim belirleyicisi, diğer erişim belirleyicileri olan private, protected ve internal ile birlikte kullanılabilir .

# Private

**private**, C# programlama dilinde bir erişim belirleyicisidir. Private erişimi, en kısıtlı erişim seviyesidir ve yalnızca tanımlandığı sınıf veya yapı içinde erişilebilir . Private erişim belirleyicisi, bir sınıfın veya sınıf üyesinin sadece tanımlandığı sınıf veya yapı içinde erişilebilir olmasını sağlar . Private erişim belirleyicisi, diğer erişim belirleyicileri olan public, protected ve internal ile birlikte kullanılabilir .

# ArrayList

ArrayList, C# programlama dilinde kullanılan bir koleksiyon sınıfıdır. ArrayList, dizilere benzer, ancak dinamik bir yapıya sahiptir. Dizilerde, dizinin boyutu bildirimi sırasında belirlenirken, ArrayList'te boyut önceden belirlenmez. ArrayList, herhangi bir veri türünden öğeleri depolayabilir ve öğeler, herhangi bir zamanda koleksiyona eklenebilir veya koleksiyondan kaldırılabilir.

# ArrayList Örnek

ArrayList sınıfı, `System.Collections` ad alanında tanımlanır ve aşağıdaki gibi kullanılabilir:

```
using System.Collections;
```

```
ArrayList liste = new ArrayList();  
liste.Add("Ahmet");  
liste.Add(42);
```

Bu örnekte, `ArrayList` sınıfından bir `liste` nesnesi oluşturulur ve `Add` yöntemi kullanılarak “Ahmet” ve 42 öğeleri koleksiyona eklenir.



# Ön Bellek Nedir?

Önbellek (cache), bilgisayar belleğinde sık kullanılan verilerin geçici olarak saklandığı bir alandır . Önbellek, veri erişim hızını artırarak, daha hızlı yanıt veren uygulamalar oluşturmak için kullanılır . Önbellek, bir web sayfasının veya uygulamanın daha hızlı yüklenmesini sağlayabilir. Önbellek, aynı verilerin tekrar tekrar yüklenmesini önleyerek, internet trafiğini azaltabilir ve bant genişliği tasarrufu sağlayabilir .

Önbellek, birçok farklı alanda kullanılır. Örneğin, web tarayıcıları, web sayfalarını daha hızlı yüklemek için önbellek kullanır. İşletim sistemleri, sık kullanılan dosyaları önbelleğe alarak daha hızlı erişim sağlar. Veritabanları, sık kullanılan verileri önbelleğe alarak daha hızlı erişim sağlar .

# UI Nedir

UI (User Interface), bir kullanıcının bir bilgisayar programı veya web sitesi gibi bir uygulamayla etkileşim kurduğu arayüzdür . UI, bir uygulamanın kullanılabilirliğini ve kullanıcı deneyimini belirleyen önemli bir faktördür . İyi tasarlanmış bir UI, kullanıcıların uygulamayı daha kolay ve verimli bir şekilde kullanmasını sağlar .

UI, birçok farklı bileşenden oluşur. Örneğin, bir web sitesindeki UI, HTML, CSS ve JavaScript gibi bileşenlerden oluşabilir. Bir masaüstü uygulamasındaki UI, pencere, düğme, menü ve diğer bileşenlerden oluşabilir .

# Regex Nedir

Regex (Regular Expressions), bir metin içindeki belirli kalıpları tanımlamak için kullanılan bir dildir . Regex, bir metnin içindeki belirli karakterleri veya karakter dizilerini bulmak için kullanılır . Regex, birçok farklı programlama dilinde kullanılabilir ve C# programlama dilinde de desteklenir .

Regex, birçok farklı karakteri tanımlamak için özel karakterler kullanır. Örneğin, . karakteri, herhangi bir karakteri temsil ederken, \* karakteri, bir karakterin sıfır veya daha fazla tekrarını temsil eder .

Regex, birçok farklı amaç için kullanılabilir. Örneğin, bir metindeki e-posta adreslerini veya telefon numaralarını bulmak için kullanılabilir .

## 29) npm,npx,create,push,pop,yarn Nedir?

npm (Node Package Manager), JavaScript programlama dilinde kullanılan bir paket yöneticisidir . npm, JavaScript kütüphanelerinin, araçlarının ve uygulamalarının paylaşılmasını ve yeniden kullanılmasını kolaylaştırır . npm, açık kaynaklı bir platformdur ve dünya genelinde milyonlarca geliştirici tarafından kullanılmaktadır .

npm, Node.js ile birlikte kullanılır ve Node.js'in yüklenmesiyle birlikte otomatik olarak yüklenir . npm, birçok farklı paket türünü destekler ve paketler, npm'in resmi web sitesinde bulunan npm Registry'den indirilebilir .

# npx

**npx**, Node.js paketlerini çalıştırmak için kullanılan bir araçtır . npx, Node.js'in bir parçasıdır ve Node.js'in sürüm 5.2.0'dan itibaren varsayılan olarak yüklenir .

npx, Node.js paketlerini yüklemek veya çalıştırmak için kullanılır. npx, paketleri yerel olarak yükler ve yalnızca bir kez kullanılacaksa, yüklemek yerine doğrudan çalıştırılmasını sağlar . Örneğin, `npx create-react-app my-app` komutu, `create-react-app` paketini yükler ve `my-app` adlı bir React uygulaması oluşturur .

npx, ayrıca birçok farklı paket türünü destekler ve paketler, npm'in resmi web sitesinde bulunan npm Registry'den indirilebilir .

# create

create, bir programlama terimi değildir. Ancak, birçok programlama dilinde nesne veya sınıf örneği oluşturmak için new anahtar kelimesi kullanılır. Örneğin, C# programlama dilinde `MyClass myObject = new MyClass();` ifadesi, MyClass sınıfından bir örnek oluşturur ve bu örneği myObject adlı bir değişkene atar.

# Push ve Pop

Push ve pop, bir veri yapısı olan yığın (stack) üzerinde yapılan işlemlerdir . Yığın, verilerin üst üste konulduğu bir yapıdır ve en son eklenen veri, yığının en üstünde yer alır . Push işlemi, yığının en üstüne yeni bir veri eklemek için kullanılırken, pop işlemi, yığının en üstündeki veriyi kaldırmak için kullanılır .

# Push, Pop Örnek

Örneğin, bir yığın oluşturmak ve bu yığına push ve pop işlemleri uygulamak için C# programlama dilinde aşağıdaki kod kullanılabilir:

```
using System;
using System.Collections;

class Program
{
    static void Main(string[] args)
    {
        Stack yigin = new Stack();

        yigin.Push(1);
        yigin.Push(2);
        yigin.Push(3);

        Console.WriteLine("Yiginin en ustundeki eleman: " + yigin.Peek());

        yigin.Pop();

        Console.WriteLine("Yiginin en ustundeki eleman: " + yigin.Peek());
    }
}
```



# yarn

Yarn, **JavaScript** paket bağımlılık yöneticisidir. Yarn, yüklenen her paketi önbelleğe alan, böylelikle paketin tekrar yüklenmesine gerek kalmayan, kaynak kullanımını en üst düzeye çıkarmak için gerçekleştirilen işlemleri paralelleştiren bir paket yöneticisidir . Yarn, npm'e ek olarak, yüksek performans ve tutarlılık sunar . Yarn'ın kurulumu için sistem seviyesinde bir başka paket yöneticisi kullanılabilir. macOS için Brew veya MacPorts tercih edilebilir. Yarn kurulumu için komut satırı üzerinden curl aracılığıyla indirme işlemi gerçekleştirilebilir . Yarn hakkında daha fazla bilgi için Yarn resmi web sayfası ve/veya GitHub sayfası incelenebilir .

# SEO Nedir?

SEO, **arama motoru optimizasyonu** anlamına gelir. SEO, web sitelerinin arama motorlarında daha çok görünürlülük elde etmesini sağlamak için yapılan çalışmalar bütünüdür . SEO, web sitelerinin arama motorlarında daha iyi performans göstermesi için yapılan çalışmalarının bir birimidir . SEO'nun amacı, web sitelerini Google ya da hedeflenen arama motorunda kelime bazlı üst sıralara çıkarmaktır . SEO, web sitelerinin arama motorlarında daha iyi performans göstermesi için yapılan çalışmaların bir birimidir.