

İNTERNET

İnternet bir bilgisayar veya herhangi bir elektronik aygıtın; dünya üzerindeki milyarlarca bilgisayar veya elektronik aygıtlarla iletişim kurmasını, bilgi paylaşmasını sağlayan ortak küresel çaplı ağa verilen isimdir. İnternet ortamında veri aktarımları, ağ ortam standartlarının belirlendiği çeşitli protokollerle gerçekleştirilmektedir. İnternet ağı üzerinde bilgi aktarımı TCP/IP protokolüyle gerçekleştirilmektedir. Günümüzde internet sayesinde haberleşme alışkanlıkları değişmiş, bilgiye erişim daha ucuz ve kolay bir yapıya bürünmüştür. Bilgiye erişime sağlanan kolaylık ve içeriklerin internet ortamında paylaşımıyla, teknolojik gelişmelerde hızlı bir ilerleme sürecine geçilmiştir.

İNTERNETİN ORTAYA ÇIKIŞI

Günümüz internetinin temelleri, ABD Savunma Bakanlığının birimler arası haberleşmeyi sağlama amacıyla verdiği desteklerle atılmıştır. İlk başlangıçta askeri amaçlı bir ağ hedefiyle oluşturulan projeler, geliştirildikten sonra zamanla bireysel kullanıcıların erişimine de açılmıştır. İnternet ağının ilk başlangıcı olarak 1969 yılında Kaliforniya Üniversitesi (UCLA) ile Stanford Araştırma Enstitüsü arasında kurulmuş olan internet ağ bağlantısı örnek verilebilir. ARPANET olarak adlandırılan bu proje zamanla geliştirilmiş ve bu projeye 1977 yılında önce TCP protokolü, sonra da IP protokolü eklenmesi bugünkü internet ağının gelişmesinde önemli bir yapı taşı olmuştur. 1989 yılında Tim Berners-Lee tarafından “Web” olarak ifade ettiğimiz “WWW” (World Wide Web) bilgi sistemi fikri ortaya atılmıştır. Bu bilgi sistemi HTML (Hypertext Markup Language) adlı metin işaretleme dilini temel almaktadır. Günümüz web sayfalarının da hala bu alt yapıyı kullandığı web ortamında daha sonraları arama motorları ve ticari içerikli sayfaların oluşumuyla beraber dünya çapında yaygın bir kullanıma ulaşmıştır.

TÜRKİYE'DE İNTERNETİN ORTAYA ÇIKIŞI VE YAYGINLAŞMASI

Türkiye'de internetin ortaya çıkışı ve günümüzdeki internetin yaygınlaşması süreci 1980'li yıllarda üniversitelerin ve TÜBİTAK'ın merkezde olduğu bir yapı ile başlanmıştır. İlk internet ağı oluşturma amacıyla Türkiye Üniversiteler ve Araştırma Kurumları Ağı (TÜVAKA) kurulmuştur. Bu kuruluşun desteklediği çalışmalarla 1987 yılında ilk geniş alan internet bağlantısı kurulmuştur. İlk geliştirilen geniş alan ağ bağlantısı daha sonra kaldırılarak; 1992 yılında Türkiye ile Hollanda arasında bir ağ bağlantısı kurulmuştur. Süreç içerisinde yaşanan gelişmelerle başlangıçta sadece üniversite ve kurumların dahil olduğu internet ağı, TURNET adlı internet sağlayıcı servisinin verdiği hizmetlerle bireysel kullanıcıların da erişimine açılmıştır. 2003 yılına gelindiğinde çevirmeli bağlantı yerine ADSL teknolojisi kullanılmış ve internet hızlarında önemli performans iyileştirmeleri sağlanmıştır. Günümüze geldiğinde hane halkında internet erişimi %92 seviyelerine ulaşmıştır. İnternet üzerinden ticari ürünler satın almalarında ise bu oran %44,3 seviyelerine ulaşmıştır.

TEMEL İNTERNET KAVRAMLARI

İnternet ortamında gezinti yaparken kullandığımız bazı temel internet kavramları bulunmaktadır. Kısa cümlelerle bu kavramları tanıtalım. Intranet/Extranet: Güvenlik amacıyla sınırlı özel amaçlı geliştirilmiş kapalı ağlardır. Extranet daha geniş bir ağı ifade eder. IP: İnternet ortamında her cihaza atanan adrestir. Statik ve dinamik IP adresleri bulunmaktadır. Bant Genişliği: İnternet ağındaki maksimum veri aktarım miktarını ifade etmektedir. WWW: Dünya çapında ağ olup, hiper metin belgelerinden oluşan içerik sisteminin adıdır. HTTP/HTTPS: HTTP, bir kaynaktan dağıtılan bilgi sistemleri için hiper metin aktarım protokolüdür. HTTPS bilgilerin şifrelendiği güvenli protokol versiyonudur. URL: Web sayfalarına ait adres bilgisini ifade etmektedir. DNS: Web ortamındaki her bir siteye erişim için IP adresi bulunmaktadır. Ancak IP adreslerini hafızada tutmak zor olduğundan URL adresleri ile erişim sağlanmaktadır. DNS sunucuları alan adlarını, IP adresleri ile eşleştirerek web sayfalarına erişim sağlarlar. Web Barındırma: Web sayfalarının içeriklerinin tutulduğu depolama hizmetinin adıdır. İstemci: İnternet ağına erişim talebinde bulunan son kullanıcıyı ifade etmektedir. Sunucu: Verilerin tutulduğu bilgisayar sistemidir. Vekil Sunucu: Kullanıcı ile erişim yapılmak istenen internet ağı arasında ara sunucu vazifesi görmektedir. Anonim erişim imkanı sunmakla beraber, performans iyileştirmesi de sağlayabilir. Ayrıca izleme ve filtreleme imkanıyla güvenli bir iletişim kurulmasını da sağlar. FTP: Dosya paylaşımına ve aktarımına imkan veren protokoldür. SMTP: E-posta göndermek için kullanılan protokoldür. IMAP: E-posta almak için kullanılan protokoldür. E-posta okuma sürecinde yerel bilgisayar üzerinden sunucuya erişim sağlanmakta ve mesajlar sunucu üzerinden okunmaktadır. POP: E-posta almak için kullanılan protokoldür. Bu protokol ile sunuculardan alınan

hizmette kullanıcı e-postalarına erişmek istediğinde sunucudan e-postalar bilgisayara indirilir.

İNTERNETİN GELİŞİMİ VE WEB

İnternet ağının ortaya çıkışıyla ve getirilen yeniliklerle web teknolojisi de çeşitli gelişim evrelerinden geçmiştir. Başlangıçta statik içeriklerin olduğu dönemden daha sonra dinamik içeriklere geçiş olmuştur. Daha sonra internetin evrimi ile web teknolojisindeki gelişmeler de hızlanmış ve bilgi çağının getirdiği yenilikler ve akıllı teknolojilerle çok daha ileri seviyelere doğru bir geçiş dönemi mümkün olmaktadır. Bilgilerin aktarılma ve kullanılma biçimine göre web teknolojisi Web 1.0, Web 2.0 ve Web 3.0 gelişme dönemleri olarak üç farklı dönem ile ifade edilmektedir. Web 1.0: Web döneminin başlangıç evresi olup en ilkel dönemi ifade etmektedir. Bu dönem statik içeriklerin bulunduğu, çoğunlukla okunma amaçlı kullanılan sade ve genellikle metin tabanlı bir tasarım dönemini ifade etmektedir. Web 2.0: Statik içeriklerden dinamik içeriklere geçişin olduğu bilgi paylaşmaya odaklı bir dönemdir. İçerik paylaşımı imkanı bulunması, zenginleştirilmiş ve etkileşimli ara yüz tasarımlarının bulunması, kullanıcılar arasında etkileşimin yoğun olması ve dinamik içeriklerle Web 1.0 döneminden ayrışmaktadır. Sosyal medyanın etkin olduğu bir dönemdir. Web 3.0: Web içeriklerinin son kullanıcıya hitap edilirken anlamsal ağ içeriklerinin de olduğu yeni dönemi ifade etmektedir. Bu dönemde yapılması planlanan merkezizsiz ağ yapılarıyla Web ortamında çok büyük yenilikler beklenmektedir. Yapay zekanın da etkin olacağı bu dönemde kullanıcılar arasında platformsuz içerik paylaşımlarının yapılması hedeflenmektedir. Her yerde bulunma felsefesinin hüküm sürmesi beklenen bu dönemde içeriklerin ara bir sunucu yerine doğrudan kullanıcılar arasında paylaşımı hedeflenmektedir.

WEB PROGRAMLAMA

Web 1.0 dönemi dünya çapında ağ üzerinde bulunan metin, görüntü, ses, video ve link bağlantıları içeriklerinden oluşmaktadır. Web tarayıcıları ile web sayfalarına erişim sağlanmaktadır. Web 1.0 döneminde web sayfası içerikleri statik içeriklerden oluşmaktaydı. Kullanıcı etkileşimi bulunmazdı. Web 2.0 dönemine geçişle beraber dinamik içeriklere erişimler sağlanmıştır. Böylece kullanıcılar web içerikleri ile etkileşim içerisinde olmuştur. Web 1.0 döneminde HTML ve CSS gibi dillerle kullanıcılar tarafından görünür kısımlarda yapılan değişiklikler ve bilgiler yer almaktaydı. Ancak Web 2.0 dönemine geçiş ile beraber ASP.NET ve PHP dilleri gibi programlama dilleri geliştirilmiş ve dinamik veri eklemeleri, veri güncellemeleri ve veri silinmesi süreçleri oluşmuştur. Bu tür veriler de veri tabanlarında güvenle saklanmaktadır. Bu süreçte verilerin tutulduğu sunucular ve sunucu hizmet yazılımları da önem kazanmıştır. Bunlarla birlikte sunucu tarafında dillerle uyumlu web geliştirme ortamları da web programlamanın geliştirici arasında pratik bir yönünü sağlamıştır. Web programlama betik diller, programlama dilleri, geliştirme araçları, veri tabanları ve sunucu yazılımları olarak bir bütün olarak ele alınmaktadır. Bu bütünlük içerisinde web programlama, yapılan işin büyüklüğü ve işlerin farklılığından dolayı ön yüz web geliştirme (istemci tarafı) ve arka yüz web geliştirme (sunucu tarafı) olmak üzere iki yönüyle ele alınmaktadır.

ÖN YÜZ WEB GELİŞTİRME

Ön yüz web geliştirme, web sayfasının son kullanıcıya hitap eden görünür kısımları olarak ifade edilmektedir. Ön yüz geliştirme bir web sayfası ya da uygulamasının istemci tarafı web geliştirme kısmı olarak ayrıca ifade edilmektedir. Bu bölümde web geliştiricileri, tasarımlarını iyileştirerek web sayfalarını kullanıcılar için ilgi çekici hale getirmeyi amaçlamaktadırlar. Ön yüz web geliştirme, geliştiriciler açısından iki yönüyle ele alınmaktadır. İlk yönü ön yüz web geliştirmede kullanılan betik ve stil dilleridir. Aynı zamanda bu dilleri temel alan daha gelişmiş özelliklere sahip bazı dillerdir. Ön yüz web geliştirmenin bir diğer aracı ise, ön yüz web geliştirme ortamları olarak kullanılan yazılımlardır. Bu tür geliştirme amaçları pratik, etkin ve kullanışlı olmaları yönüyle web geliştiricilerine yardımcı olmaktadır. Ön Yüz Web Geliştirme Dilleri Ön yüz web geliştirme dilleri olarak en temel dilleri incelediğimizde HTML, CSS ve Java Script dilleri karşımıza çıkmaktadır. Bu dillere ait kodlar web sayfasının görünür kısımlarında yer almaktadır. HTML, bir web sayfasının iskeletini oluşturma görevine sahiptir. CSS ile HTML'den bağımsız stiller oluşturularak kullanıcılar açısından web sayfası görünümü ilgi çekici hale getirilir. HTML ve CSS gibi diller tam olarak bir programlama dili olarak ifade edilmezler. Bu diller betik dil ve stil dili olarak ifade edilmektedir. JavaScript ise bu dillerden farklı olarak hem bir betik dil hem de bir programlama dili olarak ifade edilmektedir. Ön yüz web geliştirmede bu dillerin dışında JavaScript'i temel alan jQuery ve TypeScript dilleri geliştirilmiştir. CSS'i temel alarak geliştirilen Sass uzantı dili de ayrıca kullanışlı bir dildir. JQuery, TypeScript ve Sass gibi diller, JavaScript ve Sass dillerini daha etkin ve kullanışlı hale getirmek için geliştirilmiş dillerdir. Ön Yüz Web Geliştirme Ortamları Ön yüz web geliştirme dillerini ve özelliklerini daha etkin ve pratik şekilde kullanmak için bazı web geliştirme ortamları geliştirilmiştir. Bu diller hem yazım kolaylığı hem de tasarım kolaylığı sağlamaktadır. Çok sayıda web geliştirme ortamı bulunmakla beraber; en çok tercih edilen ön yüz web geliştirme ortamları olarak Sublime Text, Sketch, Angular, JS, Chrome DevTools, Visual Studio Code, Bootstrap, GitHub ve CodePen örnek olarak verilebilir.

ARKA YÜZ WEB GELİŞTİRME

Arka yüz web geliştirme dinamik içerikler web sayfalarında bulunmasıyla kendine web dünyasında yer edinmiştir. Arka yüz geliştirme yapılan işin çeşitliliğinden dolayı birçok yönüyle ele alınmaktadır. Öncelikle dinamik içerikler oluşturmak ve kullanma amacıyla web programlama dilleri bulunmaktadır. Arka yüz web programlama dillerinin etkin kullanımı için geliştirilen web geliştirme ortamları da arka yüz web geliştirmenin bir diğer yönüdür. Bir diğer yönü ise kullanıcıların etkileşimde bulunduğu dinamik içeriklerin saklandığı veri tabanları yazılımlarıdır. Arka yüz web geliştirmenin diğer bir yönü de, verilerin tutulduğu sunucularda programlama dilleri ve veri tabanları yazılımlarıyla uyumlu sunucu yazılımlarıdır. Arka Yüz Web Geliştirme Dilleri ve Dil Ortamları Arka yüz web geliştirme dilleri statik HTML ve CSS gibi betik ve stil dilleri yerine web programlama dilleri ile gerçekleştirilmektedir. Günümüzde dinamik web sayfaları oluşturmada en sık tercih edilen web programlama dilleri ve dil ortamları olarak ASP.NET, PHP, Java ve Python dillerini örnek olarak

verebiliriz. Arka Yüz Web Geliştirme Ortamları Sunucu tarafında geliştirme araçları olarak kullanan programlama dilleriyle uyumlu web geliştirme ortamlarından oluşmaktadır. Geliştiricilerin daha pratik şekilde kod yazabilmelerine olanak sağlamaktadır. Django, ASP.NET Core, Node.js, Laravel ve Java EE en sık kullanılan arka yüz web geliştirme ortamları olarak örnek verilebilir. Arka Yüz Web Geliştirme Veri Tabanları Sunucu tarafında kullanıcıların etkileşimde bulunduğu içerikler veri tabanlarında tutulmaktadır. Veri tabanlarına bilgiler eklenip, silinip, güncellenebilmektedir. Web üzerindeki verilerin veri tabanına işlenmesi süreçlerinden sorumlu popüler veri tabanı yönetim sistemleri olarak MongoDB, MySQL, PostgreSQL ve SQL Server örnek olarak verilebilir. Web Sunucu Yazılımları Web sunucuları web sayfalarına ait bilgileri depolama, işleme ve kullanıcılara sunma hizmetlerini sağlamaktadırlar. Web sunucu yazılımları vasıtasıyla bu işlemler gerçekleştirilirken; bu yazılımlar kullanıcıların içeriklere erişme süreçlerinin tamamından sorumludur. Apache HTTP Server, Internet Information Server ve NGINX popüler sunucu yazılımlarıdır.

WEB GELİŞTİRİCİ PROFİLLERİ

Günümüzde iş yükleri ve uzmanlıklara göre web geliştirici profilleri de farklılık gösterebilmektedir. İş yükünün fazla olduğu büyük ölçekli projelerde istemci tarafındaki web geliştirme kısmını ön yüz web geliştiricileri üstlenmektedir. Sunucu tarafındaki web geliştirme işlemlerini ise arka yüz web geliştiricileri üstlenmektedir. Proje büyüklüğünün küçük olduğu bazı web geliştirmelerinde ise tam yığın geliştiricileri diye tabir edilen web geliştiricileri, hem istemci tarafında hem de sunucu tarafında web geliştirme süreçlerinden sorumlu olmaktadır.

ASP NEDİR?

Web teknolojisinin ilk çıktığı dönemlerde web içerikleri statik içeriklerden oluşmaktaydı. Bu dönemde kullanıcı etkileşimi bulunmayıp web içerikleri metin, ses, görüntü vb. statik verilerinden oluşmaktaydı. ASP (Active Server Pages) ve benzeri dinamik içerikler üretilmesini sağlayan web teknolojileri ortaya çıktıktan sonra, web ağı da çok hızlı bir şekilde büyümüştür. Sunucu taraflı geliştirilen bu içeriklerle bilgi paylaşımı platformları da önemli bir artış göstermiştir. ASP, Türkçe kelime karşılığı olarak “Aktif Sunucu Sayfaları” anlamına gelmektedir. “Aktif” kelimesi ile kastedilen içeriklerin statik olmayıp dinamik içeriklerden oluşmasıdır. “Sunucu Sayfaları” kavramı da içeriklerin dinamik olması ve web geliştiricileri tarafından bu içeriklerin sunucu tarafında çalıştırılmasını ifade etmektedir. ASP, Microsoft tarafından 1996 yılında IIS 3.0 (Internet Information Server) teknolojisinin ortaya çıkışıyla beraber sunulmuştur. ASP, sunucu tarafında verileri işleyen bir betik motoru içermektedir. Bu betik motor, sunucu tarafında dinamik içerikli ASP dosyalarını işleyerek istemcilere iletmektedir. ASP, bir programlama dili olmayıp farklı betik dillerin çalıştırılabildiği bir dil ortamıdır. ASP dosya yapısında, hem sunucu taraflı hem de istemci taraflı çalışan kodlar bulunmaktadır. Bu dosya yapısı içerisinde istemci tarafında çalışan kodların bulunduğu HTML, CSS kodları gibi kodlar ile sunucu tarafında çalışan betik dillere ait kodlar bulunmaktadır. ASP sayfalarında betik dillere ait kodlar, HTML kodları içerisinde yer almaktadır. Betik dile ait kodlar “” şeklinde ASP etiketleri arasında yazılmaktadır. ASP ilk olarak 1996 yılında IIS 3.0 (Internet Information Server) teknolojisinin ortaya çıkışıyla web geliştiricilerine dinamik içerikler oluşturmak için sunulmuştur. 1997 yılına gelindiğinde IIS 4.0 ile ASP teknolojisi 2.0 sürümüne yeni nesne modelleri eklenerek geçiş yapılmıştır. 2000 yılından itibaren ise günümüzde de geçerliliğini koruyan ASP 3.0 teknolojisi sunulmuştur. ASP 3.0 ile yeni akış kontrol yetenekleri, hata işleme ve yeni ASPError nesnesi, kodlanmış asp betikleri, sunucu betikleri, performansı artırılmış aktif sunucu bileşenleri ve performans iyileştirmeleri yapılmıştır. ASP dosyalarının çalışmasını ele aldığımızda bir web tarayıcıdan ASP içerikli bir sayfa sunucu tarafına iletilir. Sunucu tarafında ASP motoru gelen ASP sayfasını satır satır işler ve istemci tarafında görüntülenebilmesi için HTML çıktısı oluşturulur. Oluşturulan HTML çıktı sayfası web tarayıcısına iletilir ve tarayıcı ekranında görüntülenir. ASP uzantılı dosyalar, yerel bilgisayarda ve sunucu bilgisayarlarda çalışabilmektedir. ASP sayfalarını çalıştırabilmek için IIS (Internet Information Server) sunucu yönetim yazılımı kullanılmaktadır. Tüm windows işletim sistemlerinde IIS desteği bulunmaktadır. Windows işletim sistemlerinde IIS pasif durumda olup, uygulamaları çalıştırabilmek için Windows işletim sistemi ayarlarından web geliştiricilerinin IIS’yi aktif hale getirmesi gerekmektedir.

ASP NESNELERİ

ASP, temelde nesne modelleri üzerine kurulmuş bir dil ortamıdır. Gücünü nesne modellerinin özelliklerinden alan ASP’nin, 1997 yılında 2.0 sürümünün gelişiyle beraber; günümüzde de sıklıkla kullanılan temel ASP nesneleri ortaya çıkmıştır. Bu nesnelere Response, Request, Application, ASPError, Session ve Server nesneleridir. Response nesnesi kullanıcıya bilgi göndermek için kullanılmaktadır. Application nesnesi sunucu tarafında global değişkenleri depolamak için kullanılan bir nesnedir. Request nesnesi, tarayıcı tarafından sunucuya gönderilen bilgileri almaktadır. ASPError nesnesi, hataların yönetimi ile ilgilenen nesnedir. Session nesnesi, herhangi bir kullanıcıya ait oturum bilgilerini saklamak amacıyla kullanılmaktadır. Server nesnesi ile sunucuda bulunan yöntem ve özelliklere erişim sağlanmaktadır.

ASP BETİK DİLLERİ

Herhangi bir programlama dilinde yazılmış ve çalıştırılmaya hazır kod bloklarının oluşturulduğu diller, betik dil olarak ifade edilmektedir. ASP bir programlama dili olmayıp farklı betik dillerini çalıştırabilen sunucu taraflı bir dil ortamıdır. Bu dil ortamında sunucu taraflı işlemleri gerçekleştirmek için; başlangıçta Microsoft tarafından geliştirilmiş olan VBScript betik dili tercih edilmiştir. Ancak daha sonra ASP’de farklı betik diller ile de kodlar çalıştırılabilmesi mümkün olmuştur. ASP dil ortamı, günümüzde JavaScript, PerlScript ve Python dillerini de desteklemektedir. VBScript, nesne tabanlı bir betik dil olup, basit bir sözdizimi yapısı ile öğrenmesi ve uygulaması kolay bir dildir. JavaScript, Netscape firması tarafından geliştirilmiş, web sayfalarının davranışlarını programlama amacıyla kullanılmaktadır. PerlScript betik dili ASP sayfalarında kod yazılabilen bir diğer betik dildir. ActiveState firması tarafından geliştirilen PerlScript betik motoru kullanılarak; PerlScript betik dili ile

yazılan kodların ASP’de çalıştırılabilmesi mümkündür. ASP 2.0 ile gelen Component Object Model (COM) betik dil desteği ile Python dili kullanılarak ASP sayfaları oluşturmak mümkündür. Python diline ait kodları ASP web sayfasında çalıştırabilmek için; Python ActiveX betik motoruna ihtiyaç bulunmaktadır.

ASP İLE WEB PROGRAMLAMA

ASP içerisinde VBScript, JavaScript, PerlScript ve Java kodları yazılıp çalıştırılabilmektedir.

VBScript ASP içerisinde standart olarak tercih edilen bir dil olduğu için; ASP ile yazılan programlama olayları bu kitap bölümünde VBScript betik dilindeki sözdizimsel ifadelerle anlatılmıştır. VBScript dilinde değişkenler doğrudan tanımlanabildiği gibi “Dim” ifadesiyle de tanımlanabilmektedir. “Dim” ifadesi ile çok boyutlu dizi değişkenleri de tanımlanabilmektedir. Koşul ifadeleri, döngüler ve prosedürler diğer programlama dillerinde olduğu gibi VBScript dilinde de benzer çalışma yapısına sahip olup; sözdizimsel olarak farklılıklar içermektedir. Koşul ifadeleri olarak if...then...else, If...Then...ElseIf, Select Case ifadeleri kullanılmaktadır. Tekrarlayan kod bloklarını çalıştırmak için; döngü ifadeleri olarak For...Next Loop, For Each...Next Loop ve Do...Loop yapıları kullanılmaktadır. Prosedür olarak ise iki farklı yapı bulunmaktadır. Değer döndürmeyen prosedür olarak Subroutine yapısı kullanılırken, değer döndüren prosedür olarak ise Function yapısı kullanılmaktadır.

ASP’DE VERİ TABANI İŞLEMLERİ

ASP dil ortamında dinamik içerikleri işlemek ve almak için veri tabanlarından yararlanılmaktadır. Veri tabanlarında verileri ekleme, silme ve değiştirme gibi bazı temel işlemleri gerçekleştirmek için; ASP dil ortamında ADO (ActiveX Data Objects) nesnelere dayanarak yararlanılmaktadır. ADO nesne yapısı sayesinde verilere erişmek için çok sayıda bileşen nesne modeli (COM) içermektedir. Bu bileşenler web geliştiricilerine birçok kolaylık sağlamaktadır. ASP web sayfalarından veri tabanlarına erişim sağlamak ve işlemler gerçekleştirmek için veri tabanlarında bir ADO bağlantısı oluşturulur ve açılır. Daha sonra Recordset nesnesinden bilgiler alınır. Gerekli işlemler tamamlandıktan sonra Recordset ve veri tabanı bağlantısı sonlandırılır.

.NET Platformunun Doğuşu ve Vizyonu

.NET ilk olarak Microsoft tarafından Temmuz 2000'de duyurulan bir platformdur. .NET platformu, 1990'ların sonlarına kadar kullanılmakta olan teknolojileri birbirine entegre eden, Windows hizmetlerine ve API'lerine yönelik yeni bir programlama arabirimine sahip bir uygulama geliştirme çerçevesidir. COM+ bileşen hizmetleri, ASP Web geliştirme çerçevesi, XML ve nesne yönelimli tasarım, SOAP, WSDL ve UDDI gibi yeni Web hizmet protokolleri için destek .NET çerçevesine dahil edilmiştir. .NET platformu geliştirme araçları, özelleştirilmiş sunucular, Web hizmetleri ve cihazlar olmak üzere dört ayrı ürün grubundan oluşmaktadır. Geliştirme araçları, C#.NET ve VB.NET gibi programlama dillerini; Visual Studio gibi uygulama dizi geliştirme araçlarını, Web ve Windows uygulamaları oluşturmak için kapsamlı bir sınıf kütüphanesini ve bu çerçevede oluşturulmuş nesnelere yürütmek için ortak dil çalışma platformunu ifade etmektedir. Microsoft .NET teknolojisi, bilgi işlem ve iletişim teknolojilerini birleştiren, yeni nesil gelişmiş bir yazılım platformudur. .NET, geliştiricilerin, işletmelerin ve tüketicilerin teknolojiyi kendi şartlarına göre kullanmalarını sağlar. .NET'in arkasındaki temel fikir, odak noktasının bireysel Web sitelerinden veya İnternet'e bağlı cihazlardan daha geniş, daha zengin çözümler sunmak için birlikte çalışan bilgisayar, cihaz ve hizmet kümelerine kaymasıdır.

.NET FRAMEWORK

Microsoft .NET, çeşitli programlama dillerinde geliştirilmiş uygulamaların yürütülmesi veya çalıştırılması için ortak bir platform sağlayan bir çerçevedir. .NET, nesneye yönelik programlama ortamı sağlamak, Windows tabanlı uygulamalar ve Web tabanlı uygulamalar gibi çeşitli uygulama türlerinin geliştirilmesi için ortam sağlamak ve .NET tabanlı kodun başka herhangi bir kodla bütünleşebilmesini sağlamak gibi amaçlar doğrultusunda geliştirilmiştir. Microsoft .NET Framework, çeşitli Microsoft Windows işletim sistemlerinde kullanılabilen, yazılımla uyumlu bir yapıdır. .NET Framework, Windows platformu için oluşturulan çoğu yeni uygulama tarafından kullanılmak üzere tasarlanmıştır. Çerçevenin temel sınıf kitaplığını oluşturan önceden kodlanmış çözümler, kullanıcı arabirimi, veri erişimi, veri tabanı bağlantısı, kriptografi, web uygulaması geliştirme, sayısal algoritmalar ve ağ iletişimi dâhil olmak üzere birçok alanda çok çeşitli programlama ihtiyaçlarını kapsar. .NET Framework'ün sunduğu avantajlar aşağıdaki gibi sıralanabilir:

- Tutarlı ve nesne yönelimli bir programlama ortamı.
- Kodun güvenli bir şekilde yürütülmesini destekler.
- Yazılım dağıtımı ve sürüm çakışmalarını en aza indirir.
- Birden çok cihazda çeşitli Windows tabanlı ve Web tabanlı uygulamalarda hem geliştiriciler hem de kullanıcılar için tutarlı bir şekilde kullanılmasını sağlar.
- .NET Framework'e dayalı kodun diğer herhangi bir kodla bütünleşebilmesini sağlamak için endüstri standartlarına dayalı iletişim sağlar.
- .NET, HTTP, XML ve SOAP gibi protokolleri içeren açık İnternet standartlarına dayanmaktadır.

.Net Framework Sürümleri ve Özellikleri

.NET Framework'ün her sürümü yeni özelliklerle güncellenmektedir. .NET 1.0, .NET Framework'ün ilk sürümüdür ve .NET Framework uygulamalarını çalıştırmak için ihtiyacınız olan her şeyi içerir. CLR, Sınıf Kitaplığı ve ASP.NET'i içerir. .NET 2.0, FTP desteği, yeni veri kontrolleri, gelişmiş kod arkası modeli ve önbelleğe alma özellikleri gibi ASP.NET'in tüm alanlarında önemli gelişmeler içermektedir. .NET 3.0, WPF, WCF, WWF ve Windows CardSpace bileşenlerini içerir. WPF esas olarak zengin istemci uygulamaları ve Web uygulaması pencereleri için kullanılır. Zengin bir kullanıcı arabirimi oluşturmak için Genişletilmiş Uygulama İşaretleme Dili'ni (Extended Application Markup Language-XAML) kullanır. Bu bileşen, donanım hızlandırmalı, 2B ve 3B grafik efektlerinin uygulanmasını destekler. WCF, sunucular ve istemciler arasında çalışan dağıtık, hizmet odaklı uygulamalar geliştirmeyi amaçlar. WWF, .Net uygulamalarında iş akışı olarak uygulamak için bir API ve bir süreç içi iş akışı motoru sağlar. Windows CardSpace, kullanıcıların kişisel bilgilerini yönetmelerine ve kontrol etmelerine yardımcı olur. Güvenli, güvenilir ve basit bir şekilde dijital bir kimlik sunarak kullanıcılara Web hizmetinde veya Web sitesi kimlik doğrulamasında yardımcı olmayı amaçlayan istemci yazılımıdır. .NET 3.5, modern uygulamaların oluşturulmasına yardımcı olmayı hedeflemektedir. Ayrıca ASP.NET, Ajax desteğini almıştır ve LINQ (Language –Integrated Query) ADO.Net tarafından erişilebilir hale gelmiştir. .NET 4.0, dağıtık veya çok çekirdekli sistemleri

hedefleyen paralel bilgi işleme desteğini etkinleştirmek için uzantılar içermektedir. .NET 4.5, Visual Basic, C# veya F# gibi programlama dillerinin birlikte kullanılarak Windows uygulamaları oluşturulmasını desteklemektedir. .NET 4.6, açık kaynaklı .NET Framework paketlerini içerir. .NET 4.7, güvenilirlik ve performans geliştirmeleriyle donatılmıştır. .NET 4.8, .NET Core 2.1'den taşınan JIT geliştirmelerini, WPF uygulamaları için yüksek DPI geliştirmelerini, erişilebilirlik iyileştirmelerini, performans güncellemelerini ve güvenlik geliştirmelerini içermektedir.

.NET'İN TEMELLERİ

.NET'in temel stratejisi yazılımı bir hizmet olarak etkinleştirmektir. .NET ayrıca dağıtık bilgi işleme, bileşenleştirme, kurumsal hizmetler, Web paradigmasındaki değişiklikler ve tecrübe kazanımına yanıt vermeye çalışmaktadır. Bileşenleştirme, farklı üreticiler tarafından geliştirilen yazılım bileşenlerinin entegrasyonunu ifade etmektedir. Kurumsal hizmetler, işlemlerin güvenliğini ve yönetilebilirlik için kod yazmadan kurumsal uygulamaların geliştirilmesine izin verilmesini ifade etmektedir. Web paradigmasındaki değişiklikler, Web uygulamalarının geliştirilmesini basitleştirmek için Web teknolojilerindeki değişiklikleri ifade eder. Tecrübe kazanımı, yazılım endüstrisinin büyük ölçekli kurumsal ve web uygulamaları geliştirirken kazandığı tecrübeleri ifade etmektedir. Microsoft .NET Framework'ün temel tasarım hedefleri, bileşen desteği, dil entegrasyonu, uygulamaların birlikte çalışabilirliği, basit geliştirme ve güvenlik olarak sıralanabilir. Bileşen desteği, geliştiricilerin kütüphane dosyası gibi bileşenleri uygulamalarına kolay bir şekilde entegre edebilmeleri ifade etmektedir.

ASP.NET

ASP.NET, Microsoft .NET Framework'e dayalı Web uygulamaları geliştirmeye yönelik bir sunucu tarafı teknolojisidir. ASP.NET'te, sayfadaki kod sunucu tarafından okunur ve ardından tarayıcıya gönderilen standart HTML/JavaScript/CSS oluşturmak için dinamik olarak kullanılır. ASP.NET kodunun işlenmesi sunucuda gerçekleştiğinden, sunucu tarafı teknolojisi denilmektedir. ASP.NET aşağıdaki avantajlara sahiptir: • ASP.NET, esnek bir şekilde programlama dillerinin kullanılmasını sağlar. • ASP.NET sayfaları derlenir, yorumlanmaz. Her dinamik sayfa istendiğinde kodunuzu okuyup yorumlamak yerine, ASP.NET dinamik sayfaları sunucunun çok hızlı bir şekilde çalıştırabileceği binary dosyalar halinde derler. • ASP.NET, .NET Framework'ün işlevlerine tam erişime sahiptir. XML, Web Hizmetleri, veri tabanı etkileşimi, e-posta, düzenli ifadeler ve diğer birçok teknoloji desteği doğrudan .NET'te yerleşiktir. • ASP.NET, sayfalarınızdaki sunucu tarafı kodunu HTML düzeninden ayırmanıza olanak tanır.

.NET MİMARİSİ VE PROGRAMLAMA DİLLERİ

.NET Framework Ortak Dil Belirtimi (CLS), Temel Sınıf Kütüphaneleri (BCL) ve Ortak Dil Çalışma Zamanı (CLR) bileşenlerinden oluşmaktadır. CLS, .NET dil uyumlularının onaylanması gereken minimum standartları tanımlar. Böylece, bir .NET derleyicisi tarafından derlenen herhangi bir kod, .NET Framework ile birlikte çalışabilir. BCL, her türlü yazılım uygulamasını oluşturmak için kullanılacak türleri tanımlar. Ayrıca, temel sınıf kütüphaneleri, belirli bir bilgisayardaki dizin ve dosya sistemiyle etkileşime girmek, ilişkisel veri tabanlarıyla (ADO.NET aracılığıyla) iletişim kurmak vb. için türler sağlar. BCL, işlevsellik ve uygulamalardan oluşan tutarlı, nesne yönelimli bir kütüphanedir. CLR, uygulama yürütme için ortak çalışma zamanı sağlayan, dilden bağımsız bir geliştirme ve yürütme ortamıdır. CLR'nin temel amacı, programcı adına .NET nesnelerini yüklemek, bulmak ve yönetmektir.

.NET SINIF KÜTÜPHANELERİ

.NET Framework sınıf kütüphaneleri, Windows Software Development Kit'e (SDK) dâhil edilen sınıflar, arabirimler ve değer türlerinden oluşmaktadır. .NET Framework sınıf kütüphaneleri, aşağıdaki işlevleri destekleyen binlerce sınıf içerir:

- Temel ve kullanıcı tanımlı veri türleri
- İstisnaların işlenmesi için destek
- Giriş/çıkış ve akış işlemleri
- Temel sistemle iletişim
- Verilere erişim
- Windows tabanlı ara yüz uygulamaları oluşturma yeteneği
- Web-istemci ve sunucu uygulamaları oluşturabilme
- Web hizmetleri oluşturma desteği.

DİL ENTEGRASYONU

Dil entegrasyonu, tamamen farklı bir dilde oluşturulmuş bir koddan sınıf türetme, farklı bir dilde yazılmış kodun istisnalarını yakalama veya farklı dillerdeki polimorfizmin avantajlarını kullanma gibi konuları kapsamaktadır. Namespace, isim çakışmalarını azaltır. Arayüz, arabirimi ortaya çıkaran nesnelere tarafından uygulanması gereken yöntemleri ve özellikleri belirtir. Kapsülleme, nesne yönelimli dillerde, bir sınıfın tüm verilerini ve davranışını kapsüllemesine izin verir. Kalıtım, bir sınıfın, üst sınıfın uyguladığı zengin işlevselliği yeniden kullanılabilmesi için bir üst sınıftan miras almasına izin verir, böylece geliştirme çabasını ve programlama hatalarını azaltır. Polimorfizm,

geliştiricilerin, türetilmiş bir sınıf tarafından geçersiz kılınabilecek bir temel sınıfta davranışları belirlemesine veya uygulamasına izin verir. Bu çok güçlü bir özelliktir çünkü geliştiricilerin başvurulan çalışma zamanı nesnesine dayalı olarak doğru davranışı seçmesine olanak tanır. İstisna işleme, anlaşılması daha kolay kod yazılmasına izin verir. Tüm hataların ortak ve anlaşılır bir düzende yakalanmasını sağlar.

GİRİŞ

C#, C ve C++'dan türetilmiş basit, modern nesne yönelimli ve güvenli bir programlama dilidir. Visual C# .NET, Microsoft'un C# geliştirme aracıdır. Etkileşimli bir geliştirme ortamı, pencereler ve Web uygulamaları oluşturmak için görsel tasarım yardımcıları, derleyici ve hata ayıklayıcı içerir. Visual C# .NET, Visual Basic .NET, Visual C++ .NET ve Jscript komut dosyası dilini de içeren Visual Studio .NET ürününün bir parçasıdır. Bu dillerin tümü, ortak bir yürütme motoru ve geniş bir sınıf kitaplığı içeren Microsoft .NET Framework'e erişim sağlar. C#'ın modern tasarımı, en yaygın C++ programlama hatalarını ortadan kaldırır. C#, programcıyı manuel bellek yönetimi yükünden kurtaran otomatik çöp toplama, otomatik olarak başlatılan değişkenler ve tip güvenli değişkenler sağlayarak geleneksel C/C++ programcısını güçlendirir.

DEĞİŞKENLER

C#'ta değişken, belirli bir veri türüne ait veri değerini içeren yapılardır. Her değişkenin boyutunu, değişkende saklanabilecek değer aralığını ve değişkene uygulanabilecek işlemler kümesini belirleyen bir türü vardır. C#'taki temel değişken türleri int, double, char, string ve bool'dur. Sabitler Sabitler, mevcut değerlerinin üzerine yazmak istemediğimiz değişkenleri ifade etmek için kullanılır. Sabitleri tanımlamak için const anahtar sözcüğü kullanılır. const anahtar sözcüğü, bir değişkenin her zaman aynı değeri saklaması istenildiğinde kullanışlıdır.

OPERATÖRLER

Operatörler, derleyiciye belirli matematiksel veya mantıksal işlemleri gerçekleştirmesini söyleyen sembollerdir. C#'ta aritmetik operatörler, ilişkisel operatörler, mantıksal operatörler, bitsel operatörler, atama operatörleri gibi operatörler bulunmaktadır. Aritmetik Operatörler Aritmetik operatörler toplama, çıkarma, çarpma ve bölme gibi temel aritmetik hesaplamalar için kullanılır. İlişkisel Operatörler İlişkisel operatörler, iki işlenen değerinin eşit olup olmadığını belirlemek gibi iki değişken arasındaki ilişkiyi kontrol etmek için kullanılmaktadır. İlişkisel operatörler, yalnızca değişkenler arasındaki ilişki doğru olduğunda true değerini döndürür. Aksi takdirde false döndürür. Mantıksal Operatörler Mantıksal operatörler, AND, OR ve NOT gibi iki değişken arasında mantıksal işlem gerçekleştirmek için kullanılır. Mantıksal operatörler her zaman Boole ifadeleriyle (true veya false) çalışır ve Boolean değer döndürür. Bitsel Operatörler Bitsel operatörler, bitler üzerinde çalışır ve bit bit işlem gerçekleştirir. Boolean ve tamsayı veriler üzerinde bit düzeyinde işlemler gerçekleştirilebilir. Atama Operatörleri Atama operatörleri, bir değişkene yeni bir değer atamak için kullanılır ve bu operatörler yalnızca bir değişkenle çalışır. Operatörlerde İşlem Önceliği Operatör önceliği, bir ifadedeki terimlerin gruplandırılmasını ifade eder. Bu işlem, bir ifadenin değerlendirilmesini etkiler. Bazı operatörler diğerlerinden daha yüksek önceliğe sahiptir. Örneğin çarpma operatörü, toplama operatöründen daha yüksek önceliğe sahiptir.

DİZİLER

Diziler, aynı türdeki öğeleri sabit boyutlu ve sıralı bir şekilde depolayan veri yapılarıdır. Dizide bulunan öğelerin veri türleri char, int, double gibi herhangi bir geçerli veri türü olabilir. Dizinin uzunluğu, dizide bulunan öğelerin sayısını ifade eder. C#'da diziler için bellek tahsisi dinamik olarak yapılır. Diziler bir tür nesnedir. Bu nedenle önceden tanımlanmış işlevleri kullanarak boyutlarını bulmak kolaydır. Dizideki değişkenler sıralanmıştır ve her birinin 0'dan başlayan bir indeksi vardır. Dizilere, indeks numaraları kullanılarak aşağıdaki örnekte görüldüğü gibi değer ataması yapılabilir. Dizi elemanlarına indeksler kullanılarak erişilebilir. Dizi adından sonra [] operatörleri arasına erişilmek istenen indeks numarası yazılarak istenilen dizi elemanına erişilir. Dizideki eleman sayısını belirlemek için length metodu kullanılmaktadır. Tek Boyutlu Diziler Tek boyutlu dizilerde, dizi elemanlarını saklamak için yalnızca bir satır bulunmaktadır. Tek boyutlu dizinin tüm değerleri, 0'dan dizi boyutuna kadar sıralı olarak depolanır. Çok Boyutlu Diziler Çok boyutlu diziler, dizi elemanlarını depolamak için birden fazla satır içermektedir. Her satır uzunluğu aynı sayıda eleman içerdiği için C# dilinde dikkatli dizi olarak da bilinir.

KONTROL YAPILARI

Koşullu ifadeler, belirli koşulların karşılanması şartıyla bilgisayara belirli eylemleri gerçekleştirmesini söyleyen programlama dillerinin özellikleridir. Koşullu ifadeler, bilgisayara bazı koşullar verildiğinde vereceği karar konusunda talimat vermek için çeşitli programlama dilleri aracılığıyla kullanılır. Bu kararlar, programcının aklındaki fonksiyonlara bağlı olarak, yalnızca

önceden belirtilen koşullar doğru veya yanlış ise verilir. Söz dizimi yapısı bir programlama dilinden diğerine biraz farklılık gösterse de tüm programlama dillerinde koşullu ifade sözdizimi vardır. if Koşul İfadesi if koşul ifadesi, söz dizimi olarak açık ve kapalı parantezler içine karşılaştırılması veya kontrol edilmesi amaçlanan şartların yazıldığı bir yapıya sahiptir. if koşul ifadesi, parantez içindeki koşulun (veya koşulların) doğru mu yanlış mı olduğunu karşılaştırır. Doğru ise, if kod bloğu yürütülür. Yanlışsa, yürütme kontrol edilerek bir sonraki bloğa geçer. if, boolean koşulun değerine dayalı olarak yürütülecek bir ifade seçer. else if deyimini (veya Python'daki kullanımıyla elif), if deyimine benzer bir sözdizimine sahiptir ve ardından else if bloğu gelir. else if ifadelerinde koşullar yukarıdan aşağıya doğru kontrol edilir, eğer ilk blok true dönerse, ikinci ve üçüncü bloklar kontrol edilmez, ancak birinci if bloğu false dönerse ikinci blok kontrol edilir. Bu kontrol, bir blok gerçek bir sonuç döndürene kadar devam eder. else ifadesi, tüm programlama dillerindeki tüm koşullu ifadelerin varsayılan ifadesidir. Yani, tüm if ve else if koşulları false döndürdüğünde (yukarıdan aşağıya), son (varsayılan) else blok ifadesi yürütülür. switch İfadesi switch ifadelerinde, her blok bir break anahtar sözcüğü ile sonlandırılır. switch içindeki ifadeler case ile ifade edilir. switch ifadesi, kontrolü gövdesindeki case ifadelerinden birine ileterek birden çok seçimi ve numaralandırmayı işleyen bir kontrol ifadesidir. switch ifadesinde her durum, gerçek bir değer döndürülene kadar test edilir. Ardından kod yürütme işlemi, break anahtar sözcüğüne ulaşır. Tüm durumlar false döndürürse, varsayılan blok yürütülür.

DÖNGÜLER

Bir programlama dilinde döngü, ifadeleri yürütmek için değerlendirilecek koşulun sonucuna bağlı olarak bir ifadeyi veya bir dizi ifadeyi birden çok kez yürütmenin bir yoludur. Döngüler içinde ifadeleri yürütmek için sonuç koşulu doğru olmalıdır. Döngüler, giriş kontrollü döngüler ve çıkış kontrollü döngüler olmak üzere temel olarak iki kategoriye ayrılır. Giriş kontrollü döngüler, döngü gövdesinin başlangıcında test edilecek koşulun bulunduğu döngülerdir. while döngüsü ve for döngüsü giriş kontrollü döngülerdir. Çıkış kontrollü döngüler, döngü gövdesinin sonunda test koşulunun bulunduğu döngülerdir. Çıkış kontrollü döngülerde, döngü gövdesinin sonunda test koşulu mevcut olduğundan, döngü gövdesi en az bir kez değerlendirilecektir. do-while döngüsü çıkış kontrollü döngülere örnektir. while Döngüsü while döngüsü, belirtilen koşul false döndürene kadar bir kod bloğunu art arda yürütmek için kullanılır. while döngüsü, while anahtar kelimesiyle başlar ve parantez içinde true veya false döndüren bir boole koşullu ifade içerir. Belirtilen koşullu ifade false döndürene kadar kod bloğunu yürütür. do-while Döngüsü do deyimini, belirtilen bir ifade false olarak değerlendirilene kadar tekrar tekrar parantezler içine alınmış bir deyim veya bir deyim bloğunu yürütür. do-while yapısı bir süreç sembolü ve bir koşuldandır. İlk olarak, blok içindeki kod çalıştırılır ve ardından koşul değerlendirilir. Koşul doğruysa, blok içindeki kod yeniden yürütülür. Bu işlem, koşul yanlış olana kadar tekrarlanır. for Döngüsü for döngüsü, belirli sayıda yürütmesi gereken bir döngüyü verimli bir şekilde yazmanıza olanak tanıyan bir tekrar kontrol yapısıdır. for döngüsü, belirtilen bir ifade yanlış olarak değerlendirilinceye kadar bir ifadeyi veya bir ifade bloğunu tekrar tekrar yürütür. for döngüsü, diziler üzerinde yineleme yapmak ve sıralı işleme için kullanışlıdır. foreach Döngüsü Bir programlama dilinde döngü, değerlendirilecek bir koşulun sonucuna bağlı olarak bir deyim veya bir dizi deyimini birden çok kez yürütmenin bir yoludur. Döngüler içinde ifadeleri yürütmek için elde edilen koşul doğru olmalıdır. foreach döngüsü, koleksiyonun öğeleri üzerinde yineleme yapmak için kullanılır.

METOTLAR

Metotlar (fonksiyonlar), belirli eylemleri gerçekleştirmek için kullanılır. Metotlar, yalnızca çağrıldığında çalışan kod bloklarıdır. Örneğin, bir daire oluşturmak ve onu renklendirmek için bir program yazılacağı varsayıldığında, daireyi çizmek için bir metot, daireyi renklendirmek için ise başka bir metot kullanılır. Bu sayede karmaşık bir sorunu daha küçük parçalara bölerek, programın kolay anlaşılır ve yeniden kullanılabilir bir hale getirilmesi sağlanır. Metotlar geri dönüş tipi, metot adı ve metot gövdesi alanlarına sahiptir. **NESNEYE YÖNELİK PROGRAMLAMA**

Nesne yönelimli programlama, kavramları nesnelere temsil eden, veri alanları ve ilişkili metotlara sahip bir programlama mantığıdır. Genellikle sınıf örnekleri olan nesnelere, birbirleriyle etkileşim kurmak için kullanılır. C#'taki tüm kod ve veriler bir sınıfa dâhil edilmelidir. Bir sınıfın dışında bir değişken tanımlanamaz ve bir sınıfta olmayan herhangi bir kod yazılamaz. Sınıflar, sınıfın bir nesnesi oluşturulduğunda yürütülen kuruculara (constructors) ve sınıfın bir nesnesi yok edildiğinde yürütülen bir yıkıcıya (destructor) sahiptir. Kalıtım C#'ta kalıtım, bir nesnenin üst nesnesinin tüm özelliklerini ve davranışlarını otomatik olarak elde ettiği bir süreçtir. Bu şekilde, başka bir sınıfta tanımlanan özellikler ve davranışlar yeniden kullanılabilir, genişletilebilir veya değiştirilebilir. Süper sınıf, özellikleri miras alınan sınıf, süper sınıf (temel sınıf veya üst sınıf) olarak bilinir. Alt sınıf, diğer sınıfı miras alan sınıf, alt sınıf (türetilmiş sınıf veya genişletilmiş sınıf) olarak bilinir. Alt sınıf, üst sınıf alanlarına ve yöntemlerine ek olarak kendi alanlarını ve yöntemlerini ekleyebilir. Yeniden kullanılabilirlik, yeni bir sınıf oluşturmak istendiğinde ve istenilen kodun bir kısmını içeren bir sınıf zaten varsa, yeni sınıf mevcut sınıftan türetilir. Nesne Oluşturma Sınıfı, bir nesne için şablon tanımlar. Bir sınıf, kendisine ait olan nesnelere içeriğini (değişkenler, metotlar) tanımlar. Nesnelere, sınıfların örnekleridir. Nesnelere kendi sınıflarının davranışlarına sahiptir. Nesne, programların gerçek

bileşenidir. Sınıf ise örneklerin nasıl oluşturulduğunu ve nasıl davrandıklarını belirtir. Soyut Sınıflar ve Metotlar Veri soyutlama, belirli ayrıntıları gizleme ve kullanıcıya yalnızca temel bilgileri gösterme işlemidir. Soyutlama, soyut sınıflar ya da arabirimler aracılığı ile gerçekleştirilebilir. Soyut anahtar kelimesi, sınıflar ve yöntemler için kullanılabilir. Soyut sınıf, nesnel oluşturmak için kullanılmayan kısıtlı bir sınıftır. Soyut sınıflara erişmek için başka bir sınıftan miras alınması gerekir. Soyut yöntem yalnızca soyut bir sınıfta kullanılabilir ve bir gövdesi yoktur. Gövde, türetilmiş sınıf tarafından kalıtımla sağlanır. Soyut bir sınıf hem soyut hem de normal yöntemlere sahip olabilir. Kapsülleme Kapsülleme, verilerin tek bir birim altında toplanması olarak tanımlanır. Kodu ve manipüle ettiği verileri birbirine bağlayan bir mekanizmadır. Kapsülleme, verilere bu kalkanın dışındaki kodun erişmesini engelleyen koruyucu bir kalkandır. Kapsüllemeye, bir sınıfın değişkenleri veya verileri diğer herhangi bir sınıftan gizlenir ve yalnızca bildirildikleri kendi sınıfının herhangi bir üye işlevi aracılığıyla erişilebilir. Kapsüllemeye, bir sınıftaki veriler diğer sınıflardan gizlenir, bu nedenle veri gizleme olarak da bilinir. Polymorphism Polimorfizm (çok biçimlilik), bir sınıfa aynı ada sahip birden çok uygulamaya sahip olma yeteneği sağlar. C#'da statik (çalışma zamanı polimorfizmi) ve dinamik (çalışma zamanı polimorfizmi) olmak üzere iki tür polimorfizm vardır.

ASP.Net MVC ve Geliştirme Ortamı

ASP.Net MVC Bu bölümde ASP.Net MVC konusu ele alınmıştır. ASP.Net, Microsoft firması tarafından geliştirilen bir web uygulaması geliştirme çatısıdır. Uygulama geliştirme çatıları, bir platformda uygulama geliştirebilmek için gerekli olan kütüphaneleri, editörleri, programlama dillerini sunan yazılımlar bütünüdür. Bu çatılar sayesinde istenilen platformda daha hızlı uygulama geliştirmek mümkündür. ASP.Net ücretsiz olarak indirilip kurulabilir. Bu platformda yazılım geliştirmek için Visual Studio yazılımı kullanılabilir. Visual Studio yine Microsoft firmasının geliştirdiği tümleşik bir yazılım geliştirme ortamıdır. Bu yazılım ile beraber bir uygulamanın kodlama, test, hata ayıklama gibi tüm süreçleri gerçekleştirilebilmektedir. Uygulama aynı zamanda kod yazımında programcılara yardımcı olarak kolaylıklar sağlamaktadır. Visual Studio'nun Community sürümü ücretsiz olarak resmi web sayfasından indirilebilir. Bu bölümün yazıldığı 2022 yılında güncel sürümü Visual Studio 2022'dir.

Klasik ASP

Klasik ASP ASP.Net öncesinde ASP (Active Server Pages) ile kullanılmaktaydı ve hem sunucu hem de istemci taraflı kodlar aynı sayfa içerisine yazılmaktaydı. Tüm kodların aynı sayfaya yazılması, geliştirme ve hata ayıklama süreçlerinde karmaşaya neden olan bir işlemdi. Klasik ASP ile geliştirilen dosyaların uzantısı “.asp” idi. Sunucu ve istemci kodlarının aynı sayfalara yazılmasının getirdiği zorluklardan dolayı Microsoft, ASP.Net çatısını geliştirmiştir. Bu çatı birlikte bu kodlar birbirinden ayrılarak daha düzenli bir kod yazım ortamı oluşturulmuştur. ASP.Net çatısının ilk lanse edildiği dönemde, web uygulamaları geliştirmek için Web Form ortamı tanıtılmıştır.

Web Form Uygulamaları

Web Form Uygulamaları Web Form uygulamaları, Windows form uygulamalarına benzer olarak olay tabanlı web uygulaması geliştirmeye olanak sağlayan web uygulaması geliştirme ortamıdır. Bu ortamın Windows form uygulamalarına benzer olması, masaüstü uygulama geliştiren yazılımcıların dikkatini çekerek web ortamında da hızlı bir şekilde uygulamalar geliştirmelerini sağlamıştır. Web Form ile geliştirilen sayfaların dosya uzantısı “.aspx” idi. Bu platformda sunucu taraflı kodlar, "code behind" ismi verilen teknikte arayüzü oluşturan istemci kodlarından ayrıştırılmıştır. Uygulamayı temel anlamda iki bölüme ayıran bu yöntem ile birçok başarılı web uygulaması geliştirilmiştir ancak uygulama geliştirilirken alınan bu performans, uygulamanın çalışması esnasında bazı sınırlılıklardan dolayı alınamamıştır. Örnek olarak

ASP.Net MVC Uygulamaları

web form uygulamalarında sayfada kullanılan kontrollerin durumlarının, sayfanın kaynak kodunda saklanması, oluşturulan kontrollerin sunucu kontrolleri olarak oluşturulmasından kaynaklı sunucunun render açısından iş yükün artması gibi sebepler gösterilebilir. ASP.Net MVC Uygulamaları Web Form uygulamalarının sonrasında, MVC tekniği ile uygulama geliştirme platformu sunulmuştur. MVC, Model-View-Controller kelimelerinin baş harflerinden oluşmaktadır ve bir uygulama geliştirme desendir. Microsoft bu deseni kendi ürettiği ASP.Net çatısına uyarlayarak yazılım geliştiricilere sunmuştur. Firmanın, ASP.Net MVC adını verdiği bu yeni geliştirme ortamı, yazılım geliştiricilere kodları çok daha iyi ayrıştırarak, test edilebilir, hata ayıklamasının daha kolay yapıldığı kodlar üreterek uygulama geliştirme imkanı sunmuştur. Günümüzde birçok web uygulamasının bu yöntem ile geliştirildiği görülmektedir. Controller Yapısı

Controller Yapısı

MVC yapısının bir parçası olan Controller'lar, kullanıcıdan gelen istekleri karşılayarak gerekli işlemleri içerisinde bulundurduğu action metod adı verilen metodlarla gerçekleştirmekte ve kullanıcıya View'lar aracılığı ile sonuçları göstermektedir. Bu işlemler sırasında verileri gösterirken gerektiğinde model katmanını da kullanmaktadır. Controller dosyalarının uzantısı “.cs” dir. Her controller bir class'dır.

View Yapısı

View'ler ise kullanıcıya işlenen verilerin gösterildiği katman olarak tanımlanmaktadır. View'ler içerisinde istemci taraflı HTML, CSS, JavaScript gibi kodlarla birlikte aynı zamanda sunucu taraflı kodlar da yazılabilmektedir. Bu kodların yazılabilmesi ve okunabilmesi Razor adı verilen yazım kuralları ile sağlanmaktadır. Bir View içerisinde sunucu taraflı bir kod yazılmak istendiğinde, “@” işareti kullanılmaktadır. Birden çok satırlı sunucu taraflı kod yazımlarında ise “@{” blok yapıları kullanılmaktadır.

Layout View Yapısı

Kullanıcıya her sayfada gösterilmek istenen verileri için Layout View adı verilen kavram kullanılmaktadır. Örneğin uygulamanın gezinme menüsü, Layout içerisinde kodlanarak her sayfada aynı şekilde gösterilebilir. Bunun getirdiği kolaylık ise, bir değişiklik yapılması gerektiğinde tek yerde yapılarak bu değişikliğin her sayfaya yansıtılmasıdır.

Model Yapısı

Model katmanında, uygulama içerisindeki verilerin bellekte temsil edilebilmesi için gerekli class'lar bulunmaktadır. Örneğin veri tabanındaki öğrenci tablosunda bulunan her öğrencinin uygulama içinde temsil edilebilmesi için gerekli olan öğrenci sınıfı, bu katman içerisine yazılmalıdır.

Controller Yapılarından View Yapılarına Veri Taşıma İşlemleri

Controller katmanı sunucu tarafı çalışan bir katman olduğundan, veri tabanları ya da diğer kaynaklardan elde ettiği verileri kullanıcıya göstermek için View'lere taşıyabilir. Controller'dan View'lere veri taşımak için, ViewData, ViewBag, TempData ve ViewModel yöntemleri kullanılabilir.

ASP.NET ve Veri Tabanı Dünyası

ASP.NET: VERİ TABANI ASP.Net Microsoft firmasının geliştirmiş olduğu bir web uygulaması geliştirme platformudur. Günümüzde birçok modern web uygulaması bu platform kullanılarak geliştirilmektedir. Kullanıcı etkileşimlerinin yoğun olarak gerçekleştirildiği web 2.0 uygulamalarında veri tabanı kullanımı kaçınılmazdır. Yazılım dünyasında birçok veri tabanı yönetim sistemi bulunsada, bu bölümde Microsoft tarafından geliştirilen ve sektörde öncü veri tabanı yönetim sistemlerinden olan MS-SQL kullanılarak bir uygulama geliştirilmiştir. Uygulamaların veri tabanlarına erişip temel veri işlemlerini yönetebilmesi için veri tabanı bağlantı kütüphanelerine ihtiyaçları bulunmaktadır. Veri erişimi için Microsoft tarafından geliştirilen kütüphane çatısına ADO.Net ismi verilmektedir. ADO.Net ile veri erişim ve temel işlemlerin gerçekleştirilebilmesi için gerekli olan kodlar yazılımcılara sunulmaktadır. Bu bölümde bir ASP.Net MVC uygulaması geliştirilmiş ve veri tabanı temel CRUD (Create-Read-Delete-Update) işlemleri için ADO.Net'in bir parçası Entity Framework kütüphanesinden faydalanılmıştır.

Veri Tabanı ve İlişkisel Yapılar

Veri Tabanı Bilgisayar ortamlarında verilerin sistematik olarak saklanması için veri tabanları kullanılmaktadır. Bilgisayar ortamında veriler birçok formatta saklanabilmektedir. Örneğin kullanıcıdan alınan veriler bir metin dosyasında da saklanabilmekle birlikte bu veriler üstünde sonradan sorgulamalar ve analizler yapmak zor olmaktadır. Verilerin sistematik olarak saklanması, veri tabanlarında oluşturulan tablolar aracılığı ile sağlanmaktadır. Tablo yapıları üstünde saklanan verilerde sorgulama işlemleri, istatikselsişlemler gibi birçok ihtiyaç duyulan işlem SQL sorgulama dili ile gerçekleştirilmektedir. Bir uygulamanın kullandığı ve saklanması gereken tüm varlıkları için veri tabanlarında tablolar oluşturulabilmektedir. Bu tablolar, ilişkisel veri tabanı sistemleri ile birbirleri ile ilişkilendirilerek verilerin daha düzenli ve tutarlı verilerin tutulmasını da sağlamaktadırlar. Örneğin öğrenciler ve sınıflar birbirlerinden ayrı varlıklar olarak tanımlanmaktadır. Bu durumda öğrenciler için bir tablo oluşturulurken, sınıflar için de ayrı bir tablo oluşturulmaktadır. Öğrencilerin tablosunda, her bir öğrencinin okuduğu sınıf, sınıflar tablosunda tanımlanan SınıfId değeri ile birlikte tutulmakta ve iki tablo birbiri ile ilişkilendirilmektedir. Tabloların ilişkili olması, veri bütünlüğü açısından önemlidir.

Entity Framework

Entity Framework Entity Framework, Microsoft tarafından geliştirilen açık kaynaklı bir veri erişim kütüphanesidir. Bu kütüphane ile bağlanılacak veri tabanı sistemi fark etmeksizin veri erişim işlemleri geliştirilebilmektedir. Bunun avantajı, uygulamanın kullandığı veri tabanı sonradan değiştirilse bile, uygulama kodlarında yapılacak değişikliklerin en az seviyede olmasıdır. Bununla birlikte herhangi bir SQL komutu yazmadan temel veri tabanı CRUD(Create-Read-Update-Delete) işlemlerinin yapılmasına olanak sağlamaktadır. Veri tabanı işlemleri her yazılımda gerçekleştirilen işlemlerdendir. Rutin olarak yapılan bu işlemlerin kısa sürede ve mümkün mertebe az kod yazarak gerçekleştirilebilmesi, yazılım geliştirme süreçlerinde önem arz etmektedir. Entity framework bu noktada yazılım geliştiricilere önemli avantajlar sağlamaktadır.

Entity Framework Yaklaşımları

Entity Framework Yaklaşımları Entity framework ile veri işlemlerinin yapılabilmesi için temel anlamda 2 yaklaşım bulunmaktadır. Bu yaklaşımlardan birinci code-first yaklaşımıdır. Code first yaklaşımında veri tabanı oluşturulmadan önce model sınıfları ve veri işlemlerinin yapılabilmesi için DbContext sınıfı yazılmaktadır. Bu sınıfların yazılmasının ardından, veri tabanı ve içindeki tablolar otomatik olarak Entity framewrok tarafından oluşturulmaktadır. Code First yaklaşımı ile oluşturulacak olan tablolar ve alanların özelleştirilebilmesi için Data Annotation Attribute'leri veya Fluent Api tekniği kullanılmaktadır. Her iki teknik de tabloların ve alanların özelliklerini belirlemek için kullanılmaktadır. Data Annotations ile özelleştirilmek istenen alanların üstüne Attribute'ler yazılırken, Fluent Api tekniğinde bu özelleştirme işlemleri DbContext sınıfı içinde bulunan OnModelCreating() metodu override edilerek zincirleme metodlar ile gerçekleştirilmektedir. Code first yaklaşımı ile veri tabanı nesnelere oluşturulurken SQL kullanmaya gerek olmadığından, programlama diline SQL'den daha hakim yazılımcıların kullanabileceği bir tekniktir. Entity framework kütüphanesinin ikinci temel yaklaşımı ise Database First yaklaşımıdır. Adından da anlaşılacağı üzere, geleneksel yazılım geliştirme modeline daha yakın olan bu yaklaşımda öncelikle veri tabanı oluşturulmaktadır. Projede

kullanılacak olan model ve DbContext sınıfları ise bu veri tabanı yapısına göre yine Entity Framework tarafından oluşturulmaktadır. Bu yaklaşımda SQL yapılarına hakim olan programcılar daha rahat kodlama yapabileceği ortam sağlanmaktadır. Birbirlerinin tam tersi şekilde çalışan bu iki yaklaşımlardan birisi ile veri tabanı işlemleri yapılan proje geliştirilebilmektedir. Yazılımcılar, kendi geliştirme alışkanlıklarına göre yaklaşımlardan birisini seçebilmektedirler.

Kurulum ve Proje Entegrasyonu

Her iki yaklaşımda da Entity Framework kütüphanesinin bir MVC projesinde kullanılabilmesi için NuGet Package Manager aracılığı ile projeye dahil edilmesi gerekmektedir. NuGet Package Manager ile dahil edilmesi gereken paketler, Microsoft.EntityFrameworkCore.SqlServer ve Microsoft.EntityFrameworkCore.Tools paketleridir. Bu paketler eklenirken uygulamanın sürümü dikkate alınarak paket sürümleri seçilmelidir. Microsoft.EntityFrameworkCore.SqlServer paketi ile temel veri tabanı işlemlerinin yapılabilmesi için sınıflar eklenirken, Microsoft.EntityFrameworkCore.Tools paketi ile veri tabanı işlemlerinin yapılabilmesi için gerekli olan komutların projeye dahil edilmesi sağlanmaktadır.

ASP.Net MVC Uygulaması Geliştirilmesi ve Şablonlar

Veri Tabanı İşlemleri Yapan ASP.Net MVC Uygulaması Geliştirilmesi ASP.Net MVC proje uygulamasında, veri tabanında yapılacak olan CRUD işlemlerinin hızlı bir şekilde yapılabilmesi için geliştirilmiş olan şablonlar bulunmaktadır. Bu şablonlar controller sınıfları ekleme sırasında seçilerek, entity framework ile veri işlemlerinin yapılması sağlanabilmektedir. Temel veri işlemlerinin yapılabilmesi için şablonlar aracılığı ile CRUD işlemlerinin View'leri de otomatik olarak oluşturulmaktadır. View'ler ile birlikte controller içerisinde bulunan metodlar ile veri işlemleri yapacak kodlar da oluşturulmaktadır. Bu yöntem temel veri CRUD (Create-Read-Update-Delete) işlemleri için geliştiricilere büyük kolaylıklar sağlamaktadır. Oluşturulan kodlar üstünde sonradan değişiklikler yapmak da mümkün olduğundan aynı zamanda esnek bir sistemdir. Bununla birlikte, yazılım geliştiriciler şablon kullanmadan yine entity framework kullanarak kendi veri erişim kodlarını da yazabilme imkanına sahiptir.

PHP'YE GİRİŞ

PHP dili web sayfaları geliştirmek için kullanılan sunucu taraflı dillerden en popüler olanlarından. Bu dilde yazılacak web sayfaları PHP uzantısı ile kaydedilir ve bir web sunucusu tarafından yorumlanmalıdır. Bunun için herhangi bir web sunucusuna ayrıca PHP yorumlama desteğinin eklenmesi gerekir. Bu bölümde PHP dilini çalıştırabilmek için öncelikle uygun bir web sunucusu kurulumu anlatılmış, ardından dilin komutları gösterilmeye çalışılmıştır.

PHP DESTEKLİ WEB SUNUCUSU KURULUMU

PHP öğrenmeye çalışan çoğu öğrenci için XAMPP paketinin kullanımı önerilmektedir. Ücretsiz olan paketi <https://www.apachefriends.org> adresinden indirip kurulumu tamamlayınız. Kurulum bittiğinde bilgisayarınızda belirlediğiniz bir klasörde paketin tüm bileşenleri (Apache web sunucusu, MariaDB veri tabanı, PHP yorumlayıcısı) kurulmuş olacaktır. Sunucunuzu başlatmak için bilgisayarınızdaki XAMPP Kontrol Paneli uygulamasını açıp, Apache servisini çalıştırmanız yeterli olacaktır. Bundan sonra XAMPP klasörü altındaki htdocs klasörü web sunucunuzun ana dizini olacaktır. Yazacağınız PHP dosyalarını bu ana dizin içine kaydetmelisiniz. Ardında tarayıcınızda <http://localhost/phpkodlari.php> gibi ifade ile PHP dosyalarınızı çalıştırabilirsiniz. Kod yazmak için herhangi bir metin editörünü kullanabilirsiniz. Sunucuyu test etmek için aşağıdaki kodları bir metin editöründe ilksayfa.php adı ile htdocs klasörüne kaydedip, ardından tarayıcınızda <http://localhost/ilksayfa.php> yazıp Enter tuşuna basın. Tarayıcınızda “Merhaba” yazısını görmelisiniz. Merhaba PHP Sunucunuz PHP ile ilgili bir çok ayarı mevcuttur; bunları görmek için tarayıcınıza <http://localhost/dashboard/phpinfo.php> yazıp Enter tuşuna basın. Birkaç sayfa uzunluğundaki listede PHP ile ilgili sürüm notları, kurulum yolları ve birçok ayar göreceksiniz. Bu ayarları değiştirmek için XAMPP kurulumunu yaptığımız klasörde PHP alt klasörü içindeki php.ini dosyasını açıp düzenlemeler yapmalısınız. Örneğin sunucuya web tarayıcısı üzerinden yükleme yapabileceğiniz dosya boyunu limiti; olası hatalarda hata mesajlarının tarayıcıda görünüp görünmemesi ya da oturma süresini buradan değiştirebilirsiniz. Değişikliklerin geçerli olması için php.ini dosyasının kaydedilmesi ve web servisinin durdurulup tekrar çalıştırılması gerekir.

PHP DİLİNİN YAPISI

PHP diliyle yazılmış komutlar, genellikle HTML sayfalarındaki HTML kodlarının içine ifadeleri arasına gömülerek yazılır. Bu ifadeler HTML komutları içinde birden fazla sayıda açılıp kapatılabilir. Açılan bu PHP kod blokları arasına istenildiği kadar PHP komutu yazılabilir. PHP komutları sonuna noktalı virgül konulur, bu koşul ile yan yana ya da alt alta yazılabilirler. Aşağıdaki örnekteki echo komutu, yanına yazılan ifade ya da değişkenlerin değerlerini tarayıcıya yazar. `PHP"; // Merhaba PHP ?` > PHP komutları arasına açıklama veya yorum satırı eklemek için iki yol vardır: Satırın başına `//` sembollerini eklemek ya da açıklama metninin başına ve sonuna `/*` ve `*/` ifadelerini yazmak. Bu şekilde yaratılan açıklama satırları çalışma sırasında göz ardı edilir. Değişkenler PHP’de değişkenler tanımlanmadan ilk değerlerini verilerek kullanılır. Değişkenlerin isimlerinin başına `$` sembolü eklenmelidir. Sonrasında boşluksuz ve noktalama işareti olmadan (sadece alt tire sembolü kullanılabilir) isimlendirme yapılmalıdır. İlk karakter hariç rakamlar da kullanılabilir. Değişkenlerin türleri atanan değere göre otomatik değişecektir. Değişken türü olarak, string, integer, float, boolean, array ve object türleri vardır. Değişkenler tanımlandıkları yere göre yerel, genel, statik veya parametre kapsamlı değişken olabilirler. Değişken sahip olduğu kapsam özelliğine göre erişilebilir veya değerini koruyabilir farklılığına sahip olurlar. Değişkenlere ek olarak, define komutu ile sayfa içerisinde geçerli olan ve tanımlandıktan sonra değeri değiştirilemeyen sabitler tanımlanabilir. Matematik ve Atama Operatörleri Toplama, çıkarma, çarpma ve bölme gibi işlemler sırasıyla `+`, `-`, `*` ve `/` operatörleri ile yapılır. Ek olarak `%` sembolü ile mod bulma ve `.` (nokta) operatörü ile metin birleştirme işlemleri yapılır. Tek eşittir ile yapılan atama işlemine ek olarak, yukarıdaki 5 matematik operatöründe değişkenin kendisi üzerinde işlem yapılacak ise `+=`, `-=`, `*=`, `/=` ve `%=` operatörleri ile yazım kısaltılabilir. Sayısal değişkenlerin mevcut değerini bir artırmak veya bir azaltmak için `++` ve `--` operatörleri kullanılır. Aynı satırda birden fazla işlem yapılacak ise işlem önceliği çarpma, bölme ve mod alma, sonrasında toplama ve çıkarma şeklindedir. Önceliği değiştirmek için normal parantezler kullanılabilir. Diziler Aynı anda birden fazla değeri tutabilen yapılara dizi denir. PHP’de diziler ya array komutu ile ya da direk olarak değerler açık biçimde bir değişkene atanarak oluşturulur. Bu durumda ilk değer dizinin sıfırıncı elemanı olur. Eğer istenirse her bir değerlere farklı sayısal veya metin türünde anahtar atanabilir. Bu durumda dizi elemanlarına anahtar ile erişilebilir. Dizideki

elemanların her biri farklı veri türünde olabilir. Örneğin `$dizi=array (“baslik” => “PHP Nedir?”, “baslikNo” =>1);` şeklinde tanımlama yapılabilir. Dizinin her bir elemanı diğer değişkenler gibi kullanılabilir. Tüm diziyi tarayıcıya yazdırmak için `var_dump()` ya da `print_r()` komutları kullanılabilir. Bu komutlar dizideki elemanların hem anahtarlarını hem de değerlerini yazar. Dizilere ilk yaratıldıkları andan sonra yeni eleman eklenebilir veya mevcut bir eleman çıkarılabilir. Eleman eklemek için dizinin adını ve boş köşeli parantezleri yeni bir değere eşitlemek yeterlidir. Eleman çıkarmak için ise `unset` komutu kullanılmalıdır. Bunun dışında dizileri küçükten büyüğe sıralamak için `sort()`, tersten sıralamak için `rsort()` veya dizinin elemanlarını dizi içinde ters çevirmek için `array_reverse()` fonksiyonları kullanılabilir.

KONTROL YAPILARI

Programın akışını değiştiren ya da belirli komutların tekrar edilmesini sağlayan ifadelere kontrol yapıları denir. `If` Komutu Programın akışını mantıksal ifadelere göre değiştirir. Tek başına, `if-else` şeklinde veya `if-elseif` şeklinde kullanılabilir. Tek başına kullanıldığında bir mantıksal ifadenin sonucunun doğru (`true`) olmasına göre devamındaki komutların çalışmasını sağlar. İfade doğru değilse, komutlar çalışmadan akış sonraki komutlara geçer. Eğer ifade doğru olmadığında farklı komutların çalışması istenirse `if-else` şeklinde kullanılır. Eğer mantıksal ifadeler birden çok ise `if-elseif` şeklinde birden fazla mantıksal ifadelerin kontrolü yoluna gidilebilir. Mantık ve Karşılaştırma Operatörleri Bir değişkenin değerini bir başka değişken ya da açık değer ile karşılaştıran `>`, `=`, `>=`, `==`, `===`, `!=`, `!==`, operatörleri sonuca göre doğru (`true`) değerini üretir. Eğer koşul birden fazla karşılaştırma gerektiriyorsa aynı `if` içinde “ve” veya “veya” (`&&` ve `||`) operatörleri ile sonuçlar birbirine bağlanır. Ünlem sembolü (`!`) ise doğru ya da yanlış değerlerinin tersini üretir. `Switch` Komutu Bir değişkenin alabileceği değerlere sahip olup olmadığını kontrol eder, eğer herhangi birine sahip ise programın akışını o yönde değiştirir. `For` Döngüsü Tekrar edilmesi gereken komutların bir sayaç değişkeninin başlangıç ve bitiş değerleri için tekrar çalıştırılmasını sağlar. Sayaç değişkeninin ilk değeri, artış ya da azalış miktarı ile sayacın son değeri `for` satırında belirtilir. Örneği `for($a=0; $a`

PHP'DE İLERİ KONULAR

Bu ünite de fonksiyonlar, PHP'deki global değişkenler olan \$_GET, \$_POST, \$_REQUEST, \$_SERVER, \$_SESSION ve \$_COOKIE üzerinde durulmuştur. Ayrıca bu değişkenler ile birlikte sıklıkla kullanılan HTML formlarından veri alma ve kontrolü ile gerektiğinde PHP komutlarının çalışmasının durdurulması veya sayfanın başka sayfalara yönlendirilmesi konularından bahsedilmiştir.

DIŞ DOSYALARIN DAHİL EDİLMESİ

Farklı sayfalarda tekrar kullanılması gereken birçok komut ayrı bir PHP sayfası halinde kaydedilip, istenilen PHP sayfasına include veya require komutları ile dahil edilebilir. Require komutu dosyayı bulamazsa hata vererek çalışmanın durmasına neden olur. Bu komutlar ile dahil edilen dosyanın tekrar aynı sayfaya eklenmesi risk oluşturur, bu riske girmemek için include_once() veya require_once() komutları ile dahil edilmesi tavsiye edilir. Bu sayede dahil edilen komut tekrar dahil edilmez.

FONKSİYONLAR

Belirli komutların bir isim altında toplanmasıyla oluşan yapılara fonksiyon denir. Fonksiyonlar parametrelili ya da parametresiz olabilir. Parametrelili fonksiyonlar tanımlandığı başlıktaki sayıda parametre ya da değer ile çağrılmalıdır. Fonksiyon içinde yapılan işlemlerin sonucunda geriye değer döndürebilir ya da döndürmeyebilir. Geriye değer döndürülmesi istendiğinde fonksiyon içinde bu değer return kelimesi ile birlikte çağrılmalıdır. Fonksiyon tanımında istenilen parametrelere varsayılan değer atanabilir. Bu değerler çağrılma satırında yazılmaz ise varsayılan değer geçerli olur. Fonksiyon isimlendirme kuralları değişken isimlendirme kuralları ile aynıdır. Fonksiyonlara gönderilen parametrelerin sadece değeri gönderilir. Ancak tanımlama satırında parametre adının önüne & eklenirse, adres gönderilmiş olur. Bu durumda fonksiyon içindeki değer değişikliği, çağrıdaki parametrenin de değerinin değişmesine neden olur. Fonksiyonlara diziler gönderilebilir veya fonksiyon geri dönüş değeri olarak dizi geri döndürülebilir.

HTML FORM ETİKETLERİ İLE ÇALIŞMA

Formlar kullanıcıların veri girdikleri arayüz elemanlarıdır. Bir forma girilen bilgiler paket olarak sunucuya gönderildiğinde, formun metot özelliğine göre sunucudaki \$_GET ya da \$_POST global değişkeni içerisinde saklanır. Ayrıca metottan bağımsız olarak aynı bilgileri \$_REQUEST global değişkeni içinde de saklanır. HTML'deki formun içindeki etiketlerin name ve value değerleri, sunucudaki global değişkende anahtar ve değer olarak bir dizi oluşturur. Sonrasında, örneğin formun metot özelliği "get" ise, sunucuda echo \$_GET["kullaniciAdi"]; şeklinde bu değere erişilebilir, değeri değiştirilebilir. Form Verilerinin Kontrolü HTML sayfasından gelen bilgilerdeki eksiklik ya da biçim farkı, sonraki işlemlerde hatalara neden olabilir. Ayrıca art niyetli kişiler, veri olarak JavaScript veya SQL betikleri girmiş olabilir. Bu nedenle gelen verilerin üç açıdan kontrolü önerilir:

- 1) Zorunlu verilerin girilip girilmediği;
- 2) girilen verilerin biçiminin uygun olup olmadığı,
- 3) girilen verilerde riskli içerik ya da karakter olup olmadığı.

isset() parametre olarak aldığı değişkenin var olup olmadığını; empty() fonksiyonu ise parametre olarak aldığı değişkenin, türüne uygun biçimde, boş olup olmadığını kontrol eder. Bu fonksiyonlar ile ilgili global değişkenin içerisindeki verilerin öncelikle kontrol edilmesi önerilir. Biçim kontrolü için filter_var(), is_numeric(), is_string() vb. komutlar, girdinin eposta, URL, tam sayı ve diğer bazı veri türü biçimlerine uygun olup olmadığını kontrol eder. Düzenli ifadeler yoluyla biçim veya karakter dizilimi kontrolü yapmak için preg_match() fonksiyonu kullanılabilir. Girdideki riskli karakterler trim(), stripslashes() ve htmlspecialchars() fonksiyonları yok edebilir veya dönüştürebilir.

\$_SERVER DEĞİŞKENLERİ VE YÖNLENDİRME

PHP komutları içerisinde, web sunucusu ya da o sayfa hakkında detaylı bilgi almak gerektiğinde \$_SERVER global değişkeninin içindeki birçok bilgi kullanılabilir. Örneğin web sayfasının adresi, sayfanın adı, sayfanın disk üzerindeki fiziksel adresi gibi bilgiler metin olarak bu değişken içerisinden ulaşılabilir. Bir sayfaya gelen istek, farklı nedenler ile bir başka sayfaya yönlendirilmek istenirse header() komutu ile bu işlem yapılabilir. Bunun için header("Location: başkaSayfa.php"); benzeri komut yeterlidir. Yönlendirme gerekirse başka bir web sitesine de yapılabilir. Sayfa içerisinde çalışmanın durdurulması gerektiğinde exit() veya die() komutları kullanılabilir. Bu komutlardan sonra sayfadaki diğer kod ya da komutlar çalıştırılmaz.

\$_SESSION DEĞİŞKENİ VE OTURUM YÖNETİMİ

Web sunucusunu ziyaret eden her kullanıcı için, sunucu üzerinde bir oturum başlatılır. Bu oturum içine, istenirse PHP aracılığı ile bilgi eklenebilir. Bu bilgilere oturum sonlanmadığı sürece farklı sayfalardan erişilebilir. Oturum, ancak tarayıcı kapatıldığında, PHP içerisinde oturum kapatma komutu (session_destroy) verildiğinde veya önceden belirlenmiş oturum süresi sona erdiğinde yok edilir. Oturum bilgileri \$_SESSION global değişkeni içerisinde anahtar - değer biçiminde bir dizi olarak eklenebilir, değiştirilebilir ya da silinebilir. Oturumun sona erme süresi, sunucu üzerindeki farklı sayfalara gidildiğinde yeniden başlatılır. Kullanıcı tarayıcısını uzun süre kapatmaz ya da farklı bir sayfaya gitmezse süre sınıra ulaşabilir (bu süre sınırı php.ini dosyasında belirlenir) ve oturum sonlandırılır. Tüm oturum değişkenlerine ulaşmak için foreach döngüsü kullanılabilir. Oturumun saklama süresi sona ermeden yok edilmesi gerekirse session_destroy() komutu ile yok edilebilir. Bunun yerine \$_SESSION değişkenlerinden biri ya da hepsi silinmek istenirse unset() ile bir tanesi, session_unset() komutu ile tüm değişkenler silinir, ancak oturum yine hayatta kalır.

\$_COOKIE VE ÇEREZ YÖNETİMİ

Çerezler, kullanıcının tarayıcısı üzerinde web siteleri tarafından saklanan verilerdir. Kullanıcılar isterse bu verilerin saklanmasına engel olabilir. Çerezler herhangi bir sayfaya ilişkin sunucuya istek gönderilirken tarayıcı tarafından otomatik gönderilir ve PHP ile \$_COOKIE global değişkeni içinde tutulur. Bu global değişkene setcookie() komutu ile yeni çerez eklenebilir, ya da mevcut değişken değiştirilebilir. Bu ekleme veya değiştirme işlemi, aynı sayfanın tekrar yüklenmesi veya farklı bir sayfaya gidilmesi üzerine geçerli olur. Bir çerezi silmek için setcookie() fonksiyonu, çerezin saklanma süresine ilişkin parametreye geçmiş bir tarih belirterek çağırılmalıdır. Çerezler, tanımlandıkları anda belirlenen süre sonunda otomatik silinirler. Çerezlere erişim aynı zamanda istemci tarafı dil olan JavaScript ile de mümkündür. Bu sayede istemci tarafında çerez desteği sorgulanabilir

PHP ve Veri Tabanına Erişim

PHP: VERİ TABANI PHP, dinamik web sayfaları oluşturmak için kullanılabilen, sunucu tarafında, HTML'ye gömülü bir betik dilidir. Çoğu işletim sistemi ve web sunucusu için mevcuttur. PHP, açık kaynak kodlu olanlar da dâhil olmak üzere en yaygın olan 20'den fazla veri tabanını destekler. Veri Tabanına PHP ile Erişim PHP'den veri tabanlarına erişmenin iki yolu vardır. Biri, veri tabanına özgü bir eklenti kullanmak, diğeri ise veri tabanından bağımsız PDO (PHP Data Objects) kitaplığını kullanmaktır. Her iki yaklaşımın avantajları ve dezavantajları vardır. Genellikle veritabanına özgü eklentiler güçlü performans sağlar. PDO bağlantısının ise, veri tabanına özgü işlevleri bir soyutlama katmanı aracılığıyla arka planda yaptığından dolayı veri tabanı sistemleri arasında geçiş yapması çok kolaydır.

İlişkisel Veri Tabanı ve SQL

İlişkisel Veri Tabanı ve SQL İlişkisel veri tabanı, birbirleriyle ilişkili verilerin aralarında bağlantı kurularak tasarlanması ile oluşturulmuş, veri tekrarını ve hatalı veri girişini engelleyen, tablo yapılarını sadeleştiren, veri tutarlılığını sağlayan ve yüksek performans sağlayan bir veri tabanı türüdür. İlişkisel Veri Tabanı Yönetim Sistemi (Relational Database Management System, RDBMS) ise verileri sizin için yöneten bir sunucudur. Yapısal Sorgulama Dili'ni ilişkisel veri tabanlarını oluşturmak, değiştirmek ve sorgulamak için kullanabilirsiniz. SQL sözdizimi iki bölüme ayrılmıştır. Birincisi, Veri İşleme Dili (DML)'dir. DML komutlarının bazıları "SELECT, INSERT, UPDATE ve DELETE" eylem veya fiil kelimeleridir. Diğeri Veri Tanımlama Dili (DDL)'dir. DDL, "CREATE, ALTER, DROP, TRUNCATE" gibi veri tabanı yapısını etkileyen komutlardan oluşur.

PDO (PHP Data Objects)

PDO (PHP Data Objects) PDO; PHP'den veri tabanlarına erişmek için kullanılan basit, tutarlı bir ara birimdir. Veri tabanının kendine özgü eklentisi ile bağlanmak daha performanslı sonuçlar verebilir. Ancak PDO Kütüphanesi kullanılırsa farklı veri tabanları ile uyumlu çalışacağı dikkate alınması gereken bir avantajdır. Bazı özellikleri şunlardır: C Programlama dili ile yazılmıştır, ara bellekteki verileri kullanır, yaygın veri tabanı özellikleri vardır ve veri işleme (transaction) tekniğini kullanır.

PHP ve MySQL İşlemleri

PHP ve MySQL Günümüzde kullanılan ve pek çok yaygın web sitelerinin bel kemiği olan MySQL, PostgreSQL ve Oracle gibi ilişkisel veri tabanı sistemleri vardır. PHP, MySQL veri tabanına bağlanmak için kendi eklentisi olan mysqli() fonksiyonunu kullanır. PHP'nin mysqli() bağlantı fonksiyonu altı parametre alır ve başarılı olduğunda bir MySQL bağlantı tanımlayıcısı (link identifier) veya başarısızlık durumunda FALSE döndürür. PHP, bir MySQL sorgusu çalıştırmak için mysqli->query() fonksiyonunu kullanır. Bu fonksiyon iki parametre alır ve başarı durumunda TRUE veya başarısızlık durumunda FALSE döndürür. Başka bir PHP fonksiyonu olan close() kullanılarak istendiği zaman MySQL veri tabanı ile bağlantıyı kesilebilir.

MySQL Veri Tabanı ve Tablo Yönetimi

MySQL Veri Tabanı Oluşturma SQL komutlarında "CREATE DATABASE " yapısı, yeni bir veri tabanı oluşturmak için kullanılır. Bu işlem için veri tabanında yeterli yetkiye sahip olması gerekir. MySQL Veri Tabanı Silme SQL komutlarında "DROP DATABASE " yapısı, mevcut bir veri tabanını silmek için kullanılır. Silme işlemi çok dikkatli yapılmalıdır, çünkü bu işlemin geri dönüşü yoktur. Veri tabanı içindeki bütün nesnelere veri tabanı ile birlikte silinecektir. MySQL Veri Tabanı Seçimi PHP'de çalışılmak istenen veri tabanını seçmek için mysqli_select_db() fonksiyonu kullanılır. Bunun nedeni, MySQL sunucusunda birden fazla veri tabanı bulunabilmesidir. MySQL Tablosu Oluşturma SQL komutlarında "CREATE TABLE ()" yapısı, yeni bir tablo oluşturmak için kullanılır. Oluşturulacak olan tablo ismiyle aynı isimde başka bir tablonun mevcut olmamasına dikkat edilmelidir. MySQL Tablosu Silme SQL komutlarında "DROP TABLE " yapısı, mevcut bir tabloyu bulunduğunuz veri tabanından silmek için kullanılır. Silme işlemi çok dikkatli yapılmalıdır, çünkü bu işlemin geri dönüşü yoktur. Bütün veriler tablo ile birlikte silinecektir.

MySQL Veri Manipülasyonu (CRUD)

MySQL Tablosuna Veri Ekleme SQL komutlarında "INSERT INTO (, , ...) VALUES (, , ...)" yapısı, tabloya veri eklemek için kullanılır. MySQL Tablosundan Veri Alma SQL komutlarında "SELECT , , ... FROM , , ... WHERE OFFSET LIMIT " yapısı, tablo veya tablolardan veri almak için kullanılır. SELECT ile tablodan kayıt alırken OFFSET parametresini kullanarak belirteceğiniz M

sayısı kadar kaydı atlayabilirsiniz (örneğin M yerine beş yazarsanız beşinci kayıttan sonraki veriler alınır). Varsayılan olarak ofset sıfırdır. MySQL Tablosunda Veri Güncelleme SQL komutlarında “UPDATE SET = , = , = ... WHERE ” yapısı, tabloda veri güncellemek için kullanılır. MySQL Tablosundan Veri Silme SQL komutlarında “DELETE FROM WHERE ” yapısı, tablodan verileri satır bazında silmek için kullanılır. Bu SQL komutu çok dikkatli kullanılmalıdır, çünkü silinen verinin geri dönüşü yoktur.

PYTHON PROGRAMLAMA DİLİNE GİRİŞ

Python, 1989 yılında Guido van Rossum tarafından oluşturulmuş, sunucu taraflı, açık kaynak kodlu, diğer programlama dillerine göre öğrenmesi daha kolay, üst seviye bir bilgisayar programlama dilidir. Python ile program geliştirme hayal gücü ile sınırlıdır. Hemen hemen her alanda kullanılabilir. Pek çok farklı işletim sistemi üzerinde çalışabilir. İngilizce söz dizimine benzer yapıdadır. Derleme (compile) yerine Yorumlama (interpreter) yoluyla çalışır ve bu sayede çok daha hızlı program üretilebilir. Yöntemsel (procedural), nesne yönelimli (object-oriented) ve fonksiyonel (functional) özellikleri destekler..

Python Programlama Dili Tarihçesi ve Popülerliği

Python Programlama Dili Python 1.0 versiyonu 1994 yılında kullanıma sunulmuştur. Aralık 2008 yılında önceki sürümüyle uyumsuz versiyon 3.0 piyasaya sürülmüştür. Aralık 2021 itibariyle Python'ın resmi web sitesi olan python.org'da 3.10 versiyonu kullanıcıların erişimine açılmıştır. Berkeley Üniversitesinden alınan istatistiklere göre 2021 yılında en çok tercih edilen programlama dili olmuştur.

Kurulum ve IDLE Arayüzü

Kurulum En yaygın kullanılan işletim sistemleri olan Windows, Linux ve macOS için kurulum dosyaları <https://python.org> sitesinde ücretsiz ve açık kaynak koduyla hizmete sunulmaktadır. Windows ve macOS'da kurulum sonrası etkileşimli ara yüzü IDLE ile program yazmak mümkün hale gelir. IDLE ile derlemeye (compile) gerek kalmadan yorumlama (interpreter) yoluyla program kodları hızlı bir şekilde işletilebilir. Linux ve macOS'un terminal bağlantıları ile de Python etkileşimli ara yüzüne erişim mümkündür. Bazı işletim sistemleri üzerinde Python kurulu olarak gelir, bazılarında ise sonradan kurmak gerekir. Her iki durumda da son sürüm kullanmak en faydalı olanıdır.

İlk Python Programınız ve Temel İşlemler

İlk Python Programınız En basit haliyle bir Python program fonksiyonu olan print() fonksiyonu içine çift tırnak veya tek tırnak arasında olmak kaydıyla istediğiniz metni yazarak ilk programınızı oluşturabilirsiniz. Örneğin, print("Merhaba Dünya!") komutu ekrana Merhaba Dünya! kelimelerini basacaktır. Temel Bilgiler IDLE etkileşimli ara yüzündeki komut satırı gelişmiş bir hesap makinesi olarak kullanılabilir. Python'da matematik işlemleri yapılırken bilinen matematik kuralları geçerlidir. Örneğin aynı anda yapılan bölme, çıkarma, toplama ve çarpma işlemlerindeki öncelik sırası komut satırında da geçerlidir. Python programlamada metin, sayısal, dizi, sözlük, küme, boolean ve ikili veri türleri mevcuttur.

Söz Dizimi ve Girinti Yapısı

Söz Dizimi Python kodu çalıştırmanın etkileşimli ara yüz (komut satırı) ve betimleme ara yüzü olmak üzere iki metodu vardır. Diğer programlama dillerinin aksine "girinti" Python programlama dilinde önemli bir söz dizimidir. Kod bloğu girintilerle belli olur ve en az bir karakter girinti boşluğu verilmelidir.

Değişkenler ve Bellek Yönetimi

Değişkenler Değişkenler, değerleri depolamak için ayrılmış bellek konumudur. Değişken oluşturmak için eşittir (=) işareti kullanılır. Yorumlayıcı, değişkenin veri tipine bağlı olarak hafızada yer ayırır ve burada nelerin saklanabileceğine karar verir. Python değişkenleri, bellek alanı ayırmak için herhangi bir bildirim ihtiyacı duymaz. Bir değişkene değer atadığınızda bildirim otomatik olarak gerçekleşir. Değişkenlere değer atamak için eşittir işareti (=) kullanılır. İşaretin solundaki değişkenin adı olur. Sağındaki ise değişkende depolanan değerdir.

Veri Türleri

Veri Türleri Python 5 standart veri türüne sahiptir. Bunlar: sayı (number), metin(string), liste (list), demet (tuple) ve sözlük (dictionary)'tür. Sayı (number) veri türü, sayısal değerleri depolar. Python, int, float ve complex olmak üzere 3 farklı sayı türünü destekler. Python'daki metinler, tırnak işaretleri içinde gösterilen bitişik bir dizi karakter olarak tanımlanır. Python, tek veya çift tırnak çiftine izin verir. Liste (list), Python'daki farklı veri türlerinin bir arada kullanılabildiği en kullanışlı olan dizi veri türüdür. Bir liste, virgülle ayrılmış ve köşeli parantez ([]) içine alınmış öğeler içerir. Demet (tuple), listeye benzer başka bir dizi veri türüdür. Demetler, parantez (()) içine alınır ve güncellenemezler. Demetler salt okunur listeler olarak düşünülebilir. Python'ın sözlükleri bir tür karma-tablo (hash-table) türüdür.

Operatörler, Karar Verme ve Döngüler

Operatörler Operatörler işlenenlerin değerini değiştiren yapılardır. Python programlama dilinde; aritmetik, karşılaştırma, atama, mantık, bit, üyelik ve kimlik başlıca operatörlerdir. En çok kullanılanları aritmetik, karşılaştırma, atama ve mantık operatörleridir. Karar verme Karar verme, bir programın yürütülmesi sırasında ortaya çıkan koşulların ve o koşullara göre yapılan belirli eylemlerin öngörülmesidir. Karar verme yapıları, sonuç olarak DOĞRU veya YANLIŞ üreten birden çok ifadeyi değerlendirir. Sonuç DOĞRU veya YANLIŞ ise hangi işlemin gerçekleştirileceğine ve hangi ifadelerin yürütüleceğine önceden karar vermek gerekir. Döngüler Programlamada, bir fonksiyondaki kod bloğunun tekrar gerektiren ifadelerinin kolay bir şekilde oluşturulmasında kullanılan yapılardır.

PYTHON İLE WEB PROGRAMLAMA

Yaygın kullanılan internet tarayıcıları Python Programlama dilini direk çalıştıramaz. Python sunucu tarafı çalışan bir programlama dilidir ve Python ile bir web uygulaması oluşturmak için, bu dilin yanı sıra HTML (Hypertext Markup Language, Hiper Metin İşaretleme Dili), CSS (Cascading Style Sheets, Basamaklı Stil Şablonları) ve JavaScript gibi bütün tarayıcıların desteklediği ve istemci tarafında çalıştırılabilir olan dilleri de bilmek gerekir. Tarayıcılar tarafından direk desteklen tek programlanabilir dilin JavaScript olduğu söylenebilir. Bu nedenle Python geliştiricileri kendi web uygulamalarını oluşturmak için Python ve JavaScript'in birlikte çalıştığı hazır sistemleri kullanırlar. Bunlardan en yaygın olanları; Flask, Django, Web2py, Tornado ve Bottle sayılabilir. **PYTHON VE WEB**

Web üzerinde çalışan bir Python uygulaması geliştirmek için güncel işletim sistemi olan ve Python programlama dilinin yüklü olduğu bir bilgisayar, metin editörü, web servisi ve web çerçevesine ihtiyaç vardır. Profesyonel anlamda kullanımda olan bir web çerçevesi (bazı Türkçe kaynaklarda çerçeve yerine çatı kelimesi de kullanılmaktadır) seçilmesi, üretilecek programa performans ve süreklilik açısından fayda sağlayacaktır.

PYTHON VE IIS

Windows İşletim sisteminin bir uygulaması olan Internet Information Services'de Python uygulamasının çalışabilmesi için bazı ayarların yapılması gerekir. py dosya uzantısının tanıtılması, CGI desteğinin yüklenmesi, kök izin ve sanal uygulama oluşturulması bunlardan bazılarıdır. CGI desteğini eklemek için Windows komut satırına yönetici olarak girip, %windir%\system32\OptionalFeatures.exe komutu çalıştırılarak özellik ekleme penceresine erişim sağlanabilir. Aynı şekilde %windir%\system32\inetrsv\iis.msc kısa yol komutu ile IIS yönetim paneli açılıp web sunucusu ayarları yapılabilir.

PYTHON İLE İLK UYGULAMA

Python programlama dilinde print komutu tırnak içinde yazılan metnin ekrana basılmasını sağlar. merhaba_dunya.py adında bir dosya oluşturup bu dosyanın içine print("Merhaba Dünya!") yazıp bunu internet tarayıcımızda görüntülediğimizde Python ile ilk uygulamamız yazılmış olacaktır.

PYTHON VE FLASK

Python programlama dili ile web üzerinde uygulama yapmak için genellikle çerçeve yapı kullanılır. Çerçeve yapı; herhangi bir programlama dilinde üzerine program yazabileceğiniz hazır yapılar olarak tanımlanabilir. Bu yapı sayesinde oluşturulacak olan programa en baştan başlamak yerine, hazır olan bir seviyeden başlamak çok daha kolay olacaktır. Program oluştururken çerçeve yapı kullanmak zaman kazandırır. Her şeyi sıfırdan yazmak gerekmez, bu nedenle hata yapma riski azalır. Bunun yanı sıra bu yapı daha önceden uzman yazılımcılar tarafında test edildiği için güvenle kullanılabilir. Flask Nedir? Flask, Python programlama dili ile yazılmış ücretsiz ve açık kaynak kodlu bir web uygulama mikro çerçeve yapılı yazılımdır. Mikro çerçeve olarak tanımlanmasının sebebi özel araçlara veya program kütüphanelerine ihtiyaç duymamasındandır. Flask, 2004 yılında kurulan uluslararası bir Python meraklıları grubu olan Pocomo'dan Armin Ronacher tarafından oluşturulmuştur. Flask, Ronacher'ın 2010 yılındaki 1 Nisan şakası olarak başlamış ve günümüzde birçok uygulamada kullanılan bir çerçeve yapısına dönüşmüştür. Flask Kurulumu Flask, iki farklı şekilde kurulabilir. Birincisi açık kaynak kodu indirip el ile kurulum yapmaktır. İkincisi pip komutu kullanarak kurulum yapmaktır. Web sunucusu ile Python arasında, WSGI ve FastCGI kullanarak köprü vazifesi gören wfastcgi eklentisinin de kurulması gerekir. escape() Dekoratörü escape() dekoratörü, internet üzerinden olası bir saldırıdan korunmak diğer bir deyişle kaçmak için kullanılır. Bir HTML döndüğünde kullanıcının gönderdiği değerlerin kontrolü otomatik olarak Jinja tarafından yapılır. Eğer kullanıcı tarafından gönderilen veri, escape() süzgecinden geçtikten sonra sonucunu döndürürse, program çalıştırılmaz onun yerine gelen değer ekrana metin olarak basılır. route() Dekoratörü Python, bu sebeple bir işlevi belirli bir adrese yönlendirmek için route() dekoratörünü kullanır. Kullanıcılar hatırlayabilecekleri bir adresi genellikle daha sık ziyaret ederler. Python, bu sebeple bir işlevi belirli bir adrese yönlendirmek için route() dekoratörünü kullanır. Değişken Dönüştürme Bir web adresine değişken isim atanarak o adresin dinamik hale gelmesi sağlanabilir. Bunu yapmak için değişken isimler küçüktür () karakterleri arasına yazılır. Değişken, istenilen veri türüne çevrilebilir.

Python programlama dilindeki dönüştürülebilir veri türleri ve açıklamaları şunlardır:

- string: Bölüm (/) işareti olmayan her metni kabul eder
- int: Pozitif tam sayıları kabul eder.
- float: Pozitif ondalık sayıları kabul eder
- path: String ile aynı fakat Bölüm (/) karakterini de kabul eder.
- uuid: UUID string türünü kabul eder. url_for() Dekoratörü url_for() dekoratörü, özel fonksiyonlar için dinamik URL oluştururken kullanılır. Benzersiz URL'ler Werkzeug kütüphanesi, kapsamlı bir WSGI web uygulama kütüphanesidir ve Python programlama dilinde oluşturulan URL'lerin benzersiz olmasını sağlar. Örneğin; adres olarak “/python” veya “/python/” kullanmak aynıymış gibi gözükür ancak “/python” klasörü için tanımlanan bir route fonksiyonu “/python/” adresinden çağrılırsa 404 – Bulunamadı hatası ile sonuçlanır. HTTP İletişim Metotları HTTP protokolü World Wide Web’de veri iletişimi için kullanılır. Açılımı Hypertext Transfer Protokol olan HTTP, bu iletişimi sağlarken farklı metotlar kullanır. Bu metotlar GET, HEAD, POST, PUT ve DELETE metotlarıdır. GET metodu, verileri şifrelenmemiş bir şekilde sunucuya iletir. En yaygın kullanılan metottur. POST metodu, sunucuya HTML form verisi göndermek için kullanılır. POST yöntemiyle alınan veriler sunucu tarafından önbelleğe alınmaz. Varsayılan olarak FLASK yönlendirmesi GET isteklerine yanıt verir. Ancak route() dekoratörüne parametre verilerek değiştirilebilir. Şablonlar (Templates) Belirli bir URL'ye bağlı bir fonksiyonun çıktısını, HTML biçimine döndürmek mümkündür. Python kodundan HTML içeriği oluşturmak; özellikle değişken veriler ve koşul veya döngüler gibi Python dili öğelerinin konulması gerektiğinde çok zahmetlidir. Bu, HTML'den sık sık kaçmayı gerektirir. Böyle durumlarda Flask’ın temel olarak aldığı Jinja2 şablon motoru kullanılır. Fonksiyon içerisine sabit HTML kodu yazmak yerine, render_template() fonksiyonu ile dışarıda oluşturulmuş bir HTML dosyası kullanılabilir.

Web Servisleri ve Genel Tanım

Bir web servisleri, internet üzerinden kullanıma sunulan standartlaştırılmış, XML mesajlaşma sistemi kullanan bir yazılım parçasıdır. Web servisler farklı yazılımların birbirleriyle veri alışverişini sağlamak için kullanılan XML tabanlı teknolojilerdir. Web servis genellikle farklı platformlar arasında belirlenen ortak bir biçim ile veri alışverişi yapmak için kullanılır. Web servisler, HTTP kullanarak hizmet sağlayan yapıların bütününe verilen isimdir. Web servisler, istemci tarafından gönderilen istekleri yorumlayarak gönderilen isteğe göre istemciye yapısından bulundurduğu bilgileri aktarmaktadır. Web servisler, istemci tarafından gönderilen istekleri yorumlayarak gönderilen isteğe göre istemciye taşıdığı bilgileri aktarır.

Web Servis Mimarisinin Temel Birimleri

Web servisleri modeli üç ana birimin etkileşimine dayanır. Bu birimler Servis Sağlayıcı (Service Provider), Servis İstemcisi (Service Requester) ve Servis Kayıt Birimi (Service Registry).

Platform Bağımsızlık ve İletişim Standartları

Web servis genellikle farklı platformlar arasında belirlenen ortak bir biçim ile veri alışverişi yapmak için kullanılır. Web üzerinden HTTP ile hizmet veren program parçaları olarak da tanımlanabilir. Bir kullanıcının HTTP üzerinden bir web servisi kullanması işlemine Remote Procedure Call (RPC) denir. HTTP üzerinden yapılan bu çağrılara karşı SOAP, XML çıktıları üretir. Veri transferi XML çıktıları ile yapıldığı için platform bağımsız bir yapısı vardır. Web servisi kullanmak istenilen program parçasının hangi dilde yazıldığı önemli değildir. Çünkü sonuçta Web servisten alınan veri tipi XML'dir. XML, tüm iletişimleri bir web hizmetine kodlamak için kullanılır. Bundan dolayı C# ile yazılmış bir Web Servis Java tarafından Java ile yazılan bir Web Servis .Net teknolojileri tarafından kolayca kullanılabilir. Bu yazılımların çalıştıkları platformlarında pek önemi yoktur Linux işletim sistemi üzerinde çalışan bir uygulamayla Windows işletim sistemi üzerinde çalışan bir uygulama aynı Web Servisi sorunsuzca kullanabilirler. Örneğin, bir istemci bir XML mesajı göndererek bir web hizmetini çağırır ve ardından karşılık gelen bir XML yanıtını bekler. Tüm iletişim XML ile gerçekleştiği, web hizmetleri herhangi bir işletim sistemine veya programlama diline bağlı değildir. Java, Perl ile konuşabilir; Windows uygulamaları Unix uygulamaları ile konuşabilir. Web servisler, ürünler, süreçler ve tedarik zincirleri oluşturmak için ağ üzerinden tanımlanabilen, yayınlanabilen, konumlandırılabilen veya çağrılabilen bağımsız, modüler, dağıtılmış, dinamik uygulamalardır. Bu uygulamalar yerel, dağıtılmış veya web tabanlı olabilir. Web hizmetleri, TCP/IP, HTTP, Java, HTML ve XML gibi açık standartların üzerine inşa edilmiştir. Örneğin; Çok kullanıcı olan bir firmada Java programlama diliyle geliştirilen muhasebe bilgi sistemi veya varlık yönetim uygulaması, kullanıcıların isteği üzerine Android işletim sistemli cep telefonlarında da kullanılmak istenebilir. Platformların birbirleriyle iletişim kurabilmesi için ortak bir biçimi kullanması gerekir. Bu durumda platformlar arası ortak bir biçimi kullanan web servis yazılarak iletişim sağlanır.

SOAP (Simple Object Access Protocol)

Günümüzde farklı web servis yöntemleri kullanılmaktadır. Bu yöntemlerin yapıları farklı olsa da yine XML dilini kullanırlar. SOAP: Uygulama bütünleştirme için geliştirilmiş XML ile formatlanmış bilgilerin iletişimini sağlayan bir protokoldür. Uygulama, bütünleştirme için middleware olarak adlandırılan daha önce birçok çözümler sunmuştur. RPC (Remote Procedure Call) , DCOM, IIOP (Internet-Inter-ORB Protocol) ve Java RMI bu çözümlerden bazılarıdır. Bu middleware çözümleri internet ortamında iletişim sağlama konusunda yetersiz kalmaktadır. Bu nedenle XML tabanlı bir protokol olan SOAP giderek yaygınlaşmaktadır. SOAP istemcilerin sunucularda olan nesne yöntemlerini çağırmasını ve sonuçların alınmasını sağlayan basit istek/yanıt (request/response) protokolüdür. SOAP mevcut internet altyapısında olan router, firewall ve Proxy sunucularında herhangi bir değişiklik yapılmadan kolayca çalışabilmektedir. SOAP mesaj yapısı XML yazım kurallarına uygun olarak şu bölümlerden oluşur: Envelope, Header, Body ve Fault alanlarını barındırır.

REST (Representational State Transfer)

REST: Sunucu ile istemci arasında veri alışverişini sağlayan en basit web servis çeşidi diyebiliriz. Veri alışverişinde HTTP kullanır. İstemci tarafında servisi çağırarak için herhangi bir objeye ihtiyaç yoktur. CSV, XML, JSON ve metin mesajlaşma yapabilmektedir. Bu konuda SOAP'tan çok daha esnekler. Facebook ve Twitter tarafından geliştiricilere açık olan REST web servisleri bulunmaktadır. İşlemler resource ile yapılır. URI yazısında resource'a dair bilgiler iletmiştir. REST servisler URI

ile ilgili metoda HTTP üzerinden istekte bulunur. GET, POST, PUT, DELETE gibi HTTP metotları ile işlemler gerçekleştirilebilmektedir. SOAP gibi bir WDSL gerektirmez. REST servisler SOAP'a göre daha esnek ve hızlıdır.

UDDI (Universal Description, Discovery, and Integration)

UDDI: Ticari internet hizmetleri için XML tabanlı bir kayıt hizmetidir. Bir hizmet sağlayıcı, hizmeti UDDI gibi bir Web Services Registry açıkça kaydedebilir veya Web Services Inspection Language (WSIL) belgeleri gibi keşfi kolaylaştırmayı amaçlayan ek belgeler yayımlayabilir. Hizmet kullanıcıları veya tüketiciler, web hizmetlerini manuel veya otomatik olarak kullanabilirler. UDDI sunucularının ve WSIL motorlarının uygulanması, Web hizmetlerinin bulunmasına yardımcı olmak için basit arama API'leri veya web tabanlı GUI (Graphical User Interface) sağlamalıdır. Web hizmetleri entegrasyonu için bir çerçeve sunmaktadır. UDDI spesifikasyonu, XML, HTTP ve Domain Name System (DNS) gibi World Wide Web Consortium (W3C) ve Internet Engineering Task Force (IETF) standartlarını kullanmaktadır. Ayrıca platformlar arası programlama için önerilen SOAP mesajlaşma yönergelerinin ilk sürümlerini kullanmaktadır.

Web Uygulamaları ve Güvenli Yazılım Geliştirme Süreçleri

Web uygulamaları günümüzde en çok kullanılan yazılımlardır. Web uygulamaları erişimi kolay olduğu için siber saldırıların doğrudan hedefi haline gelmektedir. Dolayısıyla web tabanlı uygulamaların güvenliğini sağlamak için güvenlik yazılım geliştirme veya güvenli web programlama ilkelerine uyulması gereklidir. Ayrıca güvenli yazılım geliştirme, yazılım güvenlik risklerini belirlemek ve yönetmekle ilgilidir. Uygun gereksinim mühendisliği, tasarım ve mimari uygulamaları varsayılacak olursa, bir yazılımın risklerini tanımlamanın ve yönetmenin bir sonraki en etkili yolu, geliştirilen kodun tekrarlı olarak analiz edilmesidir. Yazılım güvenliğini sağlamak için geliştirilme aşamasında güvenli yazılım geliştirme yöntemleri kullanılarak yazılımların geliştirilmesinin sağlanması ve yazılım güvenlik zafiyetlerinin en aza indirilmesi gereklidir. Yazılım güvenliği yazılım geliştirme yaşam döngüsünün veya süreçlerinin her aşamasında göz önünde bulundurulmalıdır. Böyle geliştirilen yazılımları güvenlik riskleri en aza indirilmiş olmaktadır. Keşfedilen bir yazılım güvenlik zafiyeti, sürekli tekrarlanmaya ve görünürde sonu olmayan ciddi zararlara sebep olmaktadır.

Bilgi Güvenliğinin Temel Özellikleri (CIA Triad)

Yazılım güvenliği ile ilgili birçok güvenlik terminolojisi vardır. Bunlardan bazıları, tehdit, saldırı, güvenlik açığı, kimlik doğrulama ve yetkilendirmedir. Yazılım güvenliğinin amacı yazılım ile yönetilen ve işlenen verinin güvenliğini sağlamaktır. Güvenli yazılım geliştirme yöntemleri kullanılarak yazılımlarda işlenen bilginin güvenliğini sağlamak için gizlilik, bütünlük ve kullanılabilirlik olmak üzere üç temel bilgi güvenliği özelliğinin sağlanması amaçlanmaktadır. Bilginin güvenliğini sağlamak için bu üç temel özelliğin sağlanması gereklidir. Bunlar herhangi bir bilgi sisteminin güvenliğinin sağlanması için temel beklentilerdir. Yüksek güvenlik, verilerin ifşa edilmeyeceği anlamına gelir. Verilere ve yazılım kodlarına uygun olmayan (gizlilik) bir şekilde değiştirilmeyecek veya yok edilemeyecek (bütünlük) ve her zaman uygun erişim sağlanmasıdır (kullanılabilirlik) olarak tanımlanır.

Güvenli Kodlama ve Zafiyet Analizi

Güvenli yazılım geliştirme, yazılımların geliştirme sürecinde, güvenli yazılım geliştirme ilkelerine dikkat edilmeden geliştirilen yazılımlarda meydana gelecek, güvenlik açıklarının kötü amaçlı kullanımları önlemek ve karşı koruma sağlayacak şekilde bilgisayar yazılımlarının geliştirilmesidir. Yazılım kusurları, hatalar ve mantık kusurları, sürekli ve yaygın olarak kullanılan yazılım güvenlik açıklarının birincil nedenidir. Güvenli yazılım geliştirme, yazılımları siber saldırılara veya güvenlik açıklarına karşı korunmak için yazılımın nasıl kodlanacağına güvenlik hususları uygulayan bir dizi kurallardır. Zafiyetler, hatalar ve mantık kusurları, yaygın olarak kullanılan yazılım güvenlik açıklarının başlıca nedenlerindedir. Güvenli kodlama standartları, güvenlik açıklarını kodda bırakma riskini azaltan veya ortadan kaldıran güvenlik önlemleri sunar. Hem kullanıcılar hem de sistemler için bir proje ve gereksinimleri tanımlandıktan sonra, en iyi uygulamalara ilişkin hususlar ve güvenli kod planları bu gereksinimleri göz önünde bulundurarak belirlenir ve uygulanır. Güvenlik açığının analizi sayesinde, güvenlik açıklarının çoğunun nispeten az sayıda yaygın yazılım geliştirme hatalarından kaynaklandığını ortaya koymuştur. Bu yazılım güvenlik zafiyetlerine sebep olacak hatalara yol açan güvenli olmayan kodlama uygulamalarını belirleyerek ve yazılım geliştiricileri güvenli alternatifler konusunda eğiterek, yazılımın kullanımdan önce yazılımda bulunan güvenlik açıklarını önemli ölçüde azaltmaya veya ortadan kaldırmaya yardımcı olacak proaktif önlemler alınabilir.

Temel Güvenlik İlkeleri ve Tanımları

Güvenli yazılım geliştirmek için, girdi doğrulama, kimlik doğrulama, parola yönetimi, şifreleme uygulamaları, hata işleme ve günlüğe (log) kaydetme, veri koruma, iletişim güvenliği ve geliştirme diline göre güvenli bir kodlama standardı benimsenmelidir. Bunlardan bazıları aşağıdaki gibi tanımlanmıştır. Girdi Doğrulama: Yazılımların ara yüzlerinden ve form alanlarından girilen verilerin tip ve içerik olarak uygunluk doğrulamalarının yapıldığı işlemlerdir. Doğrulanmış tüm veriler ve girdiler siteler arası betik yazma (XSS) veya SQL enjeksiyonu saldırılarının yürütülmesini önlemek için HTML sayfasına döndürüldüğünde kodlanmalıdır. Örneğin, () kodlama türü, kullanıcı tarafından doğrulanmış verilerin eklendiği sayfanın mimari yapısına özgüdür. Kimlik Doğrulama: Yazılımlara üye olan veya üyelik girişi yapan kullanıcıların gerçek kişi olup olmadıklarının doğrulama işlemlerinin yapılması süreçleridir. Kimlik doğrulama işlemlerinin temel amacı robot yazılımlar kullanılarak yapılacak kullanıcı kayıtlarını ve girişlerini engellemektir. Yetkilendirme: Kullanıcıların yazılım

üzerinde hangi işlemleri gerçekleştirebileceklerinin tanımlanması sürecidir. Normal kullanıcılar ile yöneticilerin erişebilecekleri veya yönetebilecekleri kaynakların sınırlandırılması işlemleridir. Yetkilendirme ile ilgili zafiyetler, web uygulamaları için önemli bir bilgi güvenliği zafiyetidir. Yetkilendirme işlemlerinin yanlış kullanılmasından kaynaklanan zafiyetlerin etkisi, yetkileri ele geçirilen bir kullanıcının kaynak kullanımını hem biçim hem de önem açısından oldukça değiştendirir. Parola Yönetimi: Kullanıcı parolalarının güvenlik seviyelerinin belirlenmesidir. Kolay tahmin edilebilir parola kullanımına izin vermeyecek bir yazılım güvenlik mekanizmasının oluşturulması işlemleridir. Günlük (Log) Verilerinin Kaydedilmesi: Yazılımlarda gerçekleştirilen işlem hareket verilerinin kaydedilmesidir. Geriye dönük olarak takip edilecek süreçlerde LOG verilerine başvuru olarak özellikle adli süreçlerin aydınlatılması gerçekleştirilir.

Web Uygulamalarına Yönelik Saldırı Türleri

Güvenli web programlama kullanılmadan geliştirilen yazılımlardaki güvenlik açıklıkları veya zafiyetler saldırganların web uygulamalarını manipüle etmelerine sebep olmaktadır. Bu tür zafiyetler, istenmeyen kötücül komutların yürütülmesine izin veren, kullanıcı girdileri ile meydana gelmektedir. Tipik olarak, bu tür karışıklıklar, meta karakterlerin (noktalama işaretleri) uygun olmayan şekilde kullanılmasına veya uygunsuz alıntılardan kaynaklanmaktadır. Web uygulamalarına en çok yapılan saldırı türleri arasında SQL enjeksiyon, XSS enjeksiyonu, XML enjeksiyonu, kaba kuvvet saldırıları gibi HTTP üzerinde kullanıcı istekleri yoluyla gerçekleştirilen saldırı türleri bulunmaktadır. SQL enjeksiyonu veri tabanı uygulamalarında, yazılımlarda bulunan SQL cümlelerinin manipüle edilmesi ile gerçekleştirilmektedir. Yazılımın veri tabanındaki verilerin ele geçirilmesi, silinmesi veya değiştirilmesi gibi işlemlerin gerçekleştirilmesine sebep olur. XSS enjeksiyonu ise özellikle javascript betik dili ile kullanıcıların çerez bilgilerinin çalışması ve yazılımlara yetkisiz erişimlerin gerçekleştirilmesine sebep olur. Kaba kuvvet saldırıları ise yazılımların veri giriş formlarını veya kullanıcı adı ve şifre girişi için kullanılan web formlar üzerinden yapılır. Web form alanlarına kullanıcı adı ve şifre bilgilerini tahmin etmek için sürekli deneme yapılır ve kullanıcı adı ve şifre bilgileri tahmin edilinceye kadar tekrar edilir. Eğer şifreler kolay tahmin edilebilir olursa, kaba kuvvet saldırılarının da başarı oranı artar. Günümüzde kaba kuvvet saldırıları genelde robot yazılımlar kullanılarak gerçekleştirilir. Kaba kuvvet saldırılarını engellemek için genel CAPTCHA doğrulama özellikleri kullanılır.