

**T.C.  
MİLLÎ EĞİTİM BAKANLIĞI**

# **BİLİŞİM TEKNOLOJİLERİ**

## **GELİŞMİŞ İNTERNET UYGULAMALARINA GİRİŞ**

**Ankara, 2013**

- Bu modül, mesleki ve teknik eğitim okul/kurumlarında uygulanan Çerçeve Öğretim Programlarında yer alan yeterlikleri kazandırmaya yönelik olarak öğrencilere rehberlik etmek amacıyla hazırlanmış bireysel öğrenme materyalidir.
- Millî Eğitim Bakanlığınca ücretsiz olarak verilmiştir.
- **PARA İLE SATILMAZ.**

# İÇİNDEKİLER

AÇIKLAMALAR .....	ii
GİRİŞ .....	1
ÖĞRENME FAALİYETİ – 1 .....	3
1. GİRİŞ .....	3
1.1. Uygulama Mimarisi .....	4
1.1.1. Kullanım Alanları .....	11
1.1.2. Uygulama Geliştirme ve .NET Platformu İlişkisi .....	12
1.1.3. Programlama ve Tasarım İlişkisi .....	14
1.1.4. XAML .....	17
UYGULAMA FAALİYETİ .....	19
ÖLÇME VE DEĞERLENDİRME .....	21
ÖĞRENME FAALİYETİ – 2 .....	22
2. BAŞLANGIÇ UYGULAMALARI .....	22
2.1. Örnek Uygulama Dosyaları Oluşturma.....	22
2.1.1. Betik Dosyası Oluşturma.....	22
2.1.2. HTML Sayfası Oluşturma .....	42
UYGULAMA FAALİYETİ .....	50
ÖLÇME VE DEĞERLENDİRME .....	53
MODÜL DEĞERLENDİRME .....	54
CEVAP ANAHTARLARI.....	56
KAYNAKÇA .....	57

# AÇIKLAMALAR

<b>ALAN</b>	<b>Bilişim Teknolojileri</b>
<b>DAL / MESLEK</b>	<b>Veritabanı Programcılığı</b>
<b>MODÜLÜN ADI</b>	<b>Gelişmiş İnternet Uygulamalarına Giriş</b>
<b>MODÜLÜN TANIMI</b>	Gelişmiş İnternet uygulamaları geliştirmek için gerekli olan editörün kurulumu ve temel dosyaların oluşturulmasının kazandırıldığı bir öğrenme
<b>SÜRE</b>	40/32
<b>ÖN KOŞUL</b>	
<b>YETERLİK</b>	Gelişmiş İnternet uygulamalarıyla başlangıç işlemlerini yapmak
<b>MODÜLÜN AMACI</b>	<b>Genel Amaç</b> Bu modül ile gerekli ortam sağlandığında; gelişmiş İnternet uygulamaları için temel uygulama dosyaları oluşturabileceksiniz. <b>Amaçlar</b> <b>1.</b> Arayüzü ve editörü kullanabileceksiniz. <b>2.</b> Temel uygulama dosyaları oluşturabileceksiniz.
<b>EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI</b>	<b>Ortam:</b> Bilişim teknolojileri sınıfı <b>Donanım:</b> Bilgisayar, İnternet bağlantısı
<b>ÖLÇME VE DEĞERLENDİRME</b>	Modül içinde yer alan her öğrenme faaliyetinden sonra verilen ölçme araçları ile kendinizi değerlendireceksiniz. Öğretmen modül sonunda ölçme aracı (çoktan seçmeli test, doğru-yanlış testi, boşluk doldurma, eşleştirme vb.) kullanarak modül uygulamaları ile kazandığınız bilgi ve becerileri ölçerek sizi değerlendirecektir.

# GİRİŞ

## Sevgili Öğrenci,

İnternet günlük hayatın ayrılmaz bir parçası hâline gelmiştir. Bununla birlikte *İnternet* sitelerinde yapılan işlemler ve sitelerin kullanıcı etkileşimleri de her geçen gün artmıştır. Sosyal paylaşım siteleri her geçen gün hayatımızda daha da fazla yer edinmektedir.

Günümüzde web sitesi kavramı yerine gelişmiş İnternet uygulamaları (RIA-Rich İnternet Application) kavramı geçerli olmaktadır. Artık amaç masaüstü uygulamaları kadar hızlı web siteleri geliştirmektir. Ses, video vb. kullanıcı etkileşiminin arttığı, *İnternet* hızının sınırlarının zorlandığı bu uygulamalar her geçen gün artmaktadır.

Bütün bunlara ek olarak mobil teknolojiler, akıllı telefonlar artık hayatımızın vazgeçilmez birer parçası olmuştur. Geliştirilen uygulamaların önemli bir bölümü değişik platformlarda çalışan akıllı telefonlar için geliştirilmektedir.

Bu modülde sizlere, ihtiyaca uygun editörleri kurma ve uygulamaların temel kavramları hakkında bilgiler verilecektir.



# ÖĞRENME FAALİYETİ-1

## AMAÇ

Gelişmiş *İnternet* uygulamaları geliştirmek için gerekli olan temel bilgileri edinip gerekli araçları kullanmayı öğrenebileceksiniz.

## ARAŞTIRMA

- RIA (Rich Internet Application) kavramını araştırınız.

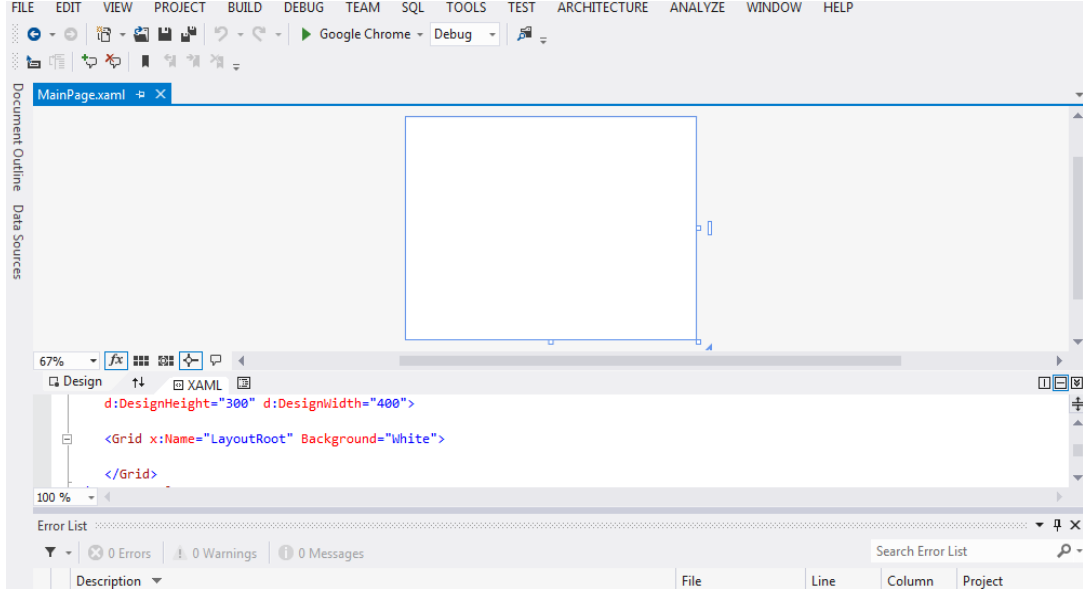
## 1. GİRİŞ

XAML tabanlı uygulama geliştirme platformu, ilk olarak .NET Framework 3.0 içerisinde entegre olarak gelen vektörel arayüzler ile yazılım geliştirilmesine olanak sağlayan Windows Presentation Foundation'ın (WPF) *İnternete* yansımasıdır.

WPF ise uygulamaların çalıştığı pencerelerin şeffaflığı, renklerin daha belirgin ve etkin kullanımı gibi özellikleri ile karşımıza çıkan yeni nesil Form uygulamalarına ve vektörel arayüzlere verilen isimdir.

WPF'in hedef makinelerde çalışabilmesi için ise .NET Framework 3.0 (veya daha üstü bir versiyon) yüklü olması şartı aranmaktadır.

Teknik olarak baktığımızda başlangıçta XAML tabanlı uygulama geliştirme platformu, JavaScript ile kontrol edilebilen bir yapı olarak görünse de daha sonraki sürümlerle beraber C# ve Basic programlama dilleri ile geliştirilme imkânı sundu. Bu yenilik ile .NET üzerinde çalışan geliştiricilerin XAML tabanlı uygulama geliştirme platformu teknolojisine uyumu çok kolay hale getirilmiştir. Geliştirme mantığı olarak WPF'e oldukça benzemektedir. ASP.NET uygulamaların yaptığımız klasik işlemleri (metod yazma, olay yakalama vb. işlemler) XAML tabanlı uygulama geliştirme platformu ile de kolay bir şekilde gerçekleştirebilmekte, aynı zamanda online video yayınlarında kullanılan alanı azaltarak daha kaliteli videoları daha etkin bir biçimde sunabilmektir.



Resim 1. 1: XAML tabanlı uygulama geliştirme platformu çalışma ekranı

## 1.1. Uygulama Mimarisi

Silverlight ile uygulamalarımızı geliştirirken Visual Studio yazılımını kullanacağız.

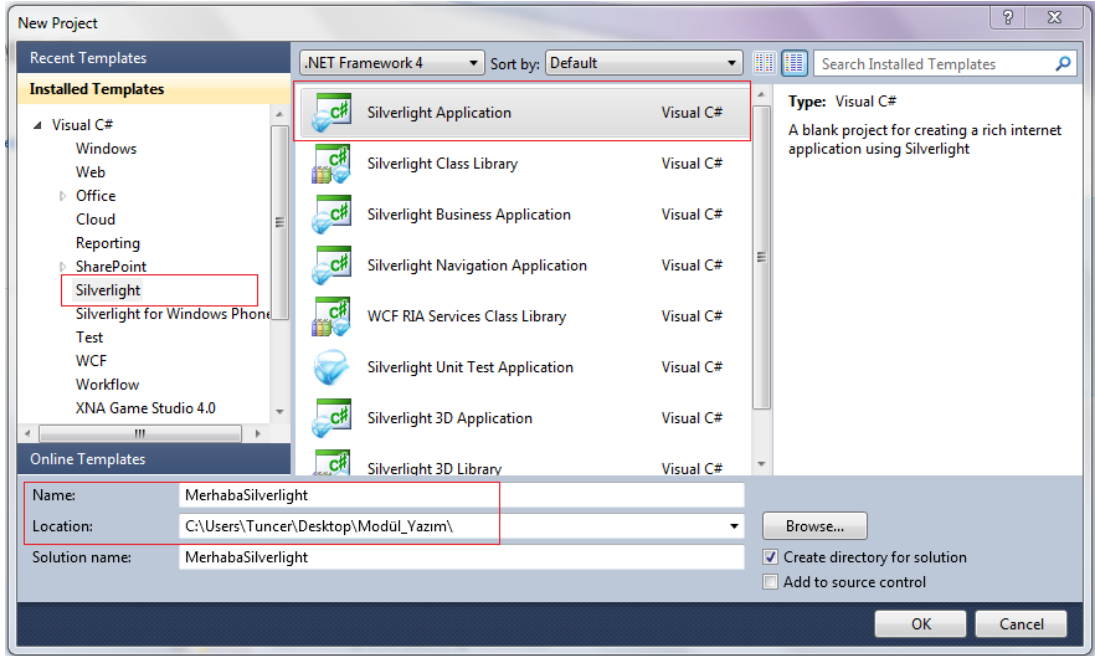
İndirmeniz gereken diğer programlar:

- Visual Studio 2010 Service Pack 1
- Silverlight 5 SDK
- Silverlight 5 Tools for Visual Studio 2010 SP 1
- Silverlight Toolkit
- Expression Blend Preview for Silverlight 5 araçlarını kurarak uygulama geliştirmeye başlayabilirsiniz.

Silverlight yine Flash gibi tarayıcı üzerinde yüklü plug-in'i ile çalışabilmektedir. Bilgisayarınızda Silverlight'in plug-in'i yoksa tarayıcı ekranına bir uyarı gelecek ve Silverlight'ı yükleyebileceğiniz adresi gösteren bir resim çıkacaktır. Bu resme tıkladıktan sonra ufak bir kurulum gerçekleştirilecektir.

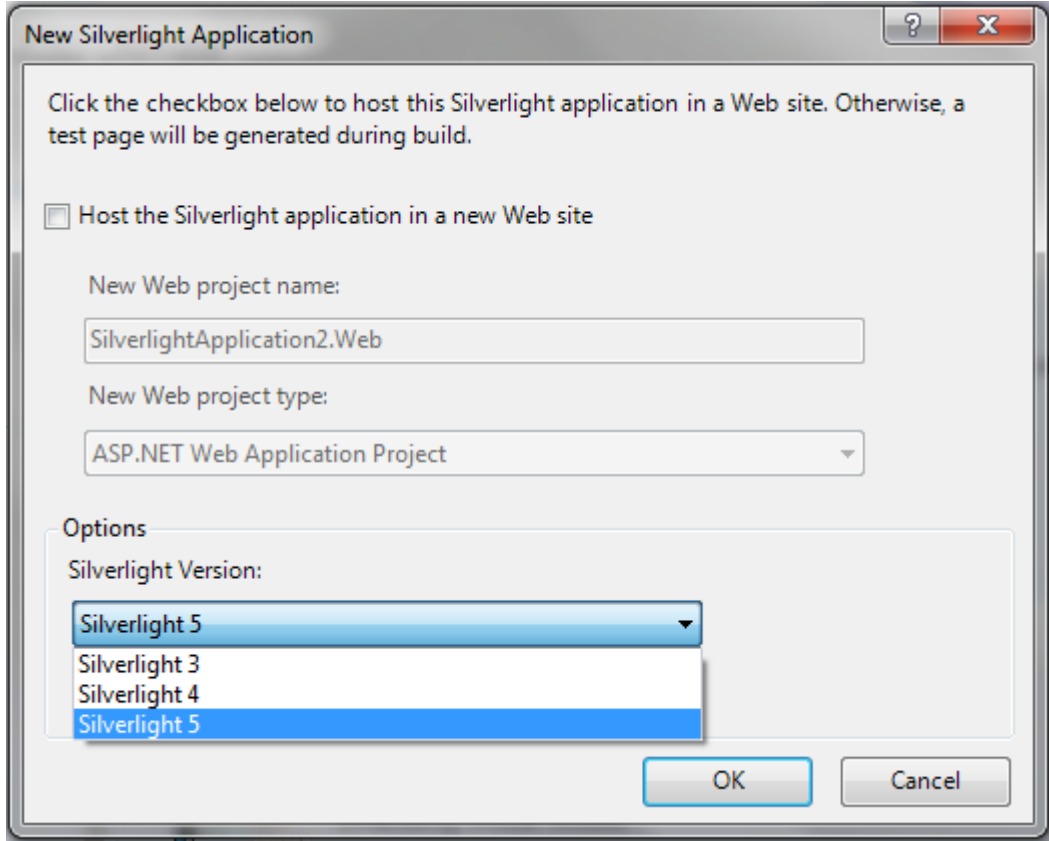
Silverlight için gerekli araçları yükledikten sonra Visual Studio 2010'da **File/New Project**'i seçeneğini seçip **Silverlight** sekmesinde yer alan **SilverlightApplication**'i seçerek ilk adımı atmaktayız.





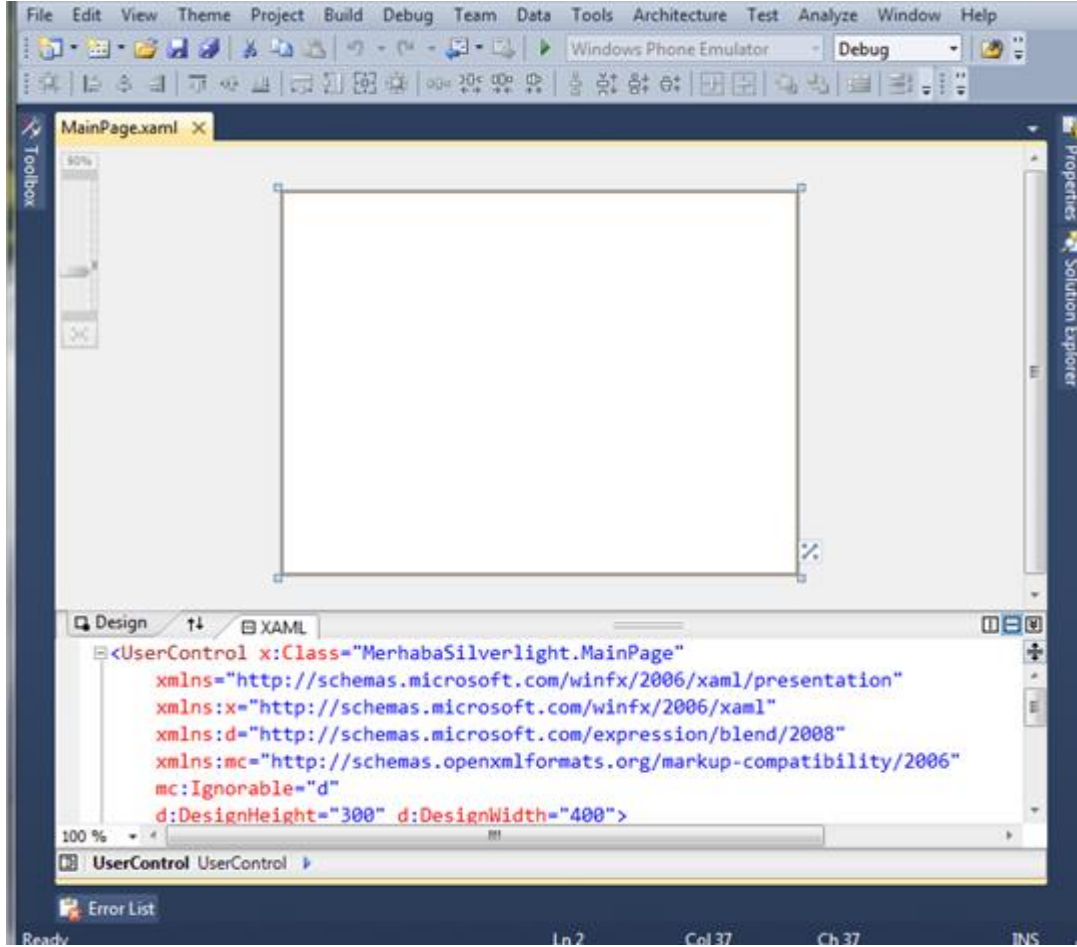
**Resim 1.2: Yeni proje oluşturma ekranı**

Uygulamaya yeni bir isim verip **OK** butonuna tıkladıktan sonra karşımıza bir ekran daha gelecektir. Bu ekranda Silverlight projesi ile beraber yeni bir ASP.NET projesinin açılmak istenip istenmediği ve hangi Silverlight sürümü ile çalışmak istediği sorulmaktadır. Burada sürüm olarak **Silverlight 5.0** seçiyoruz ve “Host the Silverlight application in a New Web site” onayını kaldırarak ASP.NET projesi olmadan, Silverlight projemizi oluşturuyoruz. Çünkü Silverlight tarayıcı üzerinde çalışacağı için runtime’da zaten bir HTML çıktısı üretecektir.



**Resim 1.3: Sistemimizde seçilebilecek Silverlight versiyonları**

Yeni bir Silverlight projesi açtıktan sonra gelen Visual Studio ekranını inceleyelim.



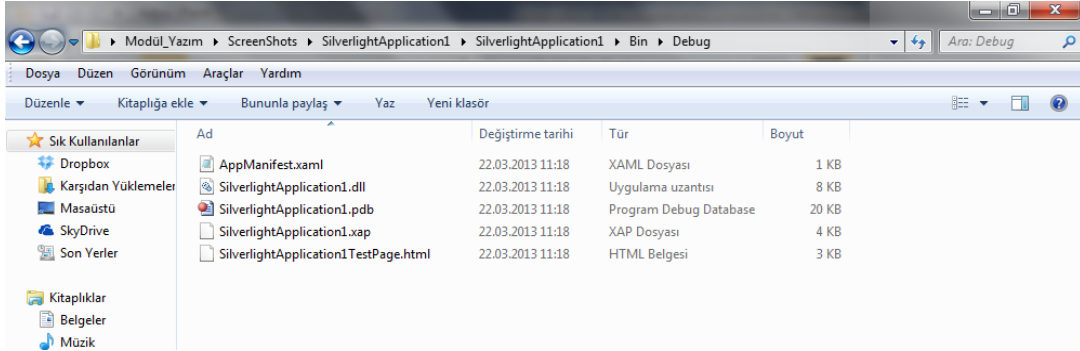
**Resim 1.4: Visual Studio 2010 ekranı**

Yukarıdaki şekle baktığımızda şeklin orta kısmında tasarım alanını görmekteyiz. Bu alanda gerek Toolbox'ı kullanarak gerekse XAML ekranını kullanarak görsel düzeni oluşturmaktayız. Tasarım alanının hemen altında tasarım alanına ait XAML kodlarını görmekteyiz. Sol ve sağ taraflarda ise Silverlight uygulaması için gerekli pencereler yer almaktadır.

Oluşturduğumuz Silverlight projesinde hiçbir işlem yapmadan hemen derleyip çalıştırılabilir ve çıktıları görelim. **F5** tuşu ile projemizi derleyerek çalıştırıyoruz. Varsayılan tarayıcı ile var olan Silverlight projemize ait bir HTML çıktısı oluşturularak tarayıcı içerisinde gösterecektir. Gördüğümüz gibi HTML uzantılı bir dosya içerisinde Silverlight uygulamamız çalıştı. Bir HTML sayfası üzerinde fare ile sağ tıkladığımız zaman Flash uygulamalarında olduğu gibi uygulamanın hangi teknoloji ile geliştirildiğini göstermektedir.

Aslında Silverlight projesi derlenip çalıştırıldığı zaman **.xap** uzantılı sıkıştırılmış bir dosya üretir. Bu dosyayı oluşturulan bir HTML sayfası nesne (object) olarak çağırır. Bu işlem Flash'ta **.swf** uzantılı dosyaları obje olarak sayfaya eklememize çok benzemektedir.

Projemizin **Bin>Debug** klasörü içerisine girdiğimizde bunu daha net bir şekilde görebilmekteyiz. Oluşturulan **MerhabaSilverlight.xap** isimli dosya sıkıştırılmış bir dosyadır. Winrar vb. herhangi bir sıkıştırılmış dosyayı açan program ile açtığımızda uygulamaya ait **xaml** dosyası ve **dll** dosyasını göreceğiz.



Resim 1.5: Silverlight uygulama klasör



Resim 1.6: xap dosyası içeriği

Oluşturulan **AppManifest.xaml** dosyası uygulama içerisinde kullanılacak **DLL**'lerin kaydedildiği dosyadır.

[**AppManifest.xaml**]

```
<Deploymentxmlns="http://schemas.microsoft.com/client/2007/deployment"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
<Deployment.Parts>
</Deployment.Parts>
</Deployment>
```

HTML çalıştığı zaman **xap** dosyası yüklenir ve ilk olarak **AppManifest.xaml** dosyası çalıştırılır. Daha sonra **AppManifest.xaml** dosyası içerisinde **EntryPointAssembly**'de tanımlanan **App.xaml** dosyası çalıştırılır.

**App.xaml** dosyasının **Application\_Startup** olayı, uygulama çalıştırıldığında hangi **UserControl**'ün ilk olarak yükleneceği bilgisi verilir. Son olarak **Mainpage.xaml** dosyası yüklenerek kullanıcı sayfa görüntülemeye başlar. İstenilirse başlangıçta ilk açılan sayfa değiştirilebilir. Bunun için **Application** sınıfın içerisindeki **Application\_Startup** metodunda tanımlı sayfayı değiştirmek yeterlidir.

[App.xaml.cs C#]

```
publicpartialclassApp : Application
{

public App()
{
this.Startup += this.Application_Startup;
this.Exit += this.Application_Exit;
this.UnhandledException += this.Application_UnhandledException;

InitializeComponent();
}

privatevoid Application_Startup(object sender, StartupEventArgs e)
{
this.RootVisual = newMainPage();
}

privatevoid Application_Exit(object sender, EventArgs e)
{
}
}
```

Oluşturulan **MerhabaSilverlightTestPage.html** dosyasını Visual Studio ile açtığımızda HTML kodları ile bir object nesnesi tanımlanıp Silverlight için gerekli tanımlamalar yapıldığını görmekteyiz.

[MerhabaSilverlightTestPage.html]

```
<!DOCTYPEhtmlPUBLIC"-//W3C//DTD XHTML 1.0
Transitional//EN""http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<htmlxmlns="http://www.w3.org/1999/xhtml">
<head><title>
MerhabaSilverlight
</title>
<stylesheetype="text/css">
html, body {
height: 100%;
overflow: auto;
}
body {
```

```

        padding: 0;
        margin: 0;
    }
    #silverlightControlHost {
        height: 100%;
        text-align:center;
    }
</style>
<scripttype="text/javascript"src="Silverlight.js"></script>
<scripttype="text/javascript">
function onSilverlightError(sender, args) {
var appSource = "";
if (sender != null&& sender != 0) {
    appSource = sender.getHost().Source;
}

var errorType = args.ErrorType;
var iErrorCode = args.ErrorCode;

if (errorType == "ImageError" || errorType == "MediaError") {
return;
}

var errMsg = "Unhandled Error in Silverlight Application " + appSource + "\n";

    errMsg += "Code: " + iErrorCode + " \n";
    errMsg += "Category: " + errorType + " \n";
    errMsg += "Message: " + args.ErrorMessage + " \n";

if (errorType == "ParserError") {
    errMsg += "File: " + args.xamlFile + " \n";
    errMsg += "Line: " + args.lineNumber + " \n";
    errMsg += "Position: " + args.charPosition + " \n";
}
elseif (errorType == "RuntimeError") {
if (args.lineNumber != 0) {
    errMsg += "Line: " + args.lineNumber + " \n";
    errMsg += "Position: " + args.charPosition + " \n";
}
    errMsg += "MethodName: " + args.methodName + " \n";
}

thrownew Error(errMsg);
}
</script>
</head>
<body>
<formmethod="post"action="MerhabaSilverlightTestPage.aspx" id="form1" style="height:100%">
<divclass="aspNetHidden">

```

```

<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/wEPDwUJNzgzNDMw
NTMzZGRjzbt7VxJSmYIMp58ttBy8xhStTTGgzRo18p8S41QHpQ==" />
</div>

<div id="silverlightControlHost">
<object data="data:application/x-silverlight-2," type="application/x-silverlight-
2" width="100%" height="100%">
    <param name="source" value="ClientBin/MerhabaSilverlight.xap" />
    <param name="onError" value="onSilverlightError" />
    <param name="background" value="white" />
    <param name="minRuntimeVersion" value="5.0.61118.0" />
    <param name="autoUpgrade" value="true" />

    <a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=5.0.61118.0" style="text-
decoration:none">
        
    </a>
</object><iframe id="_sl_historyFrame" style="visibility:hidden;height:0px;width:0px;
border:0px"></iframe></div>
</form>
</body>
</html>

```

**<object>** etiketleri arasında Silverlight uygulamamız için gerekli yapılandırma parametrelerini görmekteyiz. **Source** niteliğinde projeye ait **xap** uzantılı dosyanın konumunu belirtiyoruz. Eğer xap dosyasının konumunu değiştirmek istersek burada da yeni konumu belirtmemiz gerekmektedir. Xap dosyasını alarak farklı bir web uygulama projesine Silverlight uygulamamızı ekleyebiliriz.

```
<param name="source" value="ClientBin/MerhabaSilverlight.xap" />
```

Yine **<object>** etiketlerinde yer alan Data niteliği ise objenin **MIME** tipini belirtmektedir. Silverlight 2 uygulamaları için **MIME** tipi mutlaka **data:application/x-silverlight-2** olmalıdır (**MIME** değeri Silverlight versiyonuna bağlı olarak değişebilir).

```
<object data="data:application/x-silverlight-2," type="application/x-silverlight-2">
```

### 1.1.1. Kullanım Alanları

Silverlight kullanım alanlarını anlayabilmek için Silverlight ile karşılaştırması yapılan uygulamalara bakmak yeterli olacaktır. Kısaca bakacak olursak Silverlight İnternet üzerinde yapılacak tasarımlarda, Flash'ın kullanıldığı hemen hemen her yerde ve ek olarak kaliteli video yayını (video stream) işlemlerinde hali hazırda zaten kullanılmakta ve çıkan yeni sürümleri ile sunulacak yeni destekler ile bu yelpazenin genişlemesi beklenmektedir.

Silverlight 1.0 sürümü JavaScript dilini desteklemektedir fakat Silverlight 2.0 sürümü ile birlikte C#, Visual Basic dillerine destek vererek çok daha zengin uygulamalar geliştirmemize olanak vermiştir.

Silverlight'ı Flex ile karşılaştırabiliriz. Flex; MXML, ActionScript 3.0 ve Flex sınıf kütüphanelerinin toplamından oluşan sunucu-istemci (server-client) tabanlı bir teknolojidir.

Flex; User Interface Markup Language (Kullanıcı Arabirimi İşaretleme Dili) sınıfına dâhil edilecek geleceğin popüler web tabanlı teknolojilerinden biridir. Flex geliştiricileri **Flex Builder IDE** ortamında Coldfusion, Asp, JSP, PHP gibi popüler web programlama dilleri yardımıyla etkileşimli ve güvenli web tabanlı projeler geliştirmektedir. İstemci tarafında kullanıcılara ise sadece .swf formatında oluşturulmuş Flash dosyalarını gösterir. Flash'ta da durum aynıdır. Yapılan .fla uzantılı tasarım dosyaları .swf formatına dönüştürülerek sıkıştırılır ve çalıştırılabilir bir sürüm hâline getirerek *İnternet* üzerinde yayımlanır. Böylece yapılması muhtemel kaynak kodu çalınmalarının önüne geçilir.

Fakat WPF teknolojisi üzerine yapılan geliştirmeler sonucu piyasaya sürdüğü Silverlight açık kaynak bir projedir. Silverlight ile yapılan tasarımlar herhangi bir sıkıştırılma işlemine tabi tutulmadan *İnternet* üzerinde oynatılabilir ve paylaşılabilir. Böylece içeriğin farklı kişiler tarafından alınarak üzerinde yapılacak küçük değişiklikler ile kullanılmasına izin verilmiştir.

Pazar payı her geçen gün artan ve yaygın olarak kullanılmaya başlayan işletim sistemi yüklü telefonlarda uygulama geliştirmek için Silverlight teknolojisi kullanılmaktadır.

### 1.1.2. Uygulama Geliştirme ve .NET Platformu İlişkisi

Hızla gelişen bir teknoloji alanı olan yazılımın son kullanıcıya ulaştığı noktada kullanıcı arayüzleri karşımıza çıkmaktadır. Geliştirilen bir yazılımın işlevi ve performansı çok önemlidir. Günümüzde performans ve işlevin artık tek başlarına yeterli olmadığını net bir şekilde görmekteyiz. Ekran kartlarının, işlemcilerin ve geçici belleklerin güçlerinin artması kullanıcı arayüzlerinin yetersiz kaldığını ifade etmektedir. Bu noktada programlama teknolojilerinin yeteneklerini iyi kullanmak gerekecektir. Çok uzağa gitmeden bakmak gerekirse birçok yazılımcı standart form uygulamaları geliştirmektedir. Yazılımcıların uygulamalarında 2D-3D çizimler ya da animasyonlar yapabilmek veya kullanabilmek için **GDI, GDI+, OpenGL, DirectX** kütüphanelerini kullanmaları gerekmektedir. Bu işlemleri kullanmanın dezavantajı grafik uygulamalarının RAM belleği kullanması ve CPU'da derlenmesidir. Bu sebepten yapılan uygulamaların performansında büyük kayıplar yaşanmaktadır. Ayrıca yapılan bu animasyonlarda tutarsız akışlar, grafiksel çizimlerde pixel kayıpları yaşanması da karşılaşılabilen bir ihtimaldir.

Bu yüzden kullanıcı etkileşimli uygulamalar konusunda destek olabilecek çok önemli bir teknoloji.NET Framework 3.0 ailesi ile birlikte lanse edilmiş ve ilk duyurulduğunda **Avalon** kod ismi ile anılmıştır.



**WPF**, yazılım geliştiriciler için devrim niteliğinde bir yeniliktir. Eskiden diyorum, bir form uygulaması geliştirmek için sadece geliştirici yeterliydi. Web tarafında bu iş, geliştirici ve tasarımcının ortak çalışması sonucunda gerçekleşiyordu. Bu durum tasarım ve kod alanının farklı yerlerde yer almasından kaynaklanıyordu. Form uygulamalarında tüm iş geliştiriciye düşüyor ve eksik tasarımlar ortaya çıkabiliyordu.

WPF teknolojisine baktığımızda ilk olarak tasarım ve kod alanlarının birbirinden ayrıldığı göze çarpmaktadır. Bu yenilik sayesinde hem tasarımcı hem de geliştirici kendi alanlarında daha verimli olabilecektir.

- **WPF**'in özelliklerini incelersek;
  - Tasarım ve kodlama alanlarının birbirinden bağımsız olması ile beraber tasarım kodlaması için **XAML** isimli işaretleme dili çıkmıştır.
  - Windows Form'da **GDI+**kütüphanesi yerine **Direct-X** altyapısı kullanılmaktadır.
  - Silverlight teknolojisi ile web ve mobil alanlar üzerinde tüm yeteneklerini göstermektedir.
  - WPF piksel tabanlı değil, vektör tabanlı bir teknolojidir. Bu yüzden grafik uygulamalarında işlemci ve belleği kullanmak yerine ekran kartının işlemcisini ve belleğini kullanır. Grafik uygulamalarında performansın artması, kaynak tüketimini minimuma indirmektedir.
  - 3D kütüphanelerini içerisinde bulundurur.
  - Form uygulamalarında 2D işlemleri için **GDI+**, 3D işlemleri için **Managed Direct API**, görüntü desteği için **Media Player API** kütüphanelerine ihtiyaç vardır. Tüm bu işlemler **WPF** ile desteklenmiştir
  - Açık Kaynak Doküman formatı olarak bildiğiniz **XPS** dosyalarını desteklemektedir. Kolay bir şekilde XPS okuyucular yapmak mümkündür.

WPF teknolojisi, bu özellikleri ile daha yetenekli uygulamalar geliştirmemize imkân sağlayacaktır.

- Silverlight'a ilk bakışta WPF'in tarayıcı üzerindeki versiyonu diyebiliriz. Günümüze geldiğinde versiyon değiştirerek gelişimini devam ettirmektedir. Şuana kadar lanse edilen versiyonları aşağıdaki gibidir.
  - Silverlight 1.0
  - Silverlight 2.0
  - Silverlight 3.0
  - Silverlight 4.0
  - Silverlight 5.0

Silverlight görselliği yüksek, hız ve performans açısından oldukça verimli olmasından dolayı Flash teknolojisine eş değer bir teknoloji olarak gösterilmektedir. Bunun en büyük sebebi ise Silverlight ile **RIA (Rich Internet Application)** uygulamaları geliştirebilmemizdir.

Flash teknolojisi ile geliştirilen uygulamaların tarayıcı üzerinde gösterilebilmesi için tarayıcıya Flash'ın plug-in'in yüklenmesi gerekmektedir.

- Silverlight teknolojisinin teknik özelliklerini aşağıdaki gibi listeleyebiliriz:
  - Tasarım ve kodlama alanlarının birbirinden bağımsız olması ile beraber tasarım kodlaması için XAML isimli işaretleme dili kullanılmaktadır.
  - Silverlight piksel tabanlı değil vektörel tabanlı bir teknolojidir.
  - XML, JSON, RSS ve Atom veri formatlarını destekler. Ayrıca SOAP ve RESET hizmetleri ile birlikte çalışabilir.
  - IIS Smooth Stream kullanarak HD video ve içerik koruma yerleşik desteği ile Silverlight iş ve tüketici senaryoları için online ve offline uygulamalar sağlar.
  - Ses / Video desteği (VC-1, WMV, WMA, MP3), görüntü desteği (JPG, PNG) ve H.24 video ve AAC ses desteği vardır.
  - Yazdırma desteği, zengin raporlama, grafik ve SharePoint gibi sunucu ürünleri ile entegrasyon sayesinde iş uygulamaları geliştirmeye imkân sağlar.

**Rich Internet Application (RIA)**, resim, video veya diğer görselliklerin İnternet ortamında kullanılmasını tanımlar. Kısaca yüksek kalitede görsel efektler ve animasyonlarla desteklenebilen ekranları ifade etmektedir.

### 1.1.3. Programlama ve Tasarım İlişkisi

Silverlight ile programlama, karmaşık bir yapıya sahip olmasına rağmen çok kolay öğrenilebilecek bir teknolojidir. Her ne kadar yeni teknoloji olsa da var olan programlama dilleri ile geliştirilebilmektedir. Silverlight'ı kolay bir şekilde kavrayabilmek için Visual C# veya Visual Basic programlama dillerinden herhangi birini bilmek gerekmektedir.

Silverlight teknolojisinde tasarım veya kod ayrımı geldiğinden dolayı tasarımcılar, tasarımlarını yaparken farklı programların gereksinimine ihtiyaç duymaktadır. Bu ihtiyacı karşılamak için **Expression Studio** ürün ailesi gelmiştir. Bu ürün ailesinde çoklu ortam uygulamaları için gerekli tüm programlar geliştirilmiş durumdadır.

**Expression Studio** ailelerinin ürünlerini şu şekilde özetleyebiliriz.

- **Expression Web:** Dreamweaver ve Frontpage'e benzer bir HTML editörü gibi görünse de geliştirici ve tasarımcıların işini kolaylaştıracak birçok özelliği barındırmaktadır.
- **Expression Encoder:** Büyük boyutlu videolarımızı İnternet ortamında daha verimli kullanabilmek için geliştirilmiş bir **encoder** programıdır. Video üzerinde kırpma, ekleme, parçalama gibi işlemleri rahatlıkla yapabiliriz. Ayrıca canlı yayın uygulamaları konusunda da yardımcı olmaktadır.
- **Expression Design:** Resimler üzerinde işlemler ya da yeni tasarımlar için vektörel çizimler yapmamıza olanak sağlamaktadır. En önemli özelliği; vektörel

çizimleri matematiksel veriler şeklinde alabiliyor olmasıdır. Bu sayede yaptığımız çizimleri farklı programlar üzerinde hiç bir renk ve biçim kaybı olmadan kullanabiliriz.

- **Expression Web SuperPreview:** Bu program, web tasarımcılarının işini oldukça kolaylaştıracaktır. Çünkü yapılan web sitelerini farklı tarayıcılarda aynı anda test etmemizi sağlamaktadır. Yani web sitenizin iki farklı tarayıcıda nasıl gördüğünü aynı ekranda görebiliyor ve rahatlıkla farklılıkları tespit edebilirsiniz.
- **Expression Blend:** WPF ve Silverlight teknolojilerinin tasarımlarını ve az da olsa kodlamasını yapabileceğimiz bir programdır. Özellikle tasarım ve animasyon işlemlerinde oldukça ihtiyaç duyulmaktadır. Dikkat çeken en önemli özelliği; Visual Studio formatında projeler açarak Visual Studio ile senkron çalışabilmesidir. Ayrıca Photoshop ve Illustrator gibi önemli programların dosya formatlarını proje haline alabilmektedir. Bu sayede profesyonel vektörel çizimleri ya da tasarımları WPF ve Silverlight projelerinde kullanılabilir.

Silverlight ile uygulama geliştireceğimiz zaman **Visual Studio** ve **Expression Studio** ürünlerine ihtiyacımız olacaktır.



**Resim 1.7: Expression Blend uygulama ekranı**

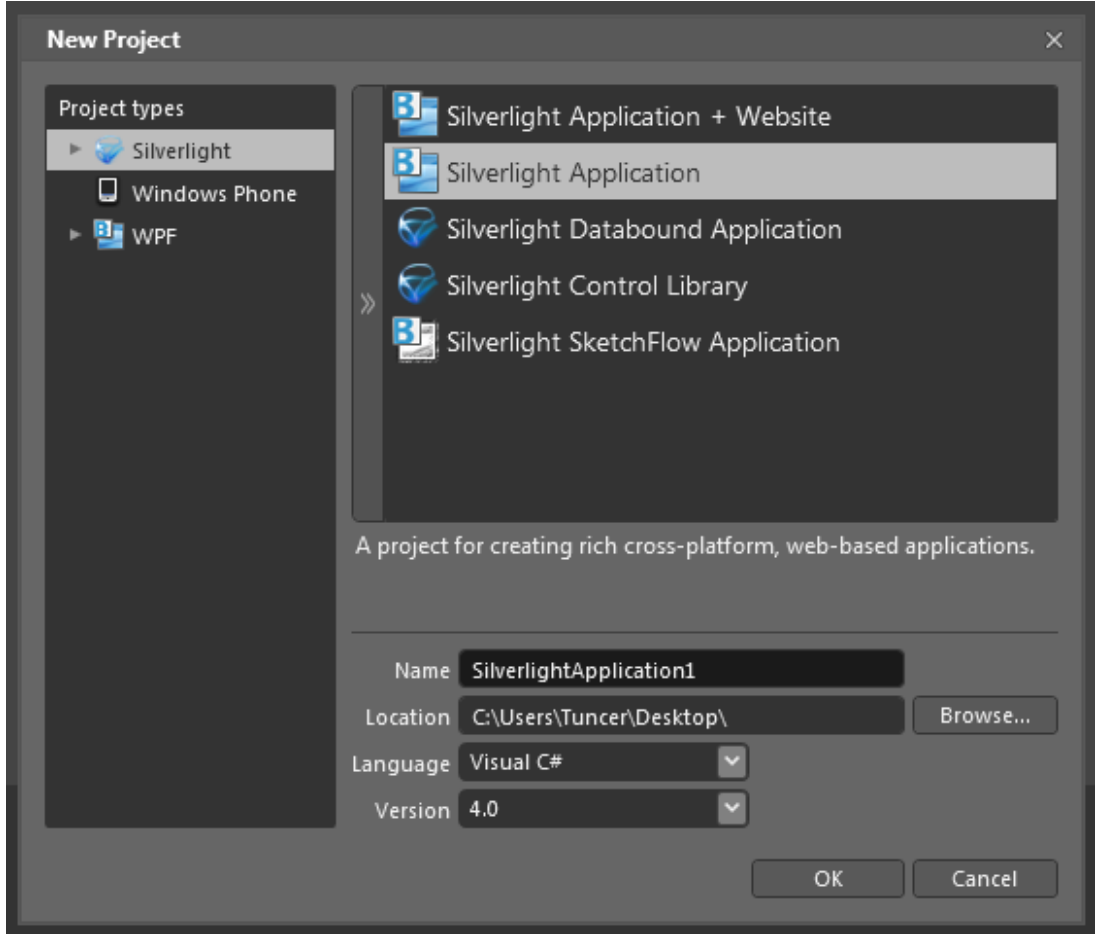
Expression Studio ürün ailesinin bir üyesi olan **Blend** ile WPF ve Silverlight uygulamalarında tasarım ve animasyon ihtiyacını kolaylıkla karşılayabiliriz. **Blend**, diğer Expression Studio ürünleri ile yapılanları da proje içerisine alabilmektedir. Yani diğer programlar ile senkron çalışabilme özelliğine sahiptir.

**Blend** ile yeni bir WPF projesi açıldığında, aslında proje dosyalarını Visual Studio için üretilmiş olduğunu görmekteyiz. Bu yüzden Silverlight projesini Blend ile tasarlarken bir yandan da Visual Studio ile geliştirebiliriz.

Blend ile neler yapılabileceği aşağıda anlatılmaktadır.

- **Expression Design** ile oluşturulan vektörel çizimleri XAML kodunu alarak Blend içerisinde, bu vektörel çizimi nesne olarak kullanabiliriz.
- Silverlight uygulamalarında Visual Studio ile sayfalarca kod yazarak yapılan animasyonları görsel olarak kod yazmadan oluşturabiliriz. Her yapılan işlemin XAML kodları otomatik olarak üretilmektedir.
- İçerisinde bulunan tasarım araçları ile rahatlıkla vektörel çizimler veya şekiller oluşturabiliriz.
- Blend içerisinde bulunan **Data** penceresi ile basit veri kaynakları oluşturabiliriz. Ayrıca bu kaynakları kolay bir şekilde kontrollere bağlayabiliriz (binding).
- **Sketchflow** mekanizması sayesinde uygulama prototipi oluşturmamıza imkân sağlar.
- Adobe Photoshop ve Illustrator gibi üst düzey görüntü işleme programlarıyla yapılan vektörel çizimleri ve profesyonel şablonları katmanlar hâlinde Silverlight uygulamalarına dâhil edebiliriz.
- Silverlight uygulamasında bulunan tüm nesne ve kontrollerin X, Y, Z koordinatlarını görsel olarak değiştirerek 3D görünüm kazandırabiliriz.

Blend ile yeni Silverlight projesini program açılışındaki New Project ekranındaki popup yardımı ile gerçekleştirebiliriz. Bu popup'ta dikkat etmeniz gereken noktalar, uygulamanın geliştirileceği programlama dilinin seçilmesi ve hangi sürüm ile çalışacağını belirtmesi olacaktır.



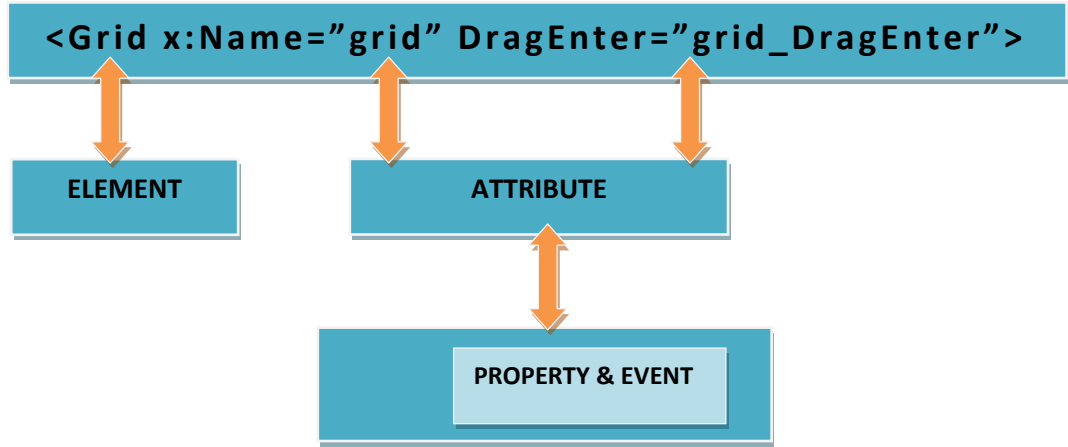
Resim 1.8: Blend yeni proje ekranı

#### 1.1.4. XAML

**XAML**, Silverlight ve WPF teknolojilerinin tasarım kısımlarını geliştireceğimiz statik ve dinamik kullanıcı arayüzleri oluşturmamızı sağlayan işaretleme dilidir. XML'in geliştirilmiş hali gibi görünmekte ve uzantısı **.xaml**'dir. Ayrıca ASP.NET ile uygulama geliştirirken tasarımlarını oluşturduğu yapıya da oldukça benzemektedir.

XAML bir programlama dili olmamakla beraber uygulama mantığı ve olaylar da codebehind dosyalarında bulunmaktadır.

Her XAML elementi aslında .NET CLR sınıfını temsil eden niteliklerin XML görünümüdür. Bu yüzden herhangi bir nesnenin özelliklerini rahatlıkla değiştirebiliriz. Form uygulamalarından bildiğimiz **property** kavramı ve **event**'lar XAML'de **attribute** olarak adlandırılır. Tüm XAML attribute'leri kod tarafından düzenlenebilir. Yani, standart uygulamalarda olduğu gibi bir TextBox'ın arkaplan rengini, yüksekliğini ya da pozisyonunu kod içinde kolay bir şekilde değiştirebilirsiniz.



**Resim 1.9: Xaml Attribute ve Element Yapısı Örneği**

XAML’de **Tree View** yapısı ile nitelikleri kullanabilmemizin yanı sıra, kullandığımız kontrolün içerisinde de kullanabilmemize imkân sağlamaktadır. XAML kodları **.xaml** uzantılı bir dosya içerisinde saklanmaktadır. Ayrıca XAML dosyasının altındaki codebehind dosyası ise **.xaml.cs** veya **.xaml.vb** uzantılı olmalıdır. Yani her XAML dosyasının altında C# veya VB’ye ait codebehind dosyası yer alabilir.

XAML’in formatına baktığımızda genel programlama mantığına uyduğunu görmekteyiz. Üst kısımda namespace (isim uzayı) tanımlamaları yer almaktadır. Temel olarak bir Silverlight projesinde tasarım ekranı **UserControl** olarak isimlendirilir. Bu yüzden XAML kodları <UserControl> etiketleri arasına yazılmaktadır.

**xmlns** (XML Name Space) ile etiketlenen bölüm XAML’ın kullandığı referansları ifade etmektedir. Yukarıdaki uzay tanımlamalarımızdaki gibi XAML içerisinde kullanacağımız kodların uzaylarını **xmlns** etiketi ile gerekmektedir.

Örneğin; TextBox, Button ve Label gibi kontrolleri XAML ile kontrol etmek için `xmlns=http://schemas.microsoft.com/winfx/2006/xaml/presentation` uzayını referans olarak eklemek gerekmektedir.

Genel olarak XAML yapısının XML’e benzer.

XAML’de ilk tanımlanan eleman, root eleman olarak adlandırılır. UserControl bir root elemanıdır. Root elemanlarının kapsadığı panel elemanları kullanıcı arayüzleri için StackPanel, DockPanel, Grid ve Canvas gibi düzen kontrolleri sunar.

Kontrol elemanları XAML elementlerinin farklı tiplerini tanımlayıp düzenlememizi sağlar. Geometrik elemanlar ise uygulama arayüzü üzerinde grafik ve şekiller çizmemizi sağlar.

## UYGULAMA FAALİYETİ

Aşağıda verilen işlem basamaklarını takip ederek konuyu pekiştiriniz.

İşlem Basamakları	Öneriler
➤ Visual Studio 2010'u kurunuz.	
➤ Visual Studio 2010 Service Pack 1'i kurunuz.	
➤ Silverlight 5 SDK'yı kurunuz.	
➤ Silverlight 5 Tools for Visual Studio 2010 SP 1'i kurunuz.	
➤ Silverlight Toolkit'i kurunuz.	
➤ Tarayıcınız için Silverlight Plug-in'i kurunuz.	
➤ Microsoft Expression Blend Preview for Silverlight 5'i kurunuz.	➤ <a href="http://www.microsoft.com/en-us/download/details.aspx?id=9503">http://www.microsoft.com/en-us/download/details.aspx?id=9503</a>
➤ Bütün araçlar kurulduktan sonra sistemi yeniden başlatınız.	➤ Başlat > Kapat > Yeniden Başlat'ı seçiniz.

### KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri **Evet**, kazanamadığınız becerileri **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Visual Studio 2010'u kurduunuz mu?		
2. Visual Studio 2010 Service Pack 1'i kurduunuz mu?		
3. Silverlight 5 SDK'yı kurduunuz mu?		
4. Silverlight 5 Tools for Visual Studio 2010 SP 1'i kurduunuz mu?		
5. Silverlight Toolkit'i kurduunuz mu?		
6. Tarayıcınız için Silverlight Plug-in'i kurduunuz mu?		
7. Expression Blend Preview for Silverlight 5'i kurduunuz mu?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Silverlight uygulamalarında görsel arayüz hangisi ile hazırlanır?  
A) XML  
B) HTML  
C) XAML  
D) AJAX
2. XAML için aşağıdakilerden hangisi **yanlıştır**?  
A) XML-tabanlı deklaratif bir dildir.  
B) Silverlight uygulamalarının kullanıcı arayüzündeki elementlerini oluşturmak için kullanılır.  
C) Silverlight uygulamalarında elementleri uygun bir düzende yerleştirmek için kullanılır.  
D) Açılımı şöyledir: Xapplication Accaptable Markup Language
3. Projeye eklediğin AnaSayfa.xaml dosyasının C# kodları, varsayılan olarak hangi dosyada yer alır?  
A) AnaSayfa.cs  
B) AnaSayfa.xaml.cs  
C) AnaSayfa.xml.cs  
D) AnaSayfa.xaml.aspx
4. Yeni bir Silverlight projesi açtığında uygulamanın ilk çalışması, yeniden aktifleştirilmesi, kapanması vb. durumlarda ele alınabilecek metodlara sahip varsayılan sınıfı aşağıdakilerden hangisi içinde yer alır?  
A) App.xaml.cs  
B) App.debud.cs  
C) App.xaml  
D) MainPage.xaml
5. Aşağıdakilerden hangisi Silverlight uygulamasında arayüze kontrol eklemenin yollarından birisi **değildir**?  
A) Toolbox penceresinden tasarım ekranına sürükleyip bırakmak.  
B) XAML yazarak eklemek  
C) ControlBox.xaml dosyasından arayüze eklemek.  
D) C# ile kod tarafında eklemek.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme faaliyetine geçiniz.

# ÖĞRENME FAALİYETİ-2

## AMAÇ

Gelişmiş *İnternet* uygulamaları için temel uygulama dosyalarını oluşturabileceksiniz.

## ARAŞTIRMA

- Asp.Net MVC(Model View Controller) kavramını araştırınız.

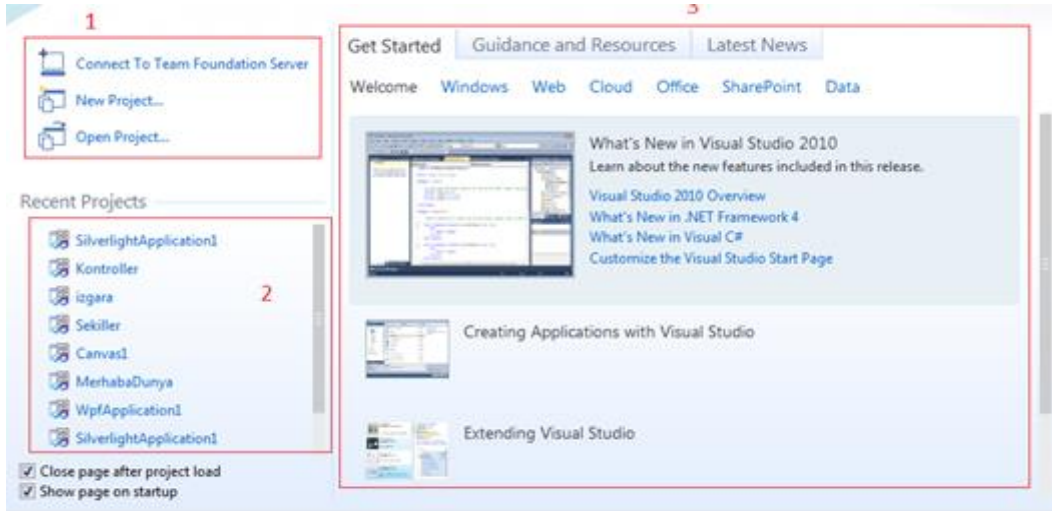
## 2. BAŞLANGIÇ UYGULAMALARI

Bu bölümde adım adım bir Silverlight uygulamasının nasıl oluşturacağımızı öğreneceğiz.

### 2.1. Örnek Uygulama Dosyaları Oluşturma

#### 2.1.1. Betik Dosyası Oluşturma

Silverlight projesi oluşturmak için Visual Studio 2010'u açıyoruz. Karşımıza Start Page başlangıç ekranı gelir.



Resim 2.1: Visual Studio Start Page ekranı

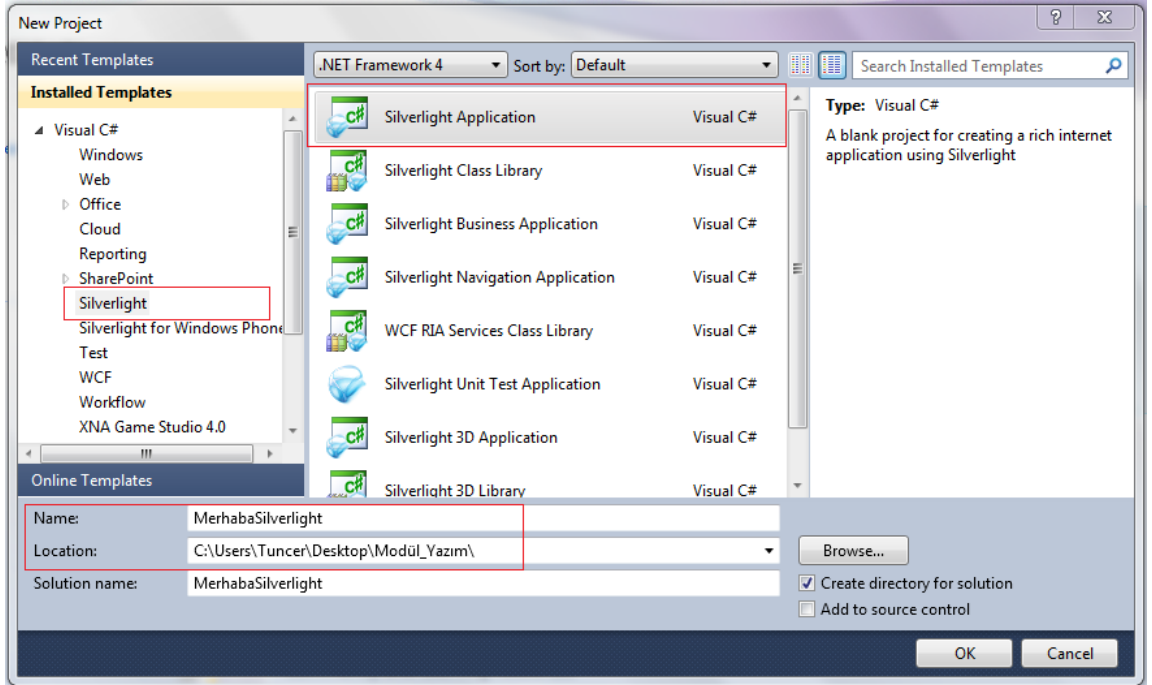
Resimde gördüğümüz gibi **Start Page** sekmesi 3 bölümden oluşur.

**1. bölümde** yeni proje oluşturabilir veya var olan bir projeyi açabiliriz.

2. bölümde en son açılan projelerin bir listesi görünür. Böylece en son işlem yaptığınız projeye hızlı bir şekilde ulaşabilirsiniz.

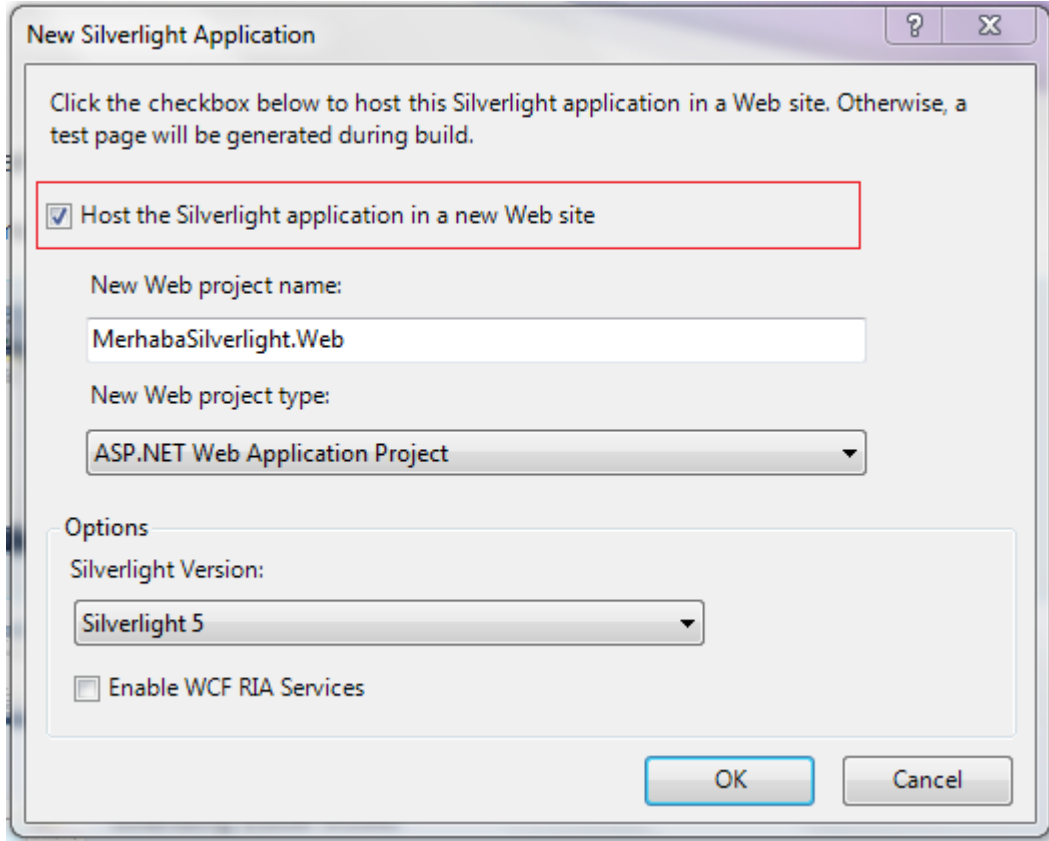
3. bölümde ise RSS kayıtları ekrana gelir. Burada isterseniz RSS'i iptal edebilirsiniz.

Şimdi **New Project** bağlantısına tıklıyoruz ya da menülerden **File > New > Project**'i seçiyoruz. Karşımıza yeni proje oluşturma penceresi gelir.



Resim 2. 2: New Project ekranı

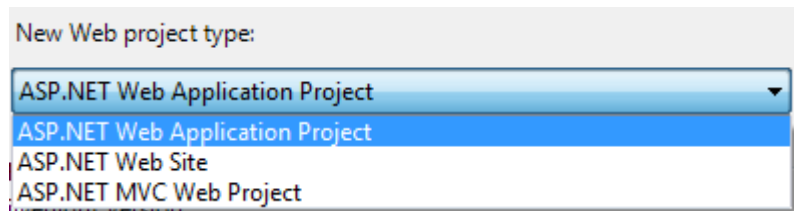
Sol taraftaki **Installed Templates** kısmından Silverlight'ı seçiyoruz. Sağ tarafta oluşturulabilecek Silverlight uygulama tipleri ekrana gelir. Biz **Silverlight Application** seçiyoruz. Alt tarafta **Name** ve **Location** kısımlarında uygulamamızın ismini ve kaydedileceği yeri belirttikten sonra **OK** butonuna tıklıyoruz.



**Resim 2.3: Onay kutusunun seçili olması**

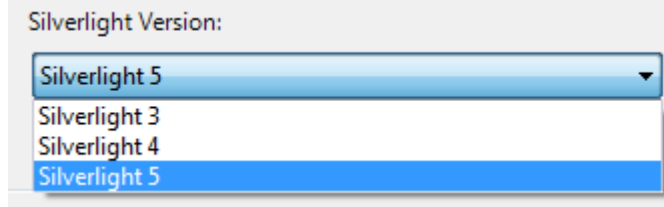
Karşımıza gelen ekranda “**Host the Silverlight Application in a new Web site**” seçeneğinin seçili olmasına dikkat edin. Eğer bu seçenek seçili ise projenize, Silverlight objesinin çalıştırılabileceği bir test Web sitesi eklenir. Böylece Silverlight çalıştırabilmek için gerekli ayarlamaları Visual Studio, bizim yerimize otomatik olarak yapacaktır.

**New Web Project name** kısmında, oluşturulacak olan Web projesinin ismi yazılır. Hemen altında ise **New Web project type** kısmı bulunmaktadır. Bu kısımda ise çeşitli ASP.NET proje tipleri bulunur. Aşağıda gördüğümüz gibi **ASP.NET Web Application Project**’i seçiyoruz.



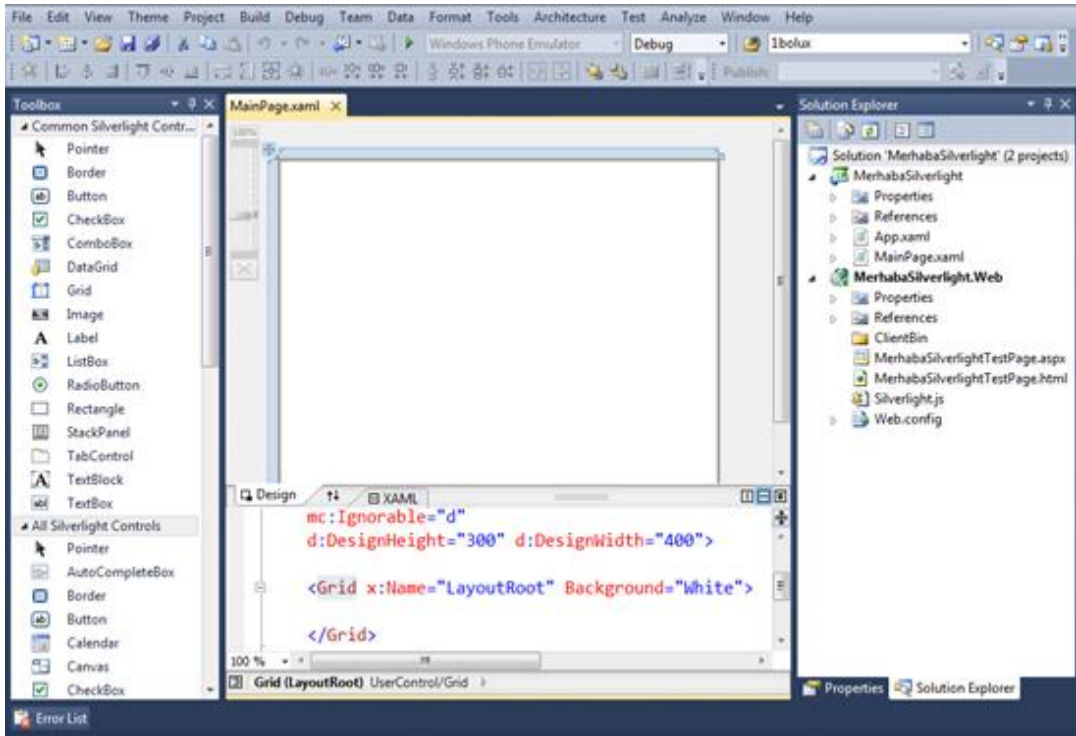
**Resim 2.2: Seçebileceğimiz proje türleri**

En altta bulunan Silverlight Version kısmında ise bilgisayarımızda kurulu olan Silverlight sürümleri listelenir. Biz, şu anda **Silverlight 5** ile çalışıyoruz ama isterseniz önceki sürümlerle de çalışabilirsiniz.



**Resim 2. 5: Uygulamamızda kullanabileceğimiz Silverlight versiyonları**

**Enable WCF RIA Services** seçeneğini seçerek RIA hizmetlerini aktif hale getirebilirsiniz. Bu örneğimizde RIA servislerini kullanmadığımız için burayı işaretlemiyoruz. **OK** butonuna basarak projemizi oluşturuyoruz. İlk Silverlight projemiz ekrana gelecektir.



**Resim 2.6: Visual Studio ekranı**

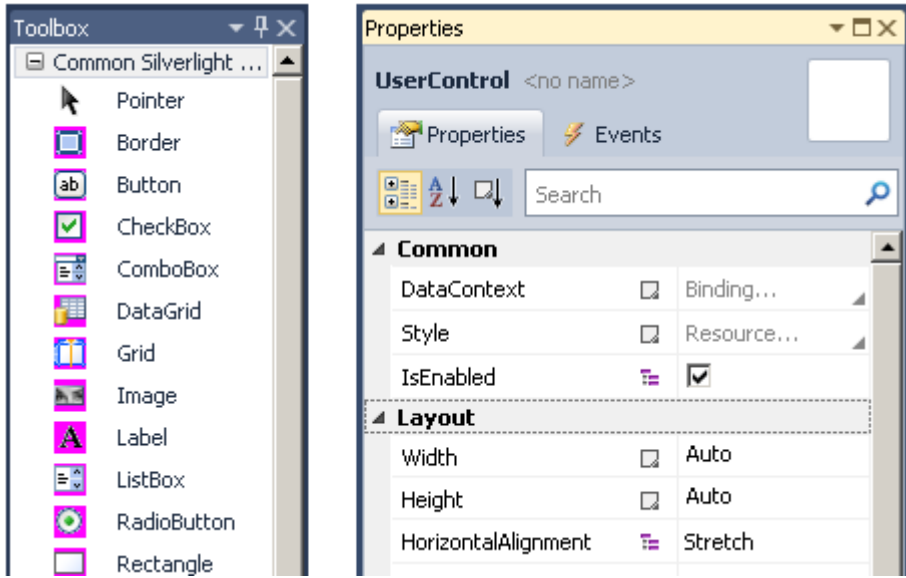
**Visual Studio** penceresinin ortasında bulunan beyaz dikdörtgen **Design** (Tasarım) ekranıdır. **ToolBox** penceresinden sürükleyip bırakarak nesne ekleyebilir ve sayfa tasarımınızı yapabilirsiniz.

**Design** ekranının altında **XAML** ekranı görülmektedir. **Design** ekranında yaptığınız değişikliklere bağlı olarak XAML ekranının içeriği otomatik olarak güncellenir. Aynı

şekilde **XAML** kodlarında yaptığımız değişikliklere bağlı olarak **Design** ekranı güncellenir. **XAML** ekranında kod yazarken **IntelliSense** (kod yardımı) penceresi yardımcımız olacaktır.

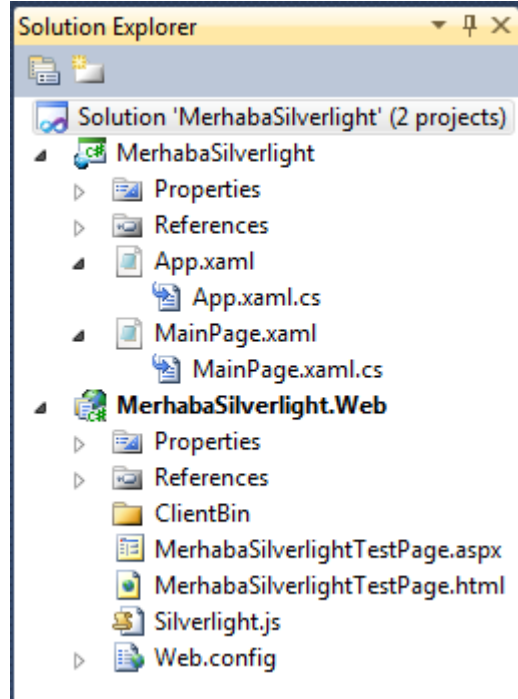
XAML yapısına baktığımızda söz dizimi açısından XML yapısı gibi olduğunu görürüz. Kodlarımıza baktığımızda Silverlight uygulamamızın **UserControl** elementi ile başladığımızı göreceğiz. Kodlarımızda, XAML söz dizimi yapısı için gerekli tanımlamalar vardır.

Hemen sol tarafta, Silverlight projesinde kullanabileceğimiz kontrollerin bulunduğu **Toolbox** penceresi bulunmaktadır. Sağ tarafta ise **Properties** penceresi bulunur. Bu pencerede, kontrollere ait özellikleri görsel olarak değiştirebilirsiniz. Visual Studio'nun önceki sürümlerinden farklı olarak, properties penceresi biraz daha görsel özelliklere sahip olarak karşımıza gelmektedir.



**Resim 2.3: ToolBox ve Properties pencereleri**

Sağ tarafta projemize ait dosyaların bulunduğu **Solution Explorer** penceresini görebilirsiniz.

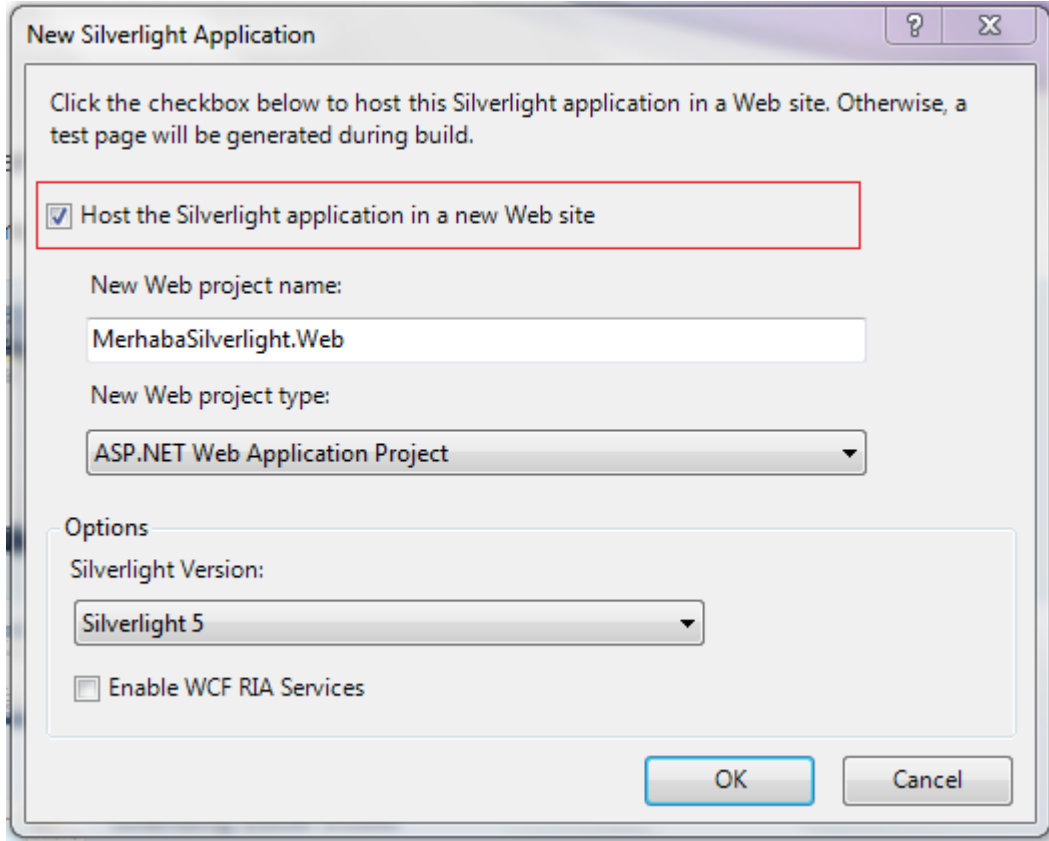


**Resim 2.4: Solution Explorer penceresi**

Klasik Windows Form uygulamalarının aksine 2 proje tipi vardır:

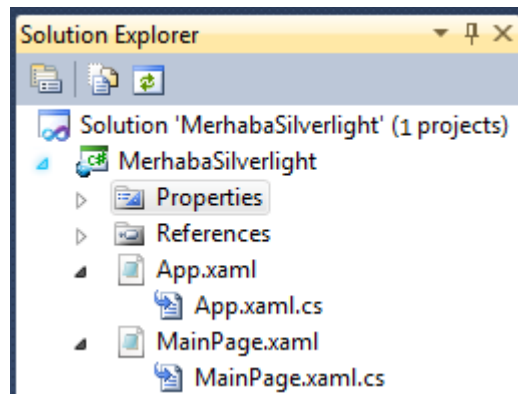
- MerhabaSilverlight
- MerhabaSilverlight.Web

Elimizde bir object nesnesi bulunur ve bu object'in gömülü olduğu bir Web projesi bulunur, Silverlight bu mantıkla çalışmaktadır. Uygulamamız ayrı bir object olarak ele alınır ve uygulamamızın gömülü olduğu Web projesi ayrı ele alınır. Visual Studio ile yeni proje oluştururken hatırlarsanız, aşağıdaki ekranda görülen seçeneği seçmiştik. Bu sayede, Visual Studio bize Silverlight objemizin gömüleceği bir Web projesini de otomatik oluşturmuştu.



**Resim 2.5: New Silverlight Application penceresi**

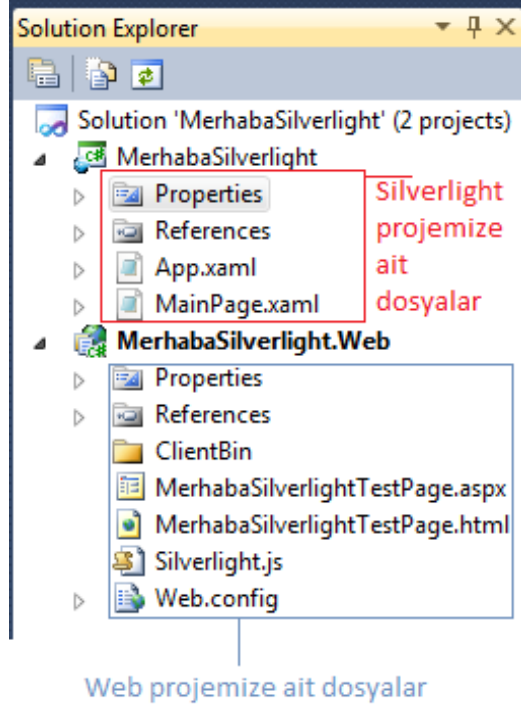
Eğer yukarıdaki başlangıç penceresinde “**Host the Silverlight Application in a new Web site**” seçeneğini seçmeseydik, aşağıdaki gibi bir Solution Explorer penceresi karşımıza gelecekti. Bu pencerede sadece Silverlight uygulamamıza ait dosyalar görünecekti, herhangi bir Web sayfası oluşturulmayacaktı.



**Resim 2.10: Silverlight projemize ait dosyalar**

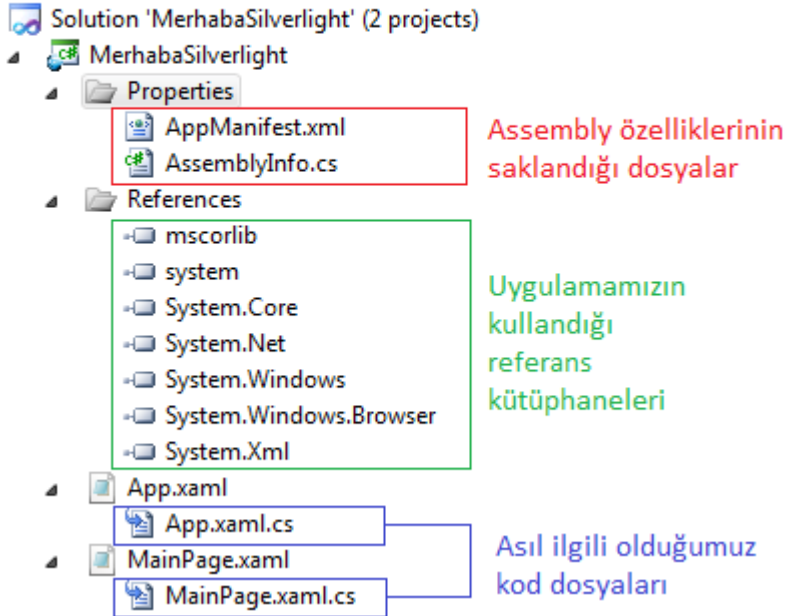


Silverlight uygulamamıza ait dosyalar ve Web sitemize ait dosyalar Solution Explorer pencerede listelenir.



**Resim 2.11: Projemize ait tüm dosyalar**

Dosyaların yanındaki + işaretine tıklarsanız yeni dosyalar görünecektir.



Resim 2.12: Proje ağacımızın açılmış hâli

Bu ekranı incelersek:

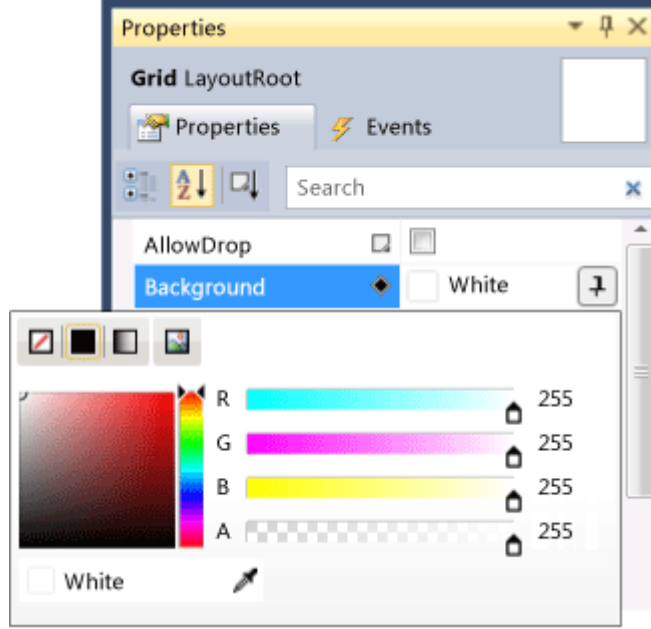
**App.xaml:** Bu dosya, Silverlight uygulamamıza ait birtakım global özelliklerin tutulduğu dosyadır. Tasarım görüntüsü yoktur sadece kodlardan oluşur.

- **App.xaml.cs:** Bu dosya, App.xaml dosyasına ait C# kodlarının tutulduğu dosyadır. Dosyaya baktığınızda kendi içinde Application\_Startup() ve Application\_Exit() metodlarına sahip olduğunu görürsünüz. Bu metodlar ile uygulama başladığında ve kapatıldığında birtakım işlemler yapabilirsiniz. Örneğin; uygulama başladığında “Merhaba” uygulama kapatıldığında “Hoşçakalın” mesajları vermek istiyorsanız bu metodlara kod yazmanız yeterli olacaktır.
- **MainPage.xaml:** Bizim sayfa tasarımını gerçekleştirdiğimiz asıl Silverlight sayfamızdır.
- **MainPage.xaml.cs:** Sayfamıza ait kodların bulunduğu C# dosyasıdır.

**Not:** Eğer projedeki herhangi bir pencereyi göremiyorsanız **View** menüsünden **Other Windows** alt menüsüyle erişebilirsiniz.

**Design** ekranının ortasındaki beyaz alana tıklayarak **Grid** nesnesini seçelim. **Properties** penceresinde **Background** özelliğini bulalım. **Background** özelliği ile **Grid** nesnesinin zemin rengini değiştirebiliriz. Background’un yanındaki sekmede aşağı ok’a

basarak renk seçim bölümünü açalım. Gelen pencerede bir renk seçelim, **Design** bölümü seçilen renk ile güncellenecektir.

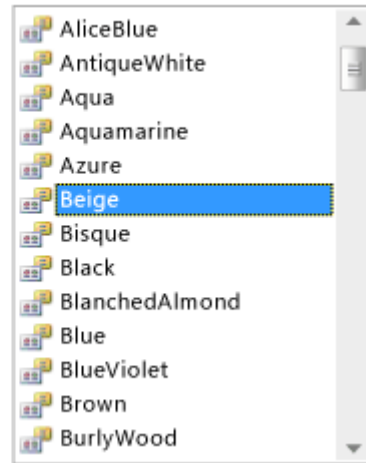


**Resim 2.13: Uygulamanın zemin rengini seçme**

**Background** özelliğini **XAML** penceresini kullanarak ta değiştirebiliriz. **XAML** ekranında **Grid** nesnesini bulalım. Grid ile başlayan bu etikette Background özelliği yoksa yazarak ekleyebiliriz. Background yazıp eşittir yazdığımızda seçebileceğimiz renkler listelenecektir. Beige rengini seçelim.

```
<Grid x:Name="LayoutRoot" Background="B">
```

```
</Grid>
```



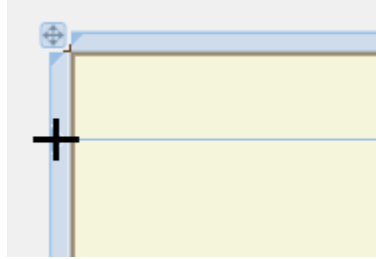
**Resim 2.14: XAML ekranından da zemin rengi belirleme**

**Grid** nesnesi HTML **table** etiketine benzer şekilde tablolara bölünebilmektedir.

Eğer seçili değilse bir defa tıklayarak **Grid** nesnesini seçelim. **Properties** penceresinden **ShowGridLines** özelliğini bulalım. Bu özelliğin sağ tarafında bulunan onay kutusunu işaretleyerek aktif duruma getirelim. XAML ekranında **ShowGridLines** özelliğine **True** değerini atayarak da yapabiliriz.

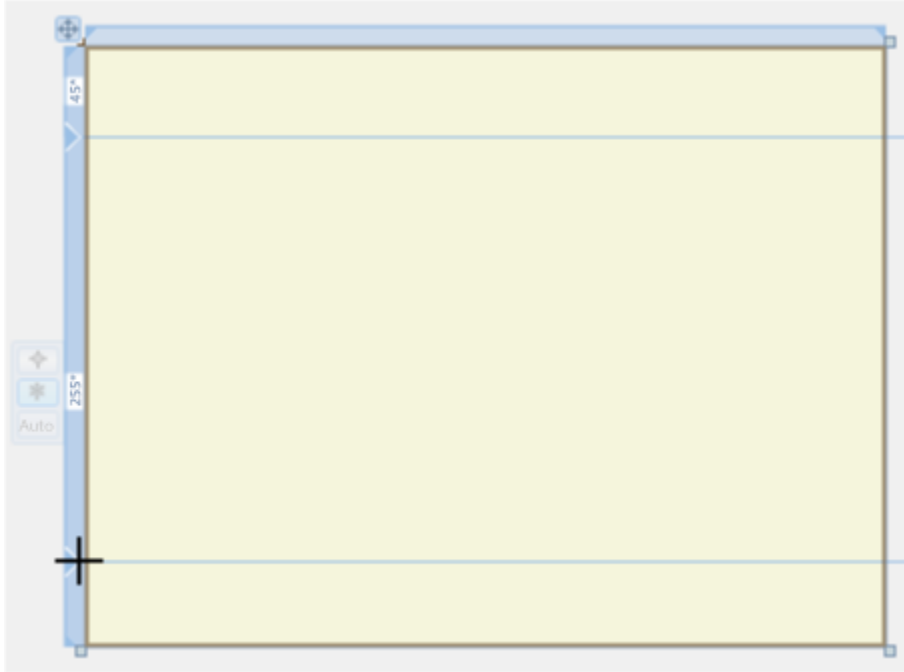
```
<Grid x:Name="LayoutRoot" Background="Beige" ShowGridLines="True">
```

Farenizi seçili durumda bulunan **Grid** nesnesinin sol veya üst bölgesinde gezdirdiğinizde ok işaretçinizin şekil değiştiğini artı simgesine dönüştüğünü göreceksiniz.



**Resim 2.6: + imlecini görmeliyiz**

**Grid**'in solunda farenin sol tuşuna tıklayarak yatay grid çizgisi ekleyebilirsiniz. Eklediğiniz grid çizgisini üstten 45 birim aşağıya sürükleyin. Alt taraftan 45 birim yukarıya da başka bir grid çizgisi ekleyiniz. Sol baştan 100 birim bırakıp dikey bir grid çizgisi ekleyiniz.



**Resim 2.7: Ekranı satırlara bölme**

Yatay ve dikey çizgileri ekledikten sonra **XAML** ekranı aşağıdaki gibi görünecektir.

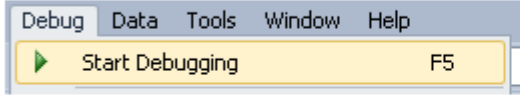


```

<Grid x:Name="LayoutRoot" Background="White" ShowGridLines="True">
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="100*" />
    <ColumnDefinition Width="300*" />
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition Height="45*" />
    <RowDefinition Height="211*" />
    <RowDefinition Height="45*" />
  </Grid.RowDefinitions>
</Grid>

```

**Resim 2.8: Uygulamanın XAML kodu**

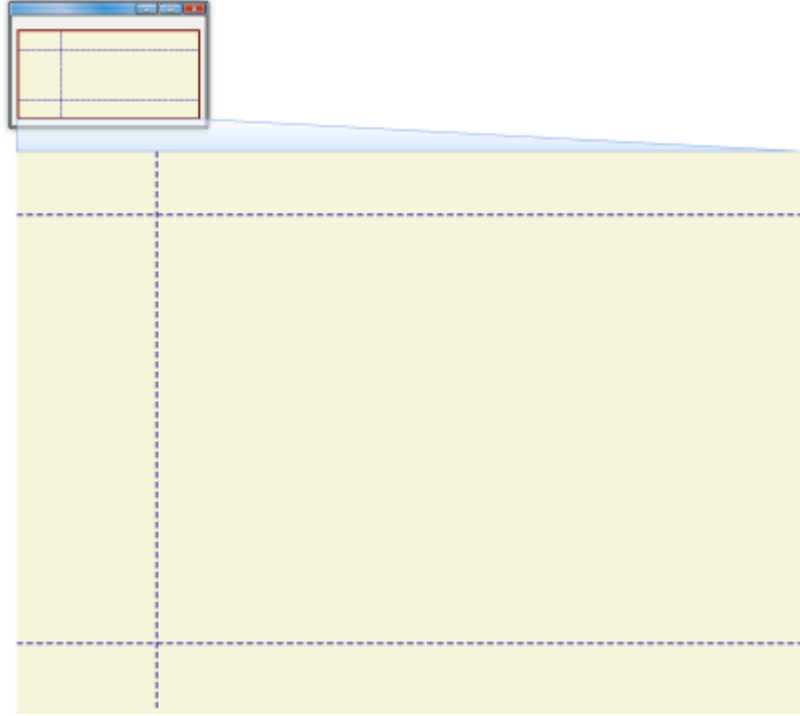
Şimdi **F5** tuşuna basarak veya menüden **Debug -> Start Debugging** seçeneğini seçerek uygulamayı çalıştıralım.

1.yol	2.yol	3.yol
		

Projeyi çalıştırmak için isterseniz Debug > start debugging menüsünü seçebilirsiniz. Yada toolbar da bulunan debug buttonuna basabilirsiniz. Yada direk F5 ile çalıştırabilirsiniz.

**Resim 2.9: Projemizi çalıştırma seçenekleri**

Varsayılan web tarayıcısı çalışacak ve uygulama ekrana gelecektir.

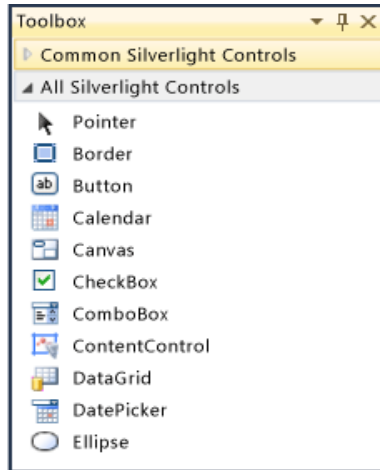


**Resim 2.10: Ekran yerleşimi**

Tarayıcıyı kapatalım ve Visual Studio'ya dönelim.

Silverlight verilerin görüntülenmesi veya kullanıcının giriş yapabilmesi için çok sayıda kontrole sahiptir.

Eğer açık değilse **ToolBox** penceresini açınız.



**Resim 2.11: ToolBox penceresi**

ToolBox'dan bir **TextBlock** nesnesini Grid'in ilk satır ilk sütununa sürükleyiniz. XAML ekranına aşağıdaki kod eklenecektir.

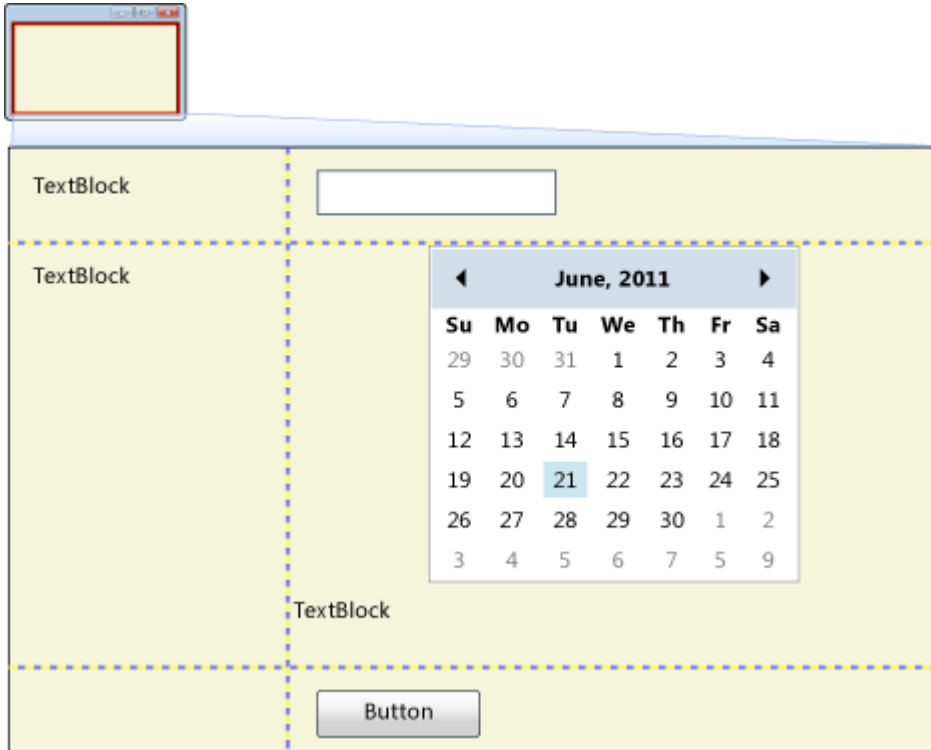
```
<TextBlock Height="23" HorizontalAlignment="Left" Margin="12,12,0,0" Name="textBlock1" VerticalAlignment="Top" />
```

Başka bir **TextBlock** nesnesini ikinci satırın birinci sütununa ekleyin.

**TextBlock** nesnesini birinci satırın ikinci sütununa sürükleyin.

İkinci satır ikinci sütununa **StackPanel** nesnesi sürükleyiniz. **StackPanel** yatay veya dikey olarak çok sayıda nesneyi gruplamayı sağlayan bir nesnedir. **StackPanel**'e sağ tıklayın **Reset All -> All**'ı seçin. Böylece **StackPanel**'in bulunduğu hücrenin tamamını kaplamasını sağladık. **ToolBox**'dan bir **Calendar** nesnesini **StackPanel**'e sürükleyiniz. **StackPanel**'e **Calendar**'ın altına gelecek şekilde bir **TextBlock** nesnesi ekleyiniz.

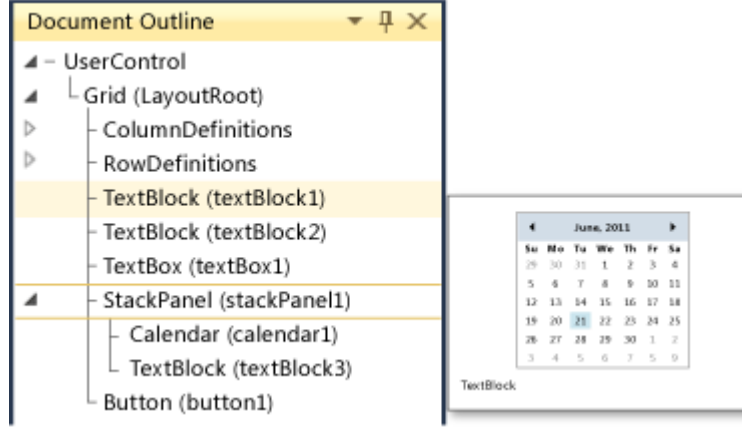
Son olarak üçüncü satır ikinci sütuna bir **Button** nesnesi ekleyiniz. Uygulamayı çalıştırınız.



**Resim 2.12: Uygulama ekranı**

Uygulamamıza çok sayıda kontrol ekledikçe bunların seçilmesi de zorlaşacaktır. Böyle durumlarda **Document Outline** penceresini kullanabiliriz.

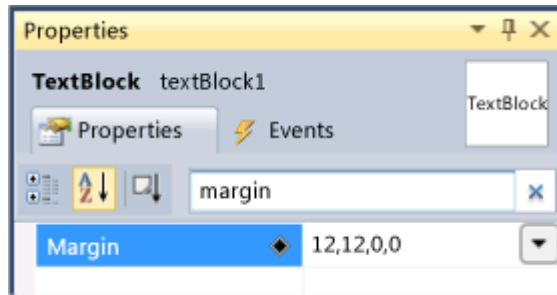
**Design** ekranında tıklayarak **Button** nesnesini seçiniz. Seçimi **XAML** ekranında **Button** kodunun herhangi bir yerine tıklayarak da yapabilirsiniz. **XAML** ekranında **TextBlock** kodunun herhangi bir yerine tıklayarak seçtiğinizde **Design** ekranında nesnenin seçildiğini ve **Properties** penceresinde nesne özelliklerinin görüntülediğini göreceksiniz.



Resim 2.13: Document Outline penceresi

Seçim işlemi için **Document Outline** penceresini de kullanabiliriz. Pencereyi açmak için **View** menüsünden **Other Windows -> Document Outline** seçeneğini seçiniz. **Document Outline** penceresinde uygulamanızda kullandığımız nesnelere hiyerarşik olarak listelenecektir. Bu listede fare imlecinizi gezdirdiğinizde listede bulunan nesnelere küçük önizleme resimlerinin açıklama olarak pencere yanında görüntülediğini göreceksiniz. Listedeki tıkladığımız kontrolün **Design** ekranında seçildiğini göreceksiniz.

Grid'in ilk satır ilk sütununda bulunan **TextBlock** nesnesini seçiniz, **Properties** penceresinde nesne özellikleri listelenecektir. **Properties** penceresini üstüne **Search (Arama)** kutusuna **margin** yazın, **Margin** özelliği karşınıza gelecektir.



Resim 2.14: Margin ayarlarını değiştirme

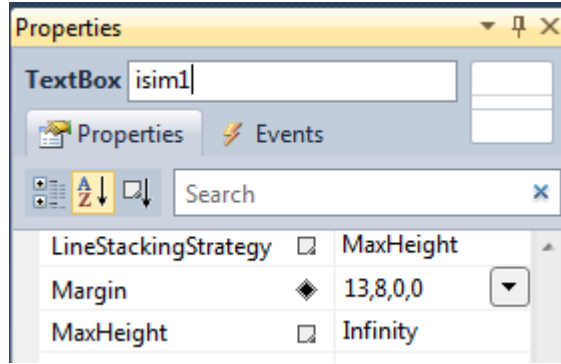
**Margin** özelliğini **10** olarak ayarlayın. **Text** özelliğini "**Adınız :**" olarak ayarlayın. **XAML** ekranında **TextBlock** nesnesi aşağıdaki şekilde görülecektir.

```
<TextBlock Margin="10" Text="Adınız :" ... />
```



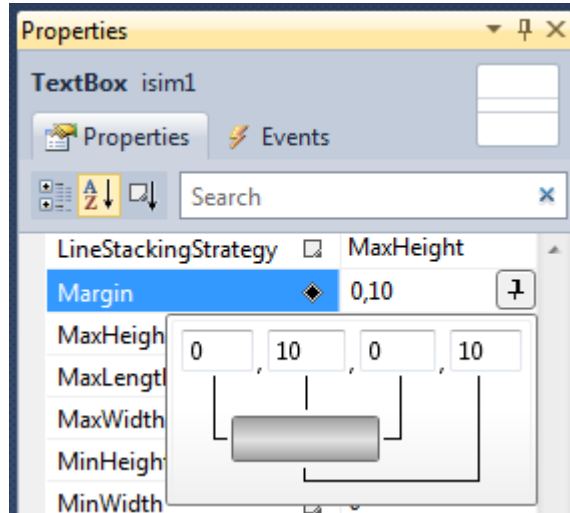
İkinci satır birinci sütundaki **TextBlock** nesnesini seçiniz. **Properties** penceresinden **Margin** değerini **10**, **Text** değerini “**Tarih :**” olarak ayarlayınız.

Birinci satırda bulunan **TextBox** nesnesini seçin ve nesnenin **Name** özelliğini **isim1** olarak ayarlayınız.



Resim 2. 15 nesnemize isim veriyoruz

**Margin** özelliğini bulun ve ikinci sütunda bulunan aşağı oka basınız. Görüntülenen ekranda sırasıyla **sol**, **üst**, **sağ** ve **alt** margin değerlerini **0**, **10**, **0**, **10** olarak ayarlayın. **Text** özelliğini “**SizinAdınız**” olarak ayarlayınız.



Resim 2.16: Margin değerlerini değiştirme

**Calendar** nesnesini seçin. **Calendar** nesnesine sağ tıklayın **Reset Layout -> All**'ı seçin. **Calendar** nesnesi yatay olarak **StackPanel** nesnesinin tamamını kaplayacaktır. **Properties** penceresinden **Name** özelliğini **cal1**, **HorizontalAligment** özelliğini **Left** olarak ayarlayınız.

**Calendar** nesnesinin altında bulunan **TextBlock** nesnesini seçin. **Properties** penceresinden **Name** özelliğini **mesaj1**, **FontSize** özelliğini **20**, **Height** özelliğini **Auto**, **Text** özelliğini “**Mesaj :**” olarak ayarlayınız.

**Button** nesnesini seçiniz. **Properties** penceresinden **Name** özelliğini **tamamButon**, **Content** özelliğini “**TAMAM**”, **Margin** özelliğini **0, 10, 0, 10** olarak ayarlayınız.

**Not:** Button nesnesinin Text özelliği yoktur, bunun yerine Content özelliği gelmiştir. Content özelliği sadece metin değil resim, kontrol vb. daha geniş bir içerik alabilir.

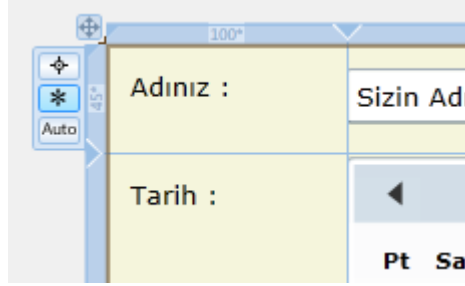
Nesnelerin özelliklerini değiştirdikten sonra **File** menüsünden **Save All** seçeneğini seçerek tüm değişiklikleri kaydediniz. Uygulamayı çalıştırınız.



**Resim 2.17: Uygulama ekranının son hâli**

Pek çok uygulamanın değişik ekran çözünürlüklerinde çalışması gerekmektedir. Bu bölümde tarayıcı yeniden boyutlandırıldığında uygulama ekranının doğru ayarlanmasını öğreneceğiz.

Grid nesnesini seçiniz ve fare imlecinizi birinci satırın solundaki mavi bölümde gezdiriniz. Küçük bir menü görüntülenecek, bu menüden **Auto** seçeneğini seçiniz. Böylece içeriğine bağlı olarak ilk satırın yüksekliğinin otomatik ayarlanmasını sağlıyoruz.



**Resim 2.18: Satır yüksekliklerini ayarlama**

Üçüncü satırı da **Auto** olarak ayarlayınız.

Uygulamayı çalıştırınız ve pencereyi tarayıcı penceresini yeniden boyutlandırınız. Birinci satır, üçüncü satır ve ilk kolonun boyutlarını koruduklarına dikkat ediniz.

Tarayıcıyı kapatınız ve Visual Studio ekranına dönünüz. Birinci **TextBlock** nesnesini seçiniz ve **Text** özelliğini “**Adınız Soyadınız**” olarak değiştiriniz. İlk kolonun otomatik olarak ayarlandığına dikkat ediniz, bunun sebebi Auto ayarıdır.

**TextBox** nesnesini seçiniz. **TextBox** seçildiğinde nesnenin sağ tarafında içi boş bir çember görünür. Farenin sol tuşu ile bu çembere sürükleyip **TextBox**'ın boyunu uzatınız.



**Resim 2.19: textBox nesnesinin boyunu uzatma**

Bütün bu değişikliklerden sonra XAML ekranı aşağıdaki şekilde görünecektir.

```
<UserControl x:Class="SilverlightApplication1.MainPage"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d"
d:DesignHeight="300" d:DesignWidth="400"
xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk">

<Grid x:Name="LayoutRoot" Background="Beige" ShowGridLines="True">
```

```

<Grid.ColumnDefinitions>
<ColumnDefinition Width="Auto" />
<ColumnDefinition Width="300*" />
</Grid.ColumnDefinitions>
<Grid.RowDefinitions>
<RowDefinition Height="Auto" />
<RowDefinition Height="211*" />
<RowDefinition Height="Auto" />
</Grid.RowDefinitions>
<TextBlock Height="23" HorizontalAlignment="Left" Margin="10" Name="textBlock1"
Text="Adınız Soyadınız :" VerticalAlignment="Top" />
<TextBlock Grid.Row="1" Height="23" HorizontalAlignment="Left" Margin="10"
Name="textBlock2" Text="Tarih :" VerticalAlignment="Top" />
<Button Content="TAMAM" Grid.Column="1" Grid.Row="2" Height="23"
HorizontalAlignment="Left" Margin="0,10" Name="tamamButton"
VerticalAlignment="Top" Width="75" Click="tamamButton_Click" />
<TextBox Grid.Column="1" Height="23" Margin="0,10,180,0" Name="isim1"
VerticalAlignment="Top" Text="Sizin Adınız" />
<StackPanel Grid.Column="1" Grid.Row="1" Name="stackPanel1">
<sdk:Calendar Name="cal1" HorizontalAlignment="Left" />
<TextBlock Height="Auto" Name="mesaj1" Text="Mesaj :" FontSize="20" />
</StackPanel>
</Grid>
</UserControl>

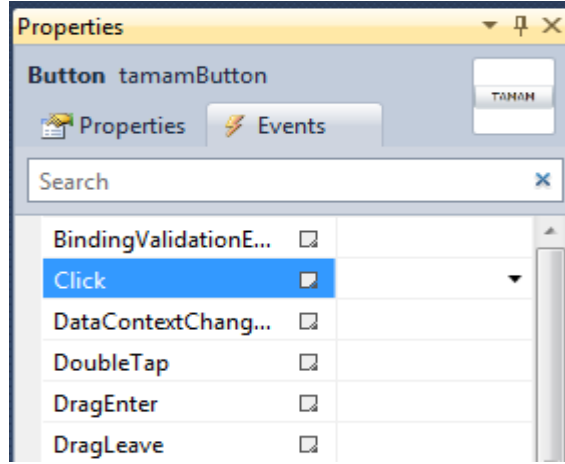
```

Uygulamayı çalıştırdınız ve tarayıcı penceresini yeniden boyutlandırınız.

Uygulamamıza eklediğimiz her kontrol kullanıcı tarafından gerçekleştirilen eylemlere tepki verir. Bunlara Event (Olay) denir. Biz event handler olarak adlandırılan metodlar oluşturarak bu olayları yakalayabiliriz. Kod yazarak erişmek istediğimiz her nesneye bir isim vermeliyiz. Bu nedenle uygulamamıza eklediğimiz her nesnemizin **Name** özelliğini kullanarak isim vermeliyiz. Bu bölümde code-behind dosyasını kullanarak olayları yakalamayı öğreneceğiz.

**Not: Visual Studio** eklenen nesnelere otomatik olarak **Name** özelliği vermektedir. **Expression Blend** ile oluşturulan uygulamalarda nesnelerin **Name** özelliği boştur. Eğer **Expression Blend** ile oluşturduğunuz uygulamaları **Visual Studio** ile açıp kod yazacaksınız mutlaka **Name** özelliklerini kendiniz vermelisiniz.

**Button** nesnesini seçelim. **Properties** penceresinin üst bölümünde bulunan **Events** sekmesine tıklayarak **Button** nesnesi ile kullanabileceğiniz olaylara ulaşın. Farenin sol tuşu ile gerçekleştirilecek olayları yakalamak için olaylar listesinden **Click** olayının sağındaki boşluğa çift tıklayınız.



Resim 2.20: Olay (events) listesi

**MainPage.xaml.cs** dosyası açılacak ve oluşturulan **tamamButton\_Click** kod bloğu görüntülenecek. Aşağıda görünen kodu bu bloğa ekleyiniz.

```
privatevoid tamamButton_Click(object sender, RoutedEventArgs e)
{
    string tarih;
    if (cal1.SelectedDate == null)
    {
        tarih = "<tarih secmelisiniz>";
    }
    else
    {
        tarih = cal1.SelectedDate.ToString();
    }
    mesaj1.Text = "Selam " + isim1.Text + "\n" + "Secilen Tarih : " + tarih;
}
```

Uygulamanın son hâlini kaydediniz ve uygulamayı çalıştırınız.

**TextBox** nesnesine adınızı yazınız, **Calendar** nesnesinden tarih seçiniz ve **TAMAM** butonuna tıklayınız.

The image shows a web application interface with a light yellow background. At the top left, there is a small, empty rectangular box. Below it, the form is divided into sections. The first section is labeled 'Adınız :' and contains a text input field with the placeholder text 'Sizin Adınız'. The second section is labeled 'Tarih :' and contains a calendar widget for the month of 'Mart 2013'. The calendar shows days of the week (Pt, Sa, Ça, Pe, Cu, Ct, Pz) and dates from 1 to 31, with the 26th highlighted. Below the calendar is a text area labeled 'Mesaj :'. At the bottom of the form, there is a button labeled 'TAMAM'.

Resim 2.21: Uygulamanın çalışması

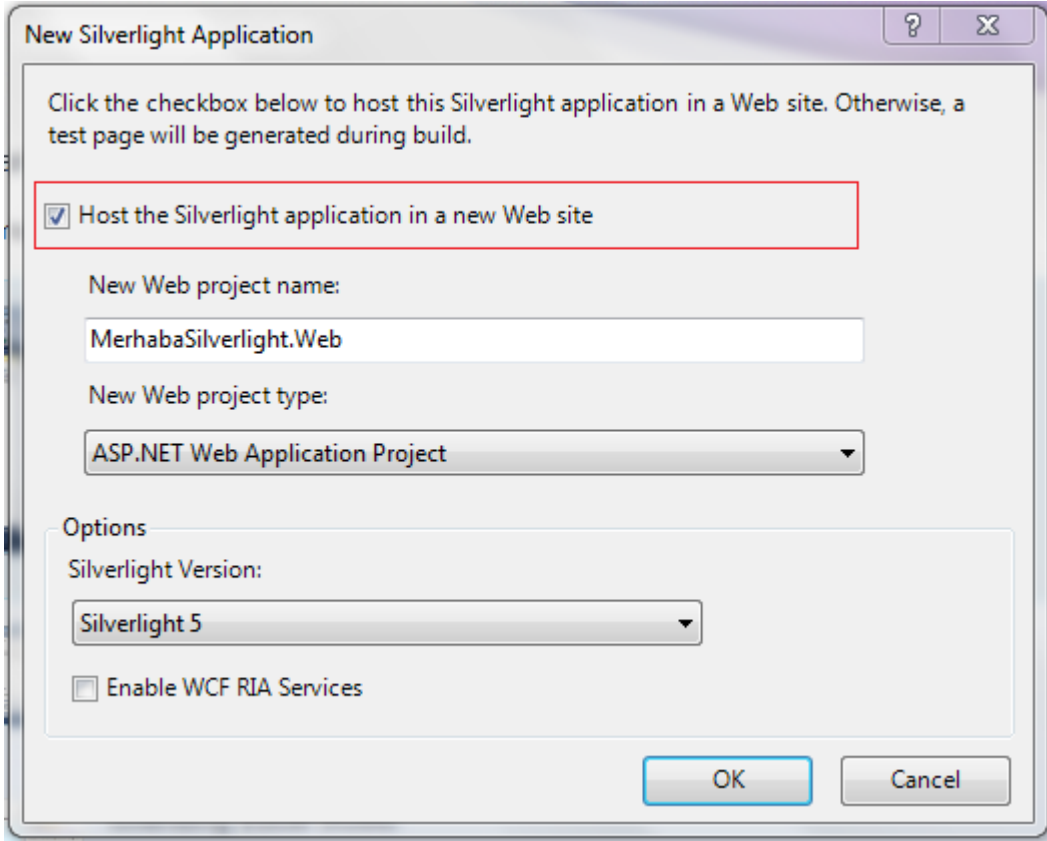
### 2.1.2. HTML Sayfası Oluşturma

Bu örnekte Silverlight ile oluşturacağınız bir **xap** dosyasını **html** sayfasına eklemeyi öğreneceğiz.

**Silverlight projesi oluşturmak için Visual Studio 2010'u açın. Karşımıza Start Page başlangıç ekranı gelecektir.**

Şimdi **New Project** bağlantısına tıklıyoruz ya da menülerden **File > New > Project**'i seçiyoruz. Ekranımıza yeni proje oluşturma penceresi gelecektir.

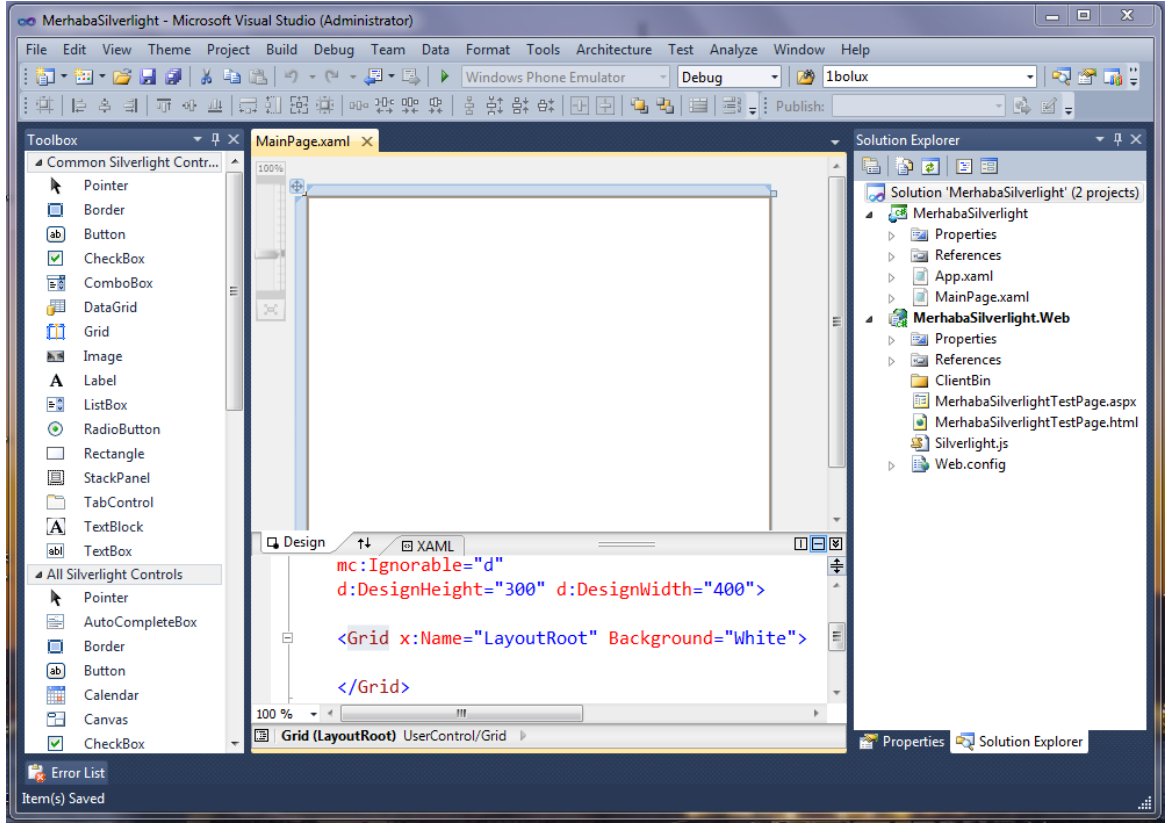
Sol taraftaki **Installed Templates** kısmından Silverlight'ı seçiyoruz. Sağ tarafta oluşturulabilecek Silverlight uygulama tipleri ekrana gelir. Biz **Silverlight Application** seçiyoruz. Alt tarafta **Name** ve **Location** kısımlarında uygulamamızın ismini ve kaydedileceği yeri belirttikten sonra **OK** butonuna tıklıyoruz.



**Resim 2.22: Onay kutusunu iptal etme**

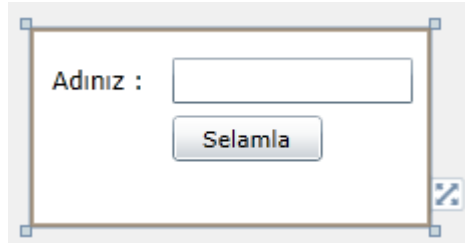
Karşımıza gelen ekranda **“Host the Silverlight Application in a new Web site”** seçeneğini iptal ediniz. Eğer bu seçenek seçili ise projenize, Silverlight objesinin çalıştırılabileceği bir test Web sitesi eklenir, biz bu örnekte web sitesini eklemek istemiyoruz.

Silverlight projemiz oluşturulacaktır.



Resim 2.23: Visual Studio ekranı

Öncelikle XAML ekranından **UserControl** nesnesinin **d:DesignHeight** değerini 100, **d:DesignWidth** değerini 200 olarak değiştiriniz. Böylece oluşturulacak uygulamanın ekran genişliğini küçültmüş olacağız.



Resim 2.33: Ekran yerleşimini ayarlama

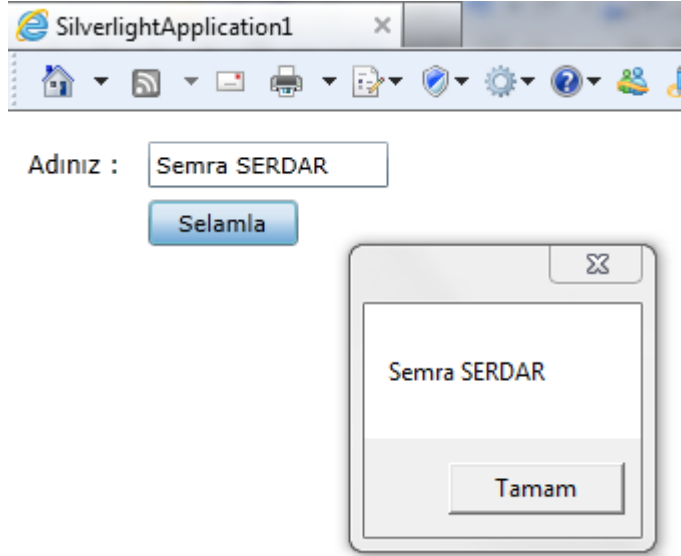
**ToolBox**'tan uygulamanıza **Label**, **TextBox** ve **Button** nesnelerini ekleyiniz ve resimdeki gibi ekran yerleşimini ayarlayın. **Label** nesnesinin **Content** özelliğini “**Adınız :** ”, **Button** nesnesinin **Content** özelliğini “**Selamla**” olarak değiştiriniz.

**Button** nesnesine çift tıklayarak kod editörüne geçiniz ve **button1\_Click** olayına aşağıda görünen kodu yazınız.



```
private void button1_Click(object sender, RoutedEventArgs e)
{
    if (!String.IsNullOrEmpty(textBox1.Text))
        MessageBox.Show(textBox1.Text);
}
```

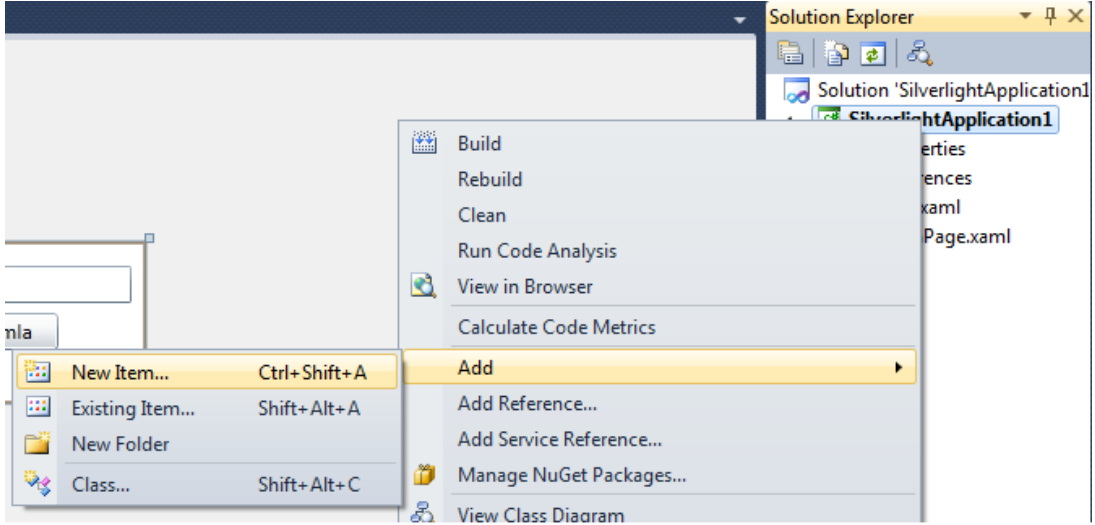
Uygulamanızı çalıştırınız ve test ediniz.



**Resim 2.24: Uygulamanın çalışma anı**

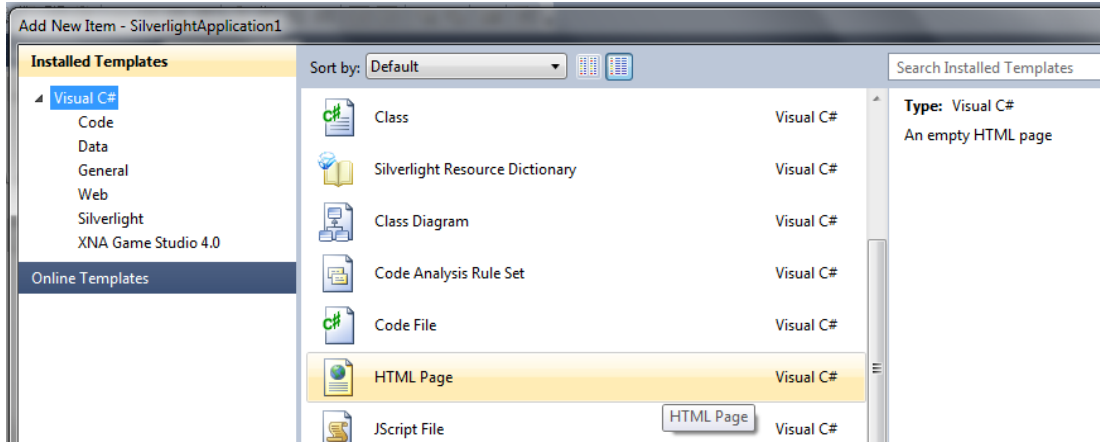
Çalışan uygulamamızı kapatıp Visual Studio ekranına dönelim.

**Solution Explorer** penceresinde uygulamamıza sağ tıklayınız, **Add > New Item...** seçeneğini seçiniz.



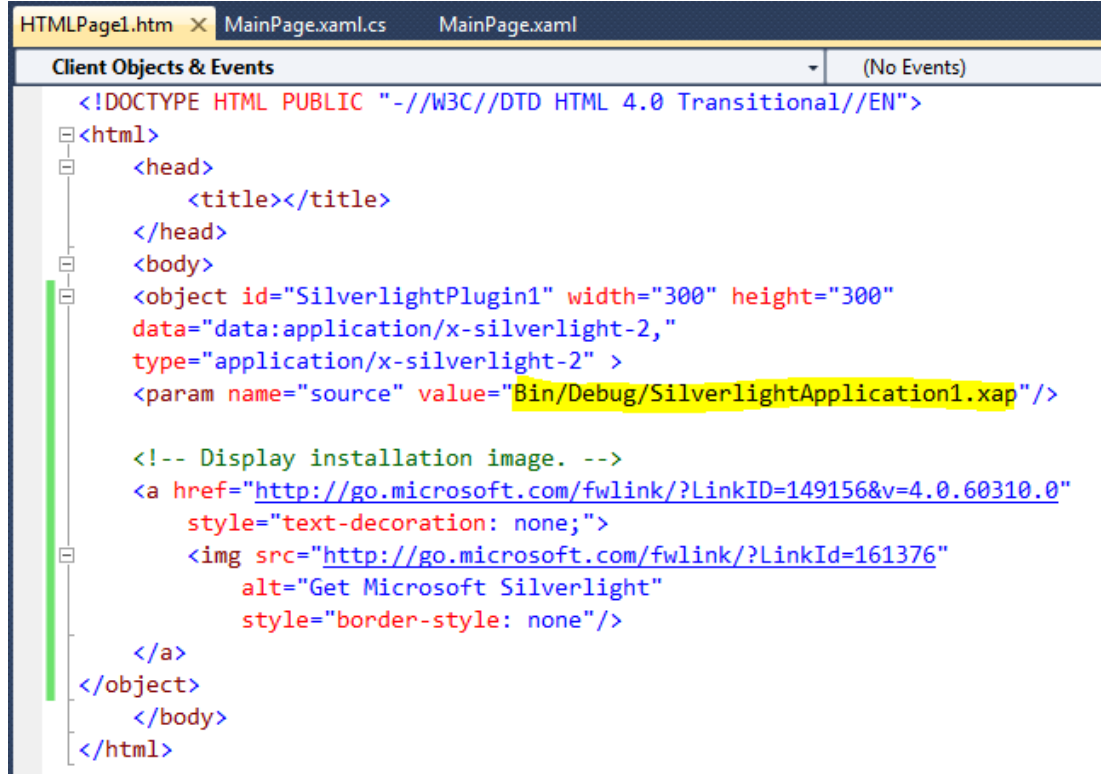
**Resim 2.25: Uygulamamıza yeni öge ekleme**

Ekranaya gelen “Add New Item” başlıklı pencereden “HTML Page” ögesini seçip Add butonuna basalım. Böylece uygulamamıza yeni bir html sayfası eklemiş olduk.



**Resim 2.26: HTML Page'i seçme**

Html sayfamıza az önce oluşturduğumuz **Silverlight** uygulamamızı eklemek için aşağıdaki kodu yazalım. Resimde sarı renkte işaretlenen bölüm **Silverlight** uygulamamızın **xap** dosyasının bulunduğu yoldur.

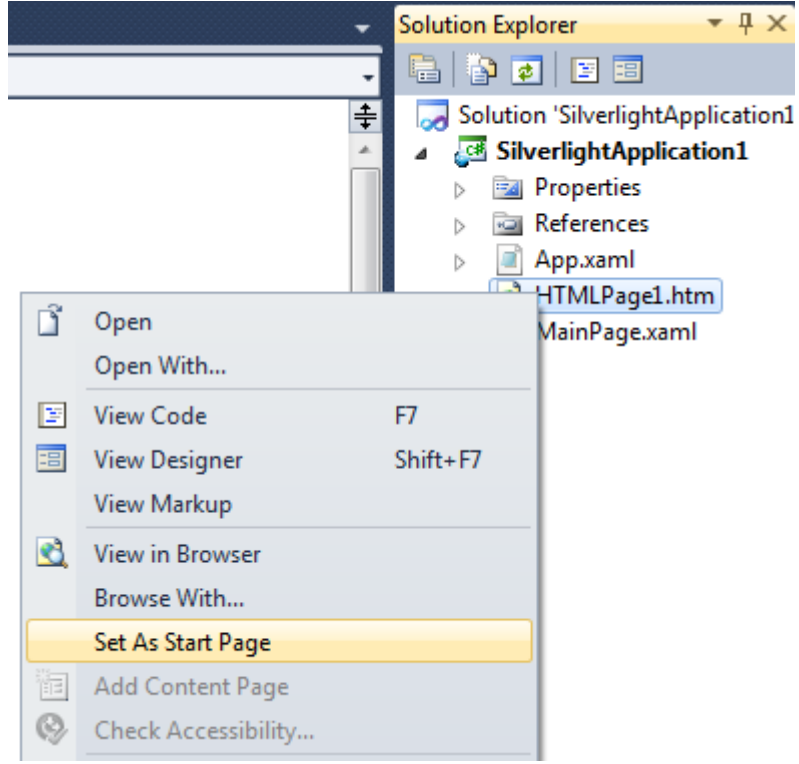


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title></title>
  </head>
  <body>
    <object id="SilverlightPlugin1" width="300" height="300"
      data="data:application/x-silverlight-2,"
      type="application/x-silverlight-2" >
      <param name="source" value="Bin/Debug/SilverlightApplication1.xap"/>

      <!-- Display installation image. -->
      <a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.60310.0"
        style="text-decoration: none;">
        
      </a>
    </object>
  </body>
</html>
```

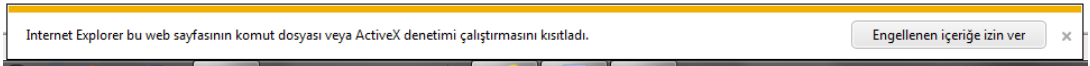
**Resim 2.27: Renkli bölümde xap dosyasının yolunu belirtme**

Projemizi çalıştırdığımızda **MainPage.xaml** ekrana gelecektir. Html sayfamızı ekrana getirmek için **Solution Explorer** penceresinde **HTMLPage1.htm** dosyasının üstüne sağ tıklayarak gelen menüden “**Set As Start Page**” seçeneğini seçelim. Böylece projeyi çalıştırmak için **F5** tuşuna bastığımızda projemiz html sayfasını çalıştıracaktır.



**Resim 2.28: Uygulamamızın başlangıç ögesini belirleme**

Uygulama çalıştığında varsayılan tarayıcınız açılacak ve **Internet Explorer** kullanıyorsanız aşağıda görünen uyarı ekrana gelecektir. Silverlight uygulamamızı görebilmek için “**Engellenen içeriğe izin ver**” butonuna basarak izin vermemiz gerekir.



**Resim 2.29: IE güvenlik uyarısı**

İzin verdikten sonra uygulamamız html sayfası içinde ekrana gelecektir.



Adınız :



**Resim 2.30: Silverlight uygulamanın görüntüsü**

Uygulamanızı test ettikten sonra kapatabilirsiniz.

## UYGULAMA FAALİYETİ

Aşağıda verilen işlem basamaklarını takip ederek konuyu pekiştiriniz.

İşlem Basamakları	Öneriler
<p>➤ Programlama Yazılımını başlatınız.</p>	<p>➤ Yazılım kısayolunu kullanabilirsiniz.</p>
<p>➤ Yeni bir Silverlight Application oluşturunuz.</p>	<p>➤ File &gt; New &gt; Project &gt; Silverlight &gt; Silverlight Application seçeneklerini kullanabilirsiniz.</p>
<p>➤ Aşağıda görünen ekranı oluşturunuz.</p> <p>Ürün <input type="text"/></p> <p>Birim Fiyat <input type="text"/></p> <p>Adet <input type="text"/></p> <p>KDV <input type="radio"/> %0 <input type="radio"/> %15 <input type="radio"/> %18</p> <p>KDV Dahil Tutar <input type="text"/></p> <p><input type="button" value="Hesapla"/></p>	<p>➤ Toolbox'ı kullanabilirsiniz. Uygulamanıza Label, ComboBox, TextBox, RadioButton ve Button nesneleri ekleyerek oluşturabilirsiniz.</p>
<p>➤</p> <p>Fare Klavye Monitör Yazıcı</p>	<p>➤ ComboBox özelliklerinden <b>Items</b> özelliğini kullanarak ekleyebilirsiniz.</p>
<p>➤ ComboBox'a aşağıda görünen kodu ekleyiniz.</p>	<p>➤ ComboBox'a çift tıklayarak kod ekranına ulaşabilirsiniz.</p>

<pre>private void comboBox1_SelectionChanged(o {     if (comboBox1.SelectedIndex == 0)         txtBirimFiyat.Text = "10";     if (comboBox1.SelectedIndex == 1)         txtBirimFiyat.Text = "8";     if (comboBox1.SelectedIndex == 2)         txtBirimFiyat.Text = "250";     if (comboBox1.SelectedIndex == 3)         txtBirimFiyat.Text = "150"; }</pre>	
<p>➤ Hesapla butonuna aşağıda görünen kodu ekleyiniz.</p> <pre>Single BirimFiyat = 0, Adet = 0, KDV = 0, Tutar = 0;     BirimFiyat           = Single.Parse(txtBirimFiyat.Text);     Adet                 = Single.Parse(txtAdet.Text); if (rB1.IsChecked == true)     KDV = 0; if (rB2.IsChecked == true)     KDV = 15; if (rB3.IsChecked == true)     KDV = 18;     Tutar=(BirimFiyat * Adet) + ((BirimFiyat*Adet)/100)*KDV;     txtTutar.Text = Tutar.ToString();</pre>	<p>➤ Hesapla butonuna çift tıklayarak kod ekranına ulaşabilirsiniz.</p>
<p>➤ Uygulamanızı çalıştırarak test ediniz.</p>	<p>➤ Klavyeden F5 tuşuna basabilirsiniz.</p>

## KONTROL LİSTESİ

Bu faaliyet kapsamında aşağıda listelenen davranışlardan kazandığınız becerileri **Evet**, kazanamadığınız becerileri **Hayır** kutucuğuna (X) işareti koyarak kendinizi değerlendiriniz.

Değerlendirme Ölçütleri	Evet	Hayır
1. Yeni bir Silverlight Application oluşturabildiniz mi?		
2. Uygulamaya nesnelere ekleyebildiniz mi?		
3. ComboBox'a kod yazabildiniz mi?		
4. Button'a kod yazabildiniz mi?		
5. Uygulamayı çalıştırıp test edebildiniz mi?		

## DEĞERLENDİRME

Değerlendirme sonunda “**Hayır**” şeklindeki cevaplarınızı bir daha gözden geçiriniz. Kendinizi yeterli görmüyorsanız öğrenme faaliyetini tekrar ediniz. Bütün cevaplarınız “**Evet**” ise “Ölçme ve Değerlendirme”ye geçiniz.



## ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Uygulamada birden fazla seçenek seçebildiğimiz nesne hangisidir?  
A) checkBox  
B) radioButton  
C) listBox  
D) comboBox
2. Bir Silverlight projesinin özellikle tasarım kısmını yapmak için hangi program kullanılır?  
A) DreamWeaver  
B) Expression Blend  
C) Photoshop  
D) Fireworks
3. C# ile geliştirilen Silverlight sayfasının tasarım dosyasının uzantısı nedir?  
A) Html  
B) Xaml  
C) Cs  
D) Xaml.cs

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

4. ( ) Silverlight **web tabanlı** bir geliştirme ortamıdır.
5. ( ) Silverlight projesinde comboBox'a eleman eklemek için **Collection** özelliği kullanılır.
6. ( ) Silverlightın kod aşamasında **VB, C#, Php** gibi programlama dili kullanılır.
7. ( ) Silverlight uygulamasında label üstünde görünmesini istediğimiz yazıyı onun **Text** özelliğine yazarız.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise “Modül Değerlendirme”ye geçiniz.

# MODÜL DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyunuz ve doğru seçeneği işaretleyiniz.

1. Bir butonun projedeki ismine ne ad verilir?  
A) Content  
B) Name  
C) Label  
D) ID
2. Label'in içeriğini bir değişkene aktarmak istersek onun hangi özelliğini kullanırız?  
A) Text  
B) Name  
C) Content  
D) Item
3. Projemizdeki nesnelerin tasarım kodlarının olduğu pencereye ne ad verilir?  
A) Design View  
B) XAML View  
C) Object Browser  
D) ToolBox
4. Bir checkBox nesnesinin seçili olup olmadığı hangi özelliği kontrol edilerek anlaşılır?  
A) Content  
B) Checked  
C) Value  
D) IsChecked

Aşağıdaki cümlelerin başında boş bırakılan parantezlere, cümlelerde verilen bilgiler doğru ise **D**, yanlış ise **Y** yazınız.

5. ( ) Silverlight'ın tasarım aşamasında **XAML** etiketleri kullanılır.
6. ( ) Uygulamada birden fazla seçenek seçebildiğimiz nesne **radioButton**'dur.
7. ( ) Bir Silverlight projesinin özellikle kod kısmını yapmak için **Expression Blend** programı kullanılır.

8. ( ) Expression Blend ile başlanmış bir silverlight projesi Visual Studio ile açılarak **tasarım** kısmına geçilebilir.
9. ( ) Silverlight uygulamaları yalnızca Internet Explorerda çalışabilir.
10. ( ) Silverlight uygulamasında butonun **Visibility** özelliğini False yaparsak ekranda buton görünmez.

## DEĞERLENDİRME

Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki modüle geçmek için öğretmeninize başvurunuz.

# CEVAP ANAHTARLARI

## ÖĞRENME FAALİYETİ-1'İN CEVAP ANAHTARI

1	C
2	D
3	B
4	A
5	C

## ÖĞRENME FAALİYETİ-2'NİN CEVAP ANAHTARI

1	A
2	B
3	D
4	Doğru
5	Yanlış
6	Yanlış
7	Yanlış

## MODÜL DEĞERLENDİRME'NİN CEVAP ANAHTARI

1	B
2	C
3	A
4	D
5	Doğru
6	Doğru
7	Yanlış
8	Yanlış
9	Yanlış
10	Yanlış

## KAYNAKÇA

- SÜZEN Ahmet Ali, **Silverlight 4**, KODLAB, İstanbul, 2011.
- TAŞDELEN Aykut, **WPF**, Pusula, İstanbul, 2010.
- [http://msdn.microsoft.com/en-us/library/cc189089\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc189089(v=vs.95).aspx) (11/04/2013)
- <http://www.csharpnedir.com/articles/read/?id=774> (20/03/2013)
- <http://www.yazgelistir.com/makale/silverlight-nedir-> (28/03/2013)
- <http://banasilverlightanlat.wordpress.com/> (19/03/2013)
- <http://www.csharpnedir.com/articles/read/?filter=&author=&cat=slv&id=859&title=Silverlight%202.0%20-%20C4%B0lk%20Uygulamam%20C4%B1z> (19/03/2013)
- <http://uzmanabi.com/icerik/windows-phone-8-uygulama-gelistirme-29437.imo> (04/04/2013)
- [http://msdn.microsoft.com/en-us/library/cc838218\(v=vs.95\).aspx](http://msdn.microsoft.com/en-us/library/cc838218(v=vs.95).aspx) (28/03/2013)
- <http://blog.hakankara.com.tr/silverlight-ve-ilk-silverlight-projemiz.html> (30/03/2013)