

Yazılım Lab. I - 2. Proje

Kapsam: Web Tabanlı Uygulamalar için Bileşen Geliştirilmesi.

Konu: Grafik Bileşen Tasarımı

İlan Tarihi : 10.11.2011

Teslim Tarihi : 10.12.2011

Açıklamalar.

1. Bahsedilen bileşeni, Microsoft .NET Class Library projesi olarak geliştirmeniz tavsiye edilir. Java gibi farklı platformlar da kullanılabilir. Yalnız projenin amacına uygun olarak web tabanlı uygulamalarda kullanılacak şekilde geliştirilmelidir.
2. Grafik türleri;
 - a. Sütun
 - b. Pasta Dilimi
3. Metodlar;
 - a. DataBind()
 - b. ExportChartImage(System.Drawing.Imaging.ImageFormat format, string filePathName)
4. Özellikler;
 - a. Grafik türü (ChartType)
 - b. Maksimum veri satır sayısı (TopNCount)
 - i. Bazı veri kaynaklarından alınan sonuçlar, grafik üzerinde göstermek için uygun olmayabilir. Örneğin, Bir alışveriş mağazasının ürün bazında satış istatistikleri grafikte gösterilmek istenmektedir. Elde edilen veri;

Ürün	Satış Miktarı (Adet)
1.Ürün A	100
2.Ürün B	110
3.Ürün C	120
4.Ürün D	130
...	
10.Ürün J	80
...	
...	
...	
55.Ürün N	9

→55 adet ürünün bir grafik üzerinde düzgün gösterilebilmesi için, grafik boyutlarının büyük olması gerekir. Bu durumda resim boyutu artar ve yüklenmesi yavaş olabilir. Ayrıca grafiği yorumlamak da zorlaşır. Diğer bir çözüm olarak; küçük boyutlarda bir resim üzerinde gösterme durumunda da grafiği yorumlamak mümkün değildir.

→Benzer sonuçlar için yazılımcı TopNCount özelliğini kullanırsa;

```
myChart.TopNCount = 10;
```

Grafik üzerinde gösterilecek ürünler sırasıyla,

Ürün D, Ürün C, Ürün B, Ürün A, Ürün J ve Diğerleri olacaktır. Diğerleri, 10. Üründen sonraki tüm satırların toplamıdır.

5. Kullanım örneği;

```
//Nesnenin oluşturulması
YazLabChart myChart = new YazLabChart();

//Verikaynağının atanması
//myDataSource → DataSet, DataTable, ArrayList ve Array olarak verilebilecektir.

myChart.DataSource = myDataSource;

//Grafik türünün belirlenmesi;
myChart.ChartType = YazLabChart.ChartType.Pie;
```



```
myChart.ChartType = YazLabChart.ChartType.Bar;

//Maksimum veri satır sayısı
myChart.TopNCount = 10;

//Ekranaya Yazdırmak için;
myChart.DataBind();

//Kaydetmek için;
myChart.ExportChartImage(System.Drawing.Imaging.ImageFormat.Png, "sample");

//Png uzantılı Dosya Sunucu üzerinde oluşturulacak ve daha sonra download
edilebilecek. Örnek kod aşağıda verilmiştir.

System.IO.FileStream fs =
System.IO.File.Open(Server.MapPath("sample.png"), System.IO.FileMode.Open);
byte[] btFile = new byte[fs.Length];
fs.Read(btFile, 0, Convert.ToInt32(fs.Length));
fs.Close();
Response.AddHeader("Content-disposition", "attachment; filename="sample.png");
Response.ContentType = "application/octet-stream";
Response.BinaryWrite(btFile);
Response.End();
```

ÖNEMLİ NOTLAR:

1. Veri kaynağı tipi kullanıcıya bırakılacaktır. DataSet, DataTable, List veya Array yapılarını desteklenmelidir.
2. Hiçbir şekilde, kapalı ya da açık kaynak kodlu hazır kütüphane ve bileşenler **kullanılamaz**.
3. Proje olarak sizden bir kütüphane beklenmektedir. Diğer bir ifade ile geliştireceğiniz kütüphane, farklı web uygulamalarında kullanılacaktır.
4. Proje sunumunda kütüphane kodları ve bu kütüphanenin kullanıldığı örnek bir web uygulaması ile hazır bulunmalısınız.
5. Projeden beklenenlerin dışında bağımsız olarak çalışan, demo, test, deneme vb. yaklaşımlarla bir şeyler karalayıp proje olarak **teslim etmeyiniz**.
6. Optimum kod yazınız. Kodlamada profesyonel olmaya çalışınız. Baştan sona kadar en azından genel kodlama kriterlerine uymaya çalışınız.
7. Proje 2 şer kişilik gruplar halinde sunulacaktır. Proje gruplandırmasında, 1. Proje grupları geçerli olacaktır.
8. Herhangi bir şekilde kopya tespit edilirse, "beraber çalıştık", "yardımlaştık" vb. gibi açıklamalar dikkate alınmayacak ve ilgili kişilerin projeleri "Teslim Etmedi" olarak değerlendirilecektir.
9. Proje ile ilgili sorularınızı, syazici@kocaeli.edu.tr adresinden sorabilirsiniz.

Başarılar...

S.YAZICI