

Teknik Çözüm Raporu

Şükrücan Tıǵlıoǵlu

1/5/2024

Contents

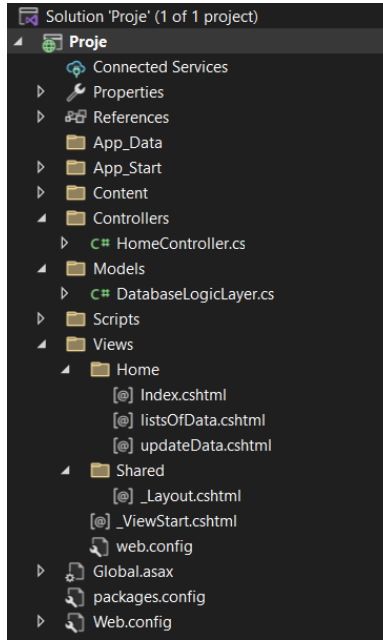
Yazılım Konfigürasyonu	3
Yazılımda kullanılan paketler	3
Veri tabanı yönetim sistemi yapılandırılması	4
Uygulamada kullanılan yazılım	4
Models – View - Controller	4
Web.config	4
_layout.cshtml	5
HomeController	5
Index	5
Upload	5
ListsOfData	5
UpdateData	6
DownloadExcel	6
Database Logic Layer	7
GetTableNames	7
GetTableCount	7
IsTableExists	7
CreateTableWithColumnCount	7
InsertDataToDatabase	7
GetTableData	7
GetItemById	7
UpdateItemById	7
DeleteItemById	7
DownloadTableDataAsExcel	7

Yazılım Konfigürasyonu

Proje kapsamında kullanılan uygulama ve programlama dilleriyle ilgili bilgiler Tablo 1’de gösterilmiş olup projenin dosya yapısı Şekil 1’de verilmiştir.

Tablo 1 Yazılım Konfigürasyon Bilgileri

Uygulama Türü	Uygulama İsmi	Uygulama Versiyonu
İşletim sistemi	Windows	Windows 10
IDE	Visual Studio	2022
Veri Tabanı Yönetim sistemi	SQL Server Management Studio	SQL Server Management Studio 20
ASP.NET Web Application	.Net Framework	.Net Framework 4.7.2



Şekil 1 Projenin Dosya Yapısı

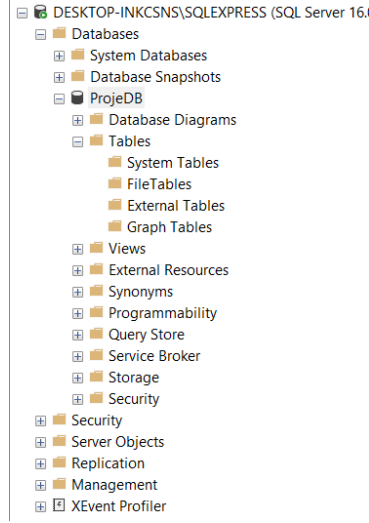
Yazılımda kullanılan paketler

ASP.NET de kullanılan paketler ve sürümleri Tablo 2’de gösterilmiştir.

Kütüphane Adı	Sürüm
Bootstrap	5.3.3
ClosedXML	0.102.2
System.Data.SqlClient	4.8.6
EPPlus	7.1.2

Veri tabanı yönetim sistemi yapılandırılması

Uygulamamızın kullanacağı veri tabanı yapılandırılması için ProjeDB adında bir database oluşturuldu ve tablo kısmı boş bırakıldı. Tablo kısmı uygulamamıza eklenen excel dosyasına göre dinamik oluşturulmaktadır. Şekil 2’de veri tabanına ait gösterilmiştir.



Şekil 2 Veri tabanı yönetim sistemine ait görüntü

Uygulamada kullanılan yazılım

Models – View - Controller

Model-View-Controller (MVC) mimarisine dayalı bir web uygulama geliştirme çatısıdır. Bu çatı, web uygulamalarını organize etmek, modülerleştirmek ve bakımını kolaylaştırmak için kullanılır. Model verileri işler, Controller istekleri işler ve View sonuçları kullanıcıya sunar. Bu üç bileşen birbirinden bağımsızdır ve her biri belirli bir işlevi yerine getirir. MVC mimarisi, kodun daha organize, bakımı daha kolay ve yeniden kullanılabilir olmasını sağlar. Bu sayede, büyük ölçekli web uygulamalarının geliştirilmesi daha verimli ve yönetilebilir hale gelir.

Web.config

Bu dosyamızda veri tabanı bağlantımızın ayarlamaları yapılmaktadır. Şekil 3’te gösterilen kısım veri tabanı bilgilerinize göre değiştirilmesi gerekmektedir.

```
<connectionStrings>
  <add name="MyConnectionString"
        connectionString="data source=DESKTOP-INKCSNS\SQLEXPRESS;
        initial catalog=ProjeDB;
        integrated security=True;"
        providerName="System.Data.SqlClient" />
</connectionStrings>
```

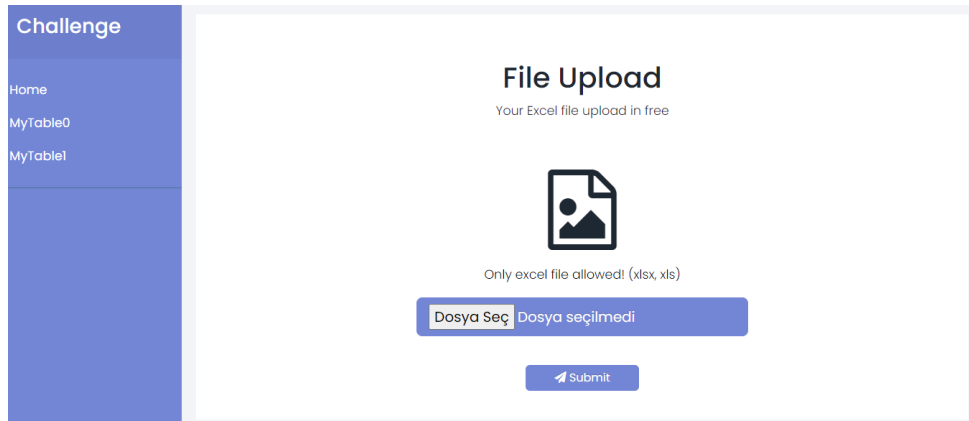
_Layout.cshtml

Bu dosya sitemiz için statik şablon görevini görmektedir. Sol menüyü ve dinamik alanımızı barındırmaktadır.

HomeController

Index

Uygulamanın ilk başladığı sınıftır. İçerisinde bulunan Index fonksiyonunu çağırılmaktadır. Index fonksiyonu bize Index.cshtml sayfasını geri dönmektedir. Bu şekilde kullanıcıya html dosyası gösterilmektedir. Şekil 4’de Index sayfasına ait görsel gösterilmektedir. Bu sayfada kullanıcıdan bir adet excel formatından dosya yüklenmesi beklenmektedir. Yüklenen excel dosyası Home controller içerisinde Upload fonksiyonuna yönlendirilmektedir.



Şekil 4 Index sayfası

Upload

Upload fonksiyonu kendisine **HttpPostedFileBase** türünde gelen excel dosyasını okuyarak **Models/DatabaseLogicLayer** sınıfında bulunan **CreateTableWithColumnCount** sınıfı ile veri tabanında yeni bir tablo oluşturmakta ve oluşturduğu tabloya dinamik olarak excel de bulunan sütun kadar sütun oluşturup verileri veri tabanına kaydetmektedir. Kaydetme işlemi bittikten sonra veri tabanında bulunan tüm tablolar **GetTableNames** fonksiyonu ile taranmakta ve tarama sonucunda bulunan veri tabanları sol menüde gösterilmektedir. Menüden seçilen tablo **listsOfData** fonksiyonu ile gösterilmektedir.

ListsOfData

Bu fonksiyonda kendisine parametre olarak gelen tablo isminde bulunan verileri **GetTableData** fonksiyonu yardımı ile çekmekte ve **listsOfData.cshtml** dosyasında bu verileri listelemektedir. Listeleme esnasında ilgili satırın yanında güncelleme ve silme butonları yer almaktadır. Bu butonlar yardımı ile veriler güncellenebilmekte ve silinebilmektedir. Silme işlemini **deleteData**

fonksiyonu yapmaktadır. Kendisine gönderilen tablo adı ve index numarasına göre ilgili tabloda yer alan veriyi silmektedir. Şekil 5'te lists of data fonksiyonunun kullanıcı arayüz kısmı gösterilmiştir.

Challenge	MyTable2			
Home	Excel olarak kaydet			
MyTable0	#	Column 1	Column 2	
MyTable1	1	harvest (noun)	hasat, ekin biçme	Düzenle Sil
MyTable2	2	thresh (verb)	harman dövmek	Düzenle Sil
	3	tool (noun)	âlet, takım	Düzenle Sil
	4	stone (noun)	taş	Düzenle Sil
	5	earn (verb)	para kazanmak	Düzenle Sil
	6	trade (noun)	ticaret, alışveriş	Düzenle Sil
	7	worthwhile (adjective)	değer, değer	Düzenle Sil
	8	spent (adjective)	harcanmış, sarfedilmiş, kullanılmış	Düzenle Sil

Şekil 5 listsOfdata.cshtml görüntüsü

UpdateData

Update Data fonksiyonu hem get hemde post şeklinde yer almaktadır bunun nedeni şu şekildedir; get isteği yaptığımızda fonksiyonun bize view klasöründe kullanıcının görüntüleyeceği bir kullanıcı arayüz döndürmesi için yazılmıştır. İstenilen parametreler get metodu ile alınarak kullanıcıya güncelleyeceği verileri listelemekte ve kullanıcı güncelleyeceği veri üzerinde düzenleme yaptıktan sonra post metodu ile ilgili post fonksiyonuna bu veriler iletilmekte ve veriler **UpdateItemById** fonksiyonu ile güncellenmektedir. Update Data kullanıcı arayüzü Şekil 6'da gösterilmiştir.

Challenge	updateData
Home	tool (noun)
MyTable0	âlet, takım
MyTable1	Gönder
MyTable2	

Şekil 6 Update data cshtml

DownloadExcel

Bu fonksiyon listsofdata.cshtml de listelenen verileri veri tabanından çekerek excel formatına dönüştürüp indirilmesini sağlar.

Database Logic Layer

Veri tabanı ile olan işlerimizin yürütüldüğü sınıftır. Bu sınıfta modal katmanımızın işleri üstlenilmektedir. Veri tabanından aldığı veriyi controller katmanına iletmektedir. Bu sınıf singleton yapısında tasarlanarak program boyunca sadece bir tane sınıf nesnesi oluşturulması sağlanmıştır bu sayede örnek oluşturulduğu yerlerde aynı değer ve verileri döndürmesi sağlanılmıştır.

GetTableNames

Bu fonksiyon veri tabanında bulunan tabloların isimlerini çekmekte ve bunları List generic türünde string tipinde geriye döndürmektedir.

GetTableCount

Bu fonksiyon veri tabanında bulunan tabloların sayısını int türünde geriye dönmektedir.

IsTableExists

Bu fonksiyon kendisine gelen parametreye göre veri tabanında o isimde tablo varmı kontrol işlemini yapmakta ve geriye boolean türünde veri döndürmektedir.

CreateTableWithColumnCount

Bu fonksiyon kendisine gelen parametreye göre sütun yaratarak veri tabanında tablo oluşturmaktadır.

InsertDataToDatabase

Oluşturulan tabloya veri ekleme işlemi yapmaktadır.

GetTableData

Kendisine gelen parametre ismine göre ilgili tablodaki verileri geriye döndürmektedir.

GetItemById

Kendisine gelen tablo adı ve id'ye göre ilgili tabloda ilgili id'ye sahip veri bularak geriye döndürmektedir.

UpdateItemById

Kendisine gelen parametrelere göre veri tabanında ki ilgili veriyi güncellemektedir.

DeleteItemById

Kendisine gelen parametrelere göre veri tabanında ki ilgili veri silmektedir.

DownloadTableDataAsExcel

Kendisine gelen parametreye göre ilgili tabloyu excele dönüştürmektedir.