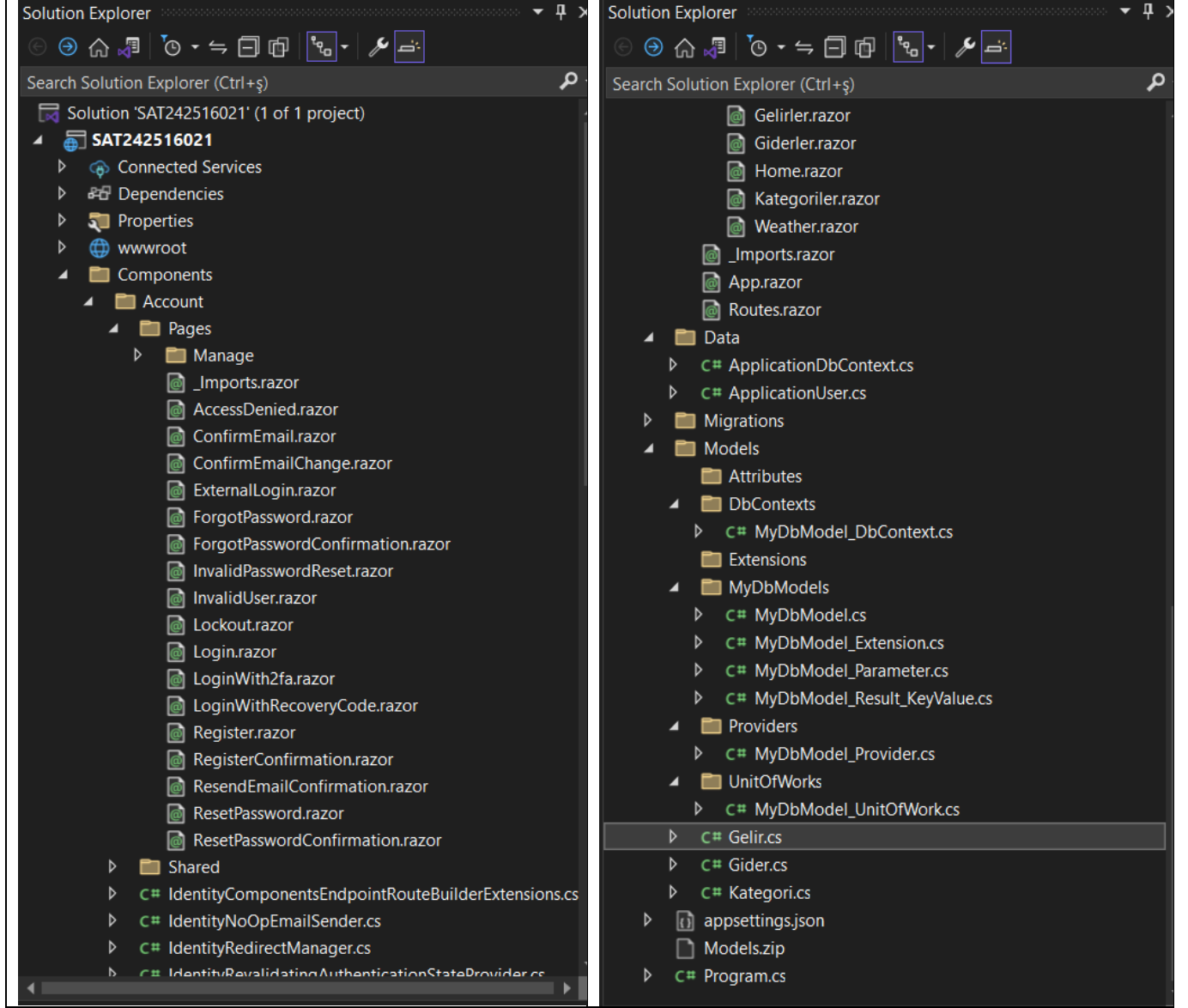


Proje Adı	Gelir Gider Takip Sistemi
Proje Konusu	Kullanıcının gelir ve giderlerini takip edip rapor oluşturabileceği bir sistem
Proje Çözüm Adı	SAT242516021

Proje ‘Çözüm Gezgini (Solution Explorer)’ penceresinin (alt klasörler ve dosyalar dahil olacak şekilde) ekran görüntüsünü resim olarak ekleyiniz



Projede oluşturulan arabirimlerin (interface) C# kodlarını yazınız.

<pre>public interface IMyDbModel {     string Message { get; set; }     IMyDbModel_Parameter Parameters { get; set; }     IDictionary&lt;object, object&gt; OrderByItems { get; set; } }</pre>
<pre>public interface IMyDbModel&lt;T&gt; : IMyDbModel where T : class, new() {     IEnumerable&lt;T&gt; Items { get; set; } }</pre>

```

public interface IMyDbModel_Parameter
{
    string OrderBy { get; set; }
    int PageNumber { get; set; }
    int PageSize { get; set; }
    int TotalPageCount { get; }
    int TotalRecordCount { get; set; }
    IDictionary<string, object> Params { get; set; }
    IDictionary<string, string> Where { get; set; }
}

public interface IMyDbModel_Result_KeyValue
{
    string Key { get; set; }
    int Value { get; set; }
}

public interface IMyDbModel_Provider
{
    ValueTask<IMyDbModel<TResult>> Execute<TResult>(
        IMyDbModel<TResult> myResultModel,
        string spName = "",
        bool isPagination = true)
        where TResult : class, new();

    ValueTask<IMyDbModel<TResult>> Execute<TResult>(
        string spName = "",
        params (string Key, object Value)[] parameters)
        where TResult : class, new();

    ValueTask<IEnumerable<TResult>> GetItems<TResult>(
        string spName = "",
        params (string Key, object Value)[] parameters)
        where TResult : class, new();

    ValueTask<IEnumerable<TResult>> SetItems<TResult>(
        string spName = "",
        params (string Key, object Value)[] parameters)
        where TResult : class, new();
}

public interface IMyDbModel_UnitOfWork
{
    Task Execute<T>(
        IMyDbModel<T> myDbModel,
        string spName = "",
        bool isPagination = true)
        where T : class, new();
}

```

**Projede oluşturulan sınıfların (class) C# kodlarını yazınız.**

Uyarı : Sınıf içerisinde metotların sadece isim ve parametreleri yazılacak.

Örnek : public async Task MethodAsync(parameters...){ return null; }

```

public class MyDbModel<T> : IMyDbModel<T> where T : class, new()
{
    public MyDbModel() { }

    public MyDbModel(int pageNumber, int pageSize, string orderBy) { }

    public string Message { get; set; }

    public IMyDbModel_Parameter Parameters { get; set; }

    public IDictionary<object, object> OrderByItems { get; set; }

    public IEnumerable<T> Items { get; set; }
}

public class MyDbModel_Parameter : IMyDbModel_Parameter
{
    public static MyDbModel_Parameter Create(int pageNumber, int pageSize, string
orderBy)
    {
        return null;
    }

    public MyDbModel_Parameter(int pageNumber, int pageSize, string orderBy) { }

    public int PageNumber { get; set; }

    public int PageSize { get; set; }

    public int TotalRecordCount { get; set; }

    public int TotalPageCount { get; }

    public string OrderBy { get; set; }

    public IDictionary<string, object> Params { get; set; }

    public IDictionary<string, string> Where { get; set; }
}

public class MyDbModel_Result_KeyValue : IMyDbModel_Result_KeyValue
{
    public string Key { get; set; }

    public int Value { get; set; }
}

public static class MyDbModel_Extension
{
    public static IDictionary<object, object> GetOrderByItems<E>(MyDbModel<E>
model)
    {
        where E : class, new()
        {
            return null;
        }
    }
}

public MyDbModel_Provider(IMyDbModel_UnitOfWork uow) { }

public async ValueTask<IMyDbModel<TResult>> Execute<TResult>(
    IMyDbModel<TResult> myResultModel,
    string spName = "",
    bool isPagination = true)
    where TResult : class, new()
{
    return null;
}

```

```
public MyDbModel_UnitOfWork(TDbContext context) { }
```

```
    public async Task Execute<T>(  
        IMyDbModel<T> myDbModel,  
        string spName = "",  
        bool isPagination = true)  
        where T : class, new()  
    {  
        return;  
    }
```

```
public class MyDbModel_DbContext : DbContext
```

```
{  
    public MyDbModel_DbContext(DbContextOptions<MyDbModel_DbContext> options)  
        : base(options)  
    {  
    }  
}
```

```
public static class Extensions_DataTable
```

```
{  
    public static IEnumerable<T> DataTableToList<T>(DataTable table)  
    {  
        return null;  
    }  
  
    public static T GetObject<T>(DataRow row, List<string> columnNames)  
    {  
        return default;  
    }  
}
```

```
public static class Extensions_Enum
```

```
{  
    public static string GetDescription(Enum enumValue)  
    {  
        return null;  
    }  
}
```

```
public static class Extensions_Json
```

```
{  
    public static string ToJson(object value)  
    {  
        return null;  
    }  
  
    public static T FromJson<T>(string json)  
    {  
        return default;  
    }  
}
```

```

public static class Extensions_SqlParameter
{
    public static SqlParameter ToSqlParameter_Table_Type_Dictionary<TKey,
TValue>(
        IDictionary<TKey, TValue> dictionary,
        string parameterName,
        string parameterTypeName = "",
        int length = 0)
    {
        return null;
    }

    public static SqlParameter ToSqlParameter_Data_Type<T>(
        T value,
        string parameterName)
    {
        return null;
    }
}

```

```

public class ColorAttribute : Attribute
{
    public string Value { get; set; }

    public ColorAttribute(string value)
    {
        Value = value;
    }
}

```

```

public class TitleAttribute : Attribute
{
    public string Value { get; set; }

    public TitleAttribute(string value)
    {
        Value = value;
    }
}

```

**appsettings.json dosyasında “DefaultConnection” ifadesini projenize göre yazınız**

```

"DefaultConnection": "Server=localhost,1445;Database=SAT242516021;User
Id=sa;Password=0o_454545;TrustServerCertificate=True;Encrypt=False;

```

**Program.cs dosyasında, gerekli servis kayıtlarını yapan C# kodlarını yazınız.**

```

builder.Services.AddDbContext<MyDbModel_DbContext>(options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection")))
;
builder.Services.AddScoped<IMyDbModel_UnitOfWork, MyDbModel_UnitOfWork<MyDbModel_DbCon
text>>();
builder.Services.AddScoped(typeof(IMyDbModel<>), typeof(MyDbModel<>));
builder.Services.AddScoped<IMyDbModel_Provider, MyDbModel_Provider>();

```

**App.razor bileşeninde, gerekli C# kodlarını yazınız**

```

<Router AppAssembly="@typeof(App).Assembly">
  <Found Context="routeData">
    <RouteView RouteData="@routeData" DefaultLayout="@typeof(MainLayout)" />
  </Found>
  <NotFound>
    <LayoutView Layout="@typeof(MainLayout)">
      <p>Sayfa bulunamadı.</p>
    </LayoutView>
  </NotFound>
</Router>

@code {
}

```

**Projenizde oluşturmuş olduğunuz bileşenlerin tasarım ve C# kodlarını yazınız (Blazor Component)**

NOT: daha fazla bileşen için, aşağıdaki tabloyu çoğaltınız.

Bileşen Adı	Gelirler.razor
Tasarım Kodları	<pre> @page "/gelirler" @using SAT242516021.Models @using Providers @inject IMyDbModel_Provider provider  &lt;h3&gt;Gelir Listesi&lt;/h3&gt;  @if (_model == null) {   &lt;p&gt;Yükleniyor...&lt;/p&gt; } else {   &lt;table class="table table-striped"&gt;     &lt;thead&gt;       &lt;tr&gt;         &lt;th&gt;Id&lt;/th&gt;         &lt;th&gt;Açıklama&lt;/th&gt;         &lt;th&gt;Tutar&lt;/th&gt;         &lt;th&gt;Tarih&lt;/th&gt;         &lt;th&gt;Kategori&lt;/th&gt;       &lt;/tr&gt;     &lt;/thead&gt;     &lt;tbody&gt;       @foreach (var item in _model.Items)       {         &lt;tr&gt;           &lt;td&gt;@item.Id&lt;/td&gt;           &lt;td&gt;@item.Aciklama&lt;/td&gt;           &lt;td&gt;@item.Tutar&lt;/td&gt;           &lt;td&gt;@item.Tarih.ToShortDateString()&lt;/td&gt;           &lt;td&gt;@item.KategoriAdi&lt;/td&gt;         &lt;/tr&gt;       }     &lt;/tbody&gt;   &lt;/table&gt; } </pre>

C# Kodları	<pre>@code {     private IMyDbModel&lt;Gelir&gt;? _model;      protected override async Task OnInitializedAsync()     {         _model = await provider.Execute(             new MyDbModels.MyDbModel&lt;Gelir&gt;(1, 10, "Id"),             "sp_Gelirler"         );     } }</pre>		
Ekran Görüntüleri			
UYARI: Resim boyutlarının çok büyük olmamasına dikkat ediniz			