



# Proje Adı: Hata ve Talep Takip Sistemi



## Proje Amacı:

Bu projede yazılım geliştirme süreçlerinde karşılaşılan hata, talep ve soru gibi durumların takibini yapabileceğimiz basit bir ticket (bilet) sistemi geliştirilecektir. Kullanıcılar yeni ticket'lar açabilecek, geliştiriciler bu ticket'lara yanıt verebilecek ve admin kullanıcılar sistemi yönetecektir.

---



## Proje Özellikleri:

### 1. Kullanıcı Girişi ve Roller:

- ~~Sisteme kayıt olunabilecek ve giriş yapılabilir.~~
- ~~Üç farklı kullanıcı tipi olacaktır:~~
  - ✓ **Kullanıcı:** Ticket oluşturur ve takip eder.
  - ✓ **Developer (Yazılımcı):** Ticket'lara çözüm yazar ve durum günceller.
  - ✓ **Admin:** Sistemi yönetir, kullanıcı ve ticket kontrollerini sağlar.

---

## 2. Ticket (Bilet) Yönetimi:

- Yeni ticket oluşturulabilir:
    - Başlık
    - Açıklama
    - Önem derecesi (Düşük / Orta / Yüksek)
    - Tür: Hata / Talep / Soru
    - Durum: Açık,Kapatıldı
  - Developer'lar ticket'a atanabilir ve durumunu güncelleyebilir.
  - Kullanıcı kendi ticket'larını görebilir, yorumları takip edebilir.
-

## Veritabanı Tasarımı

- Identity'nin rol yönetimini araştır(Tablolar otomatik oluşacak Asp ile olanlar)

```
Tickets {
    int Id PK
    string Title
    string Description
    TicketType Type Enum
    TicketPriority Priority Enum
    TicketStatus Status Enum
    DateTime CreatedAt
    string CreatedByUserId FK "AppUser.Id"
    string? AssignedToUserId FK "AppUser.Id"
    int? ProjectId;
}
```

```
TicketComments {
    int Id PK
    int TicketId FK
    string CommentText
    DateTime CommentedAt

    DateTime UpdatedAt

    string CommentedByUserId FK "AppUser.Id"
}
```

```
TicketAttachments {
    int Id PK
    int TicketId FK
    string FileName
    string FilePath
    string ContentType
    DateTime UploadedAt
    string UploadedByUserId FK "AppUser.Id"
}
```

Projects

```
{  
    int Id;  
    string Name;  
    string Description;  
    DateTime CreatedAt;  
    string CreatedByUserId;  
}
```

AppUsers {

```
    string Id PK "IdentityUser.Id"  
    string Name  
    string Email
```

}

AppUserRoles {

```
    string UserId PK,FK "AppUser.Id"  
    string RoleId PK,FK "IdentityRole.Id"
```

}

IdentityRoles {

```
    string Id PK  
    string Name
```

}

## Detaylı Tablo Açıklamaları

### 2. AppUsers Tablosu:

- **Amaç:** Sistemin kullanıcı bilgilerini Identity sistemi ile entegre bir şekilde tutar.
- **Alanlar:**
  - Id (PK, string): Identity sistemi tarafından üretilen benzersiz kullanıcı kimliği.
  - Name (string): Kullanıcının görünen adı.
  - Email (string): Kullanıcının e-posta adresi (Identity tarafından yönetilir).
- **Not:** Role alanı burada doğrudan tutulmayacak. Identity sistemi **IdentityRole** ve **AppUserRoles** (veya **AspNetUserRoles**) tabloları aracılığıyla rol yönetimini kendi içinde halledecektir.

### 3. Tickets Tablosu:

- **Amaç:** Her bir hatanın, talebin veya sorunun temel bilgilerini içerir.

- **Alanlar:**

- Id (PK, int): Benzersiz ticket kimliği.
- Title (string): Ticket'ın kısa başlığı.
- Description (string): Ticket'ın detaylı açıklaması.
- Type (Enum: Hata, Talep, Soru): Ticket'ın türünü belirtir.
- Priority (Enum: Dusuk, Orta, Yuksek): Ticket'ın önem derecesi.
- Status (Enum: Acik, Kapatildi): Ticket'ın mevcut durumu.
- CreatedAt (DateTime): Ticket'ın oluşturulma tarihi ve saati.
- CreatedByUserId (FK, string): Ticket'ı oluşturan kullanıcının AppUser.Id'si.
- AssignedToUserId (FK, string, nullable): Ticket'ın atandığı geliştiricinin AppUser.Id'si. Bir ticket henüz atanmamış olabilir, bu yüzden nullable (?) olmalı.

#### 4. TicketComments Tablosu:

- **Amaç:** Ticket'lar üzerindeki yorumları ve iletişim geçmişini tutar.

- **Alanlar:**

- Id (PK, int): Benzersiz yorum kimliği.
- TicketId (FK, int): Hangi ticket'a ait olduğunu gösterir (Tickets.Id'ye referans).
- CommentText (string): Yorumun içeriği.
- CommentedAt (DateTime): Yorumun yapıldığı tarih ve saati.
- UpdatedAt (DateTime): Yorumun guncellendiği tarih ve saati.
- CommentedByUserId (FK, string): Yorumu yapan kullanıcının AppUser.Id'si.

## 5. TicketAttachments Tablosu:

- **Amaç:** Ticket'lara eklenen dosyaların (ekran görüntüleri, log dosyaları vb.) bilgilerini tutar.
- **Alanlar:**
  - Id (PK, int): Benzersiz ek kimliği.
  - TicketId (FK, int): Hangi ticket'a ait olduğunu gösterir (Tickets.Id'ye referans).
  - FileName (string): Dosyanın orijinal adı.
  - FilePath (string): Dosyanın sunucuda saklandığı yol.

- ContentType (string): Dosyanın MIME türü (örneğin: image/png, application/pdf).
- UploadedAt (DateTime): Dosyanın yüklendiği tarih ve saati.
- UploadedByUserId (FK, string): Dosyayı yükleyen kullanıcının AppUser.Id'si.



## Veritabanı İlişkileri (Foreign Keys)

- Tickets:CreatedByUserId rightarrow AppUsers:Id (Bir kullanıcı birden fazla ticket oluşturabilir)
- Tickets:AssignedToUserId rightarrow AppUsers:Id (Bir geliştiriciye birden fazla ticket atanabilir)
- TicketComments:TicketId rightarrow Tickets:Id (Bir ticket'a birden fazla yorum yapılabilir)
- TicketComments:CommentedByUserId rightarrow AppUsers:Id (Bir kullanıcı birden fazla yorum yapabilir)
- TicketAttachments:TicketId rightarrow Tickets:Id (Bir ticket'a birden fazla dosya eklenebilir)
- TicketAttachments:UploadedByUserId rightarrow AppUsers:Id (Bir kullanıcı birden fazla dosya yükleyebilir)

## Enumlar:

```
public enum TicketPriority { Dusuk, Orta, Yuksek }  
public enum TicketStatus { Acik,Kapatildi }  
public enum TicketType { Hata, Talep, Soru }
```



## Kullanılacak Teknolojiler:

- **Backend:** ASP.NET Core 8 (MVC veya Razor Pages)
  - **Authentication:** Identity (Microsoft.AspNetCore.Identity)
  - **Database:** Entity Framework Core + SQL Server
  - **Frontend:** Razor Pages (veya basit MVC Views)
  - **UI Kütüphanesi :** Bootstrap+ Telerik Extensions
- 
- ASP.NET Core projesinde basit de olsa temiz bir katmanlı mimari (**örn: Controller/Service/Repository**)



# Yapılacaklar Listesi:

## 1. Proje Kurulumu

- ASP.NET Core projesi oluştur
- EntityFrameworkCore ve Identity paketlerini yükle
- SQL Server bağlantısı yapılandır

## 2. Kullanıcı Sistemi

- Kayıt, giriş, çıkış işlemleri
- Roller: User, Developer, Admin
- Yetki kontrolleri: `[Authorize(Roles = "Developer")]` gibi

## 3. Ticket İşlemleri

- Ticket oluşturma sayfası
- Ticket listeleme
- Ticket detay ve yorum alanı
- Ticket durumu değiştirme (Developer)
- Admin: Tüm ticket'ları ve kullanıcıları görebilsin



## Daha Sonra:

- ✓ **Basit Grafikler:** Admin paneline açık, çözülmüş ticket sayılarını gösteren basit pasta veya çubuk grafikler eklenebilir.
- • Ticket detaylarında dosya yükleme
- ✓ • Admin Dashboard (istatistiksel özet: kaç açık ticket, kaç çözüldü vs.)
- ✓ • Gelişmiş filtreleme (önem derecesi, duruma göre)
- ✓ • **Bildirimler (Basit Düzeyde):** Örneğin, bir kullanıcı ticket'ına yanıt geldiğinde veya bir geliştiriciye yeni bir ticket atandığında sistem içi basit bir bildirim (örneğin, zil ikonunda sayı) veya ilerleyen aşamalarda e-posta bildirimi düşünülebilir.



## Teslim Formatı:

- Temiz commit mesajları olacak
- ReadMe.md dosyası yazılacak (projenin amacı, teknolojiler vs.)
- Kod okunabilir olacak, yorum satırları eklenecek
- Projenin akışını anlatan dokuman hazırlanmış olacak