* **Proje Adı**: Kredi Başvuru Sistemi
* **Açıklama**: Bu proje, bir kredi başvuru sistemini simüle etmeyi amaçlamaktadır. Müşteriler, kredi başvurularını yapabilirler ve başvurular değerlendirilir. Proje, [ASP.NET](http://asp.net/) Core ile bir RESTful API geliştirmeyi (MediatR ile) ve kodunun birim testlerini (NUnit kullanılabilir) yazmayı içerecektir.
* **Gereksinimler**:
  + **Müşteri Entity**: Müşterilerin adı, soyadı, T.C. kimlik numarası ve e-posta adresi gibi temel bilgilerini saklayan bir müşteri sınıfı oluşturun.
  + **Kredi Başvurusu** Entity: Kredi başvurularının miktarı, vadesi, faiz oranı, başvuru tarihi, sonuç durumu gibi bilgileri saklayan bir kredi başvurusu sınıfı oluşturun.
  + **API Endpoint'leri**:
    - Müşteri bilgilerini eklemek için bir POST isteği kabul eden bir endpoint oluşturun.
    - Kredi başvurusu yapmak için bir POST isteği kabul eden bir endpoint oluşturun.
    - Tüm kredi başvurularını listelemek için bir GET isteği kabul eden bir endpoint oluşturun.
    - Belirli bir başvuruyu getirmek için bir GET isteği kabul eden bir endpoint oluşturun.
  + **Kredi Değerlendirmesi**: Kredi başvurularını değerlendiren basit bir algoritma oluşturun. Başvurunun sonuç durumunu hesaplayın (kabul, reddedildi, inceleme gerekiyor).  
    - **Kredi Değerlendirme Kuralları**:
      * **Kredi Puanı Kontrolü**: Müşterinin kredi puanı belirli bir eşik değerin üzerinde mi? **Örneğin**, kredi puanı 700 veya daha yüksekse kabul edilir. Aksi takdirde, reddedilir.
      * **Borç-Gelir Oranı Kontrolü**: Müşterinin aylık geliri, aylık borcunun belirli bir katı mı? **Örneğin**, borç-gelir oranı 0.4 veya daha düşükse kabul edilir. Aksi takdirde, reddedilir.
      * **Kredi Miktarı Sınırlaması**: Müşterinin talep ettiği kredi miktarı belirli bir eşik değerin altında mı? Örneğin, talep edilen kredi miktarı 100,000 TL veya daha düşükse kabul edilir. Aksi takdirde, reddedilir.
      * **Kredi Süresi Sınırlaması**: Kredi süresi belirli bir sürenin içinde mi? Örneğin, kredi süresi 60 ay veya daha kısa ise kabul edilir. Aksi takdirde, reddedilir.
      * **Kötü Kredi Geçmişi Kontrolü**: Müşterinin geçmişte kötü bir kredi geçmişi var mı? Eğer varsa, reddedilir.
  + **Unit Testler**: İş mantığı kodlarınız için unit testler yazın. Özellikle kredi başvurusu yapma işlemini ve sonuçları test edin.
  + **Dökümantasyon**: API'nin nasıl kullanılacağına dair basit bir dökümantasyon ekleyin.
  + **Hata Kontrolleri**: İstekleri işlerken hata durumlarını ele alın ve uygun hata kodları ve mesajları döndürün.
  + **Swagger**: Swagger veya benzeri bir araç kullanarak API'nizi belgelendirin ve test edilebilir hale getirin.
  + **Dependency Injection**: Servislerin kullanımı ve bağımlılıkları yönetmek için Dependency Injection kullanın.