Proje Adı: AI Destekli Abonelik İşlemleri Chatbotu

Proje Amacı:

Kullanıcının elektrik aboneliği ile ilgili işlemleri (iptal, ödeme, fatura itirazı, arıza bildirimi vs.) GPT API'si üzerinden etkileşimli şekilde yöneten bir chatbot arayüzü geliştirmek.

Görev Listesi

1. Proje Kurulumu

- Yeni bir .NET Core Web projesi başlat (tercihen ASP.NET Core MVC veya minimal API + Razor Pages).
- Gerekli NuGet paketlerini kur (HttpClient, Newtonsoft.Json, vs.)
- OpenAI GPT API anahtarını appsettings.json üzerinden yönetilecek şekilde yapılandır.

---

2. Landing Page

- Kullanıcıyı karşılayan basit bir landing page oluştur.
- Burada kullanıcıya "Ne işlem yapmak istiyorsunuz?" sorusu sorulsun.
- İşlem seçenekleri butonlar olarak gösterilsin:
    - Fatura Ödemesi
    - Abonelik İptali
    - Fatura İtirazı
    - Elektrik Arızası
    - Diğer
- Seçim yapıldığında ilgili senaryo ile GPT agent'i başlatılsın.

---

3. Chatbot Arayüzü (Frontend)

- Basit bir “chatbox” UI tasarla (HTML + JS + Bootstrap ile).
- Kullanıcının mesajı aşağıda girip gönderebildiği bir yapı olsun.
- Chat geçmişi yukarıdan aşağıya doğru mesaj balonları ile gösterilsin.
- Kullanıcının mesajı gönderildiğinde backend'e AJAX/Fetch ile gönderilsin.

---

4. Backend - GPT Agent Entegrasyonu

- Kullanıcının başlattığı senaryoya göre GPT agent prompt’unu oluştur.
- Chat geçmişini GPT’ye uygun şekilde sürekli aktarmak için session veya basit geçici hafıza sistemi oluştur (MemoryCache veya temp session kullanılabilir).
- OpenAI’ye gelen mesajı ve prompt’u POST et ve cevabı al.
- Cevabı frontend’e dön.

---

5. Basit Bilgi Toplama ve Kontrol

- Kullanıcıdan T.C. kimlik no, abone numarası gibi bilgiler istenmesi gerektiğinde bu adımlar senaryoya uygun işlesin.
- Bu bilgiler ön yüzde form halinde alınabilir ve chat içinde iletilebilir.
- Doğrulama kısmı şu an için fake/mock olabilir (örnek: belirli bir T.C. girilirse “doğru” kabul et).

---

6. Senaryo Yönetimi (Basit)

- Kullanıcının seçtiği senaryoya göre, başlangıç prompt’ları farklı olmalı.
- Her senaryo için bir `ChatScenario` enum’u tanımla ve senaryo bazlı prompt mantığını uygula.

- Senaryolar daha sonra geliştirilecekse, konfigürasyon dosyası veya sabit sınıf üzerinden tanımlanabilir.

---

7. Geliştirme & Test

- Temel senaryolar (örneğin ödeme ve iptal) için uçtan uca test yap.
- Farklı cihazlarda UI testi yap (responsive tasarım kontrolü).
- Kullanıcıdan gelen hatalı/eksik bilgilerle nasıl davranacağını test et.

---

8. Bonus (Opsiyonel)

- Kullanıcı oturumu takip etmek için SignalR ile canlı sohbet sistemi entegre etmeye çalışabilir.
- Temel bir admin paneli eklenip gelen kullanıcı mesajları ve senaryo kayıtları gösterilebilir.

Teknolojiler

- Backend: ASP.NET Core 7 veya 8
- Frontend: Razor Pages veya basit HTML/Bootstrap
- JS: Fetch API veya jQuery (geliştiriciye kalmış)
- AI: OpenAI GPT API (Chat Completion v1 endpoint)
- Session: TempData / MemoryCache / Cookie bazlı kullanıcı takibi