

Mükemmel düşünmüşsun 🔥

**Tentacle** ismi Hydra evreninde **uzanti, bağlantı ve görev yürüten kolları** temsil ettiği için tam olarak bu fikre oturuyor.

Yani Hydra “çekirdek zeka”ysa, Tentacle onun **işleri yöneten kolu** gibi duruyor — harika konsept uyumu 🤝

---



## Senin fikrini toparlayayım:

Tentacle iki ana modülden oluşacak:

### 1 Requests (İstekler / Görev Talepleri)

- Bir görev veya iş tanımı oluşturulabiliyor.
- Bu istek **başkası tarafından atanabiliyor** veya **kullanıcı kendisi için** oluşturulabiliyor.
- İçinde:
  - Başlık
  - Açıklama / detay
  - Sorumlu kişi
  - Durum (Open / InProgress / Done vs.)
  - Tarihler (CreatedAt, DueDate vs.)
  - Priority, Tag'ler
  - İlişkili notlar olabilir

Bu kısım, “iş takip” tarafını yönetiyor.

---

### 2 Notes (Kişisel Notlar / Düşünceler)

- Kullanıcı kendi kişisel düşüncelerini, fikirlerini veya teknik notlarını tutabiliyor.
- “Private” veya “Shared” olabilir (örneğin takım arkadaşlarıyla paylaşılan teknik notlar).
- Notlar isteğe bağlı olarak **Requests** ile ilişkilendirilebilir.

örnek: `Note.RequestId = Guid?`

- Markdown desteği veya tag bazlı arama planlanabilir.
- 

## 🎯 Kısa vizyon (bunu README veya proje tanımına yazabilirsın)

**Tentacle**, Hydra ekosisteminin görev ve düşünce yönetim uzantısıdır.

Kullanıcılar, kendi işlerini veya başkalarının taleplerini organize eder, notlarını

ilişkilendirir,  
böylece kişisel ve ekip verimliliğini aynı yerde yönetir.

---

## İleride genişleyebilecek alanlar:

- Request → Task alt kırılımı
  - Notes → AI destekli “öneri / özet çıkarma”
  - “Timeline” veya “Activity log” sistemi
  - “Hydra.Cortex” gibi bir üst zeka bu notları analiz edebilir
- 

İstersen sana bu yapının **temel class taslağını** (Request, Note, TentacleContext gibi) C#'ta Entity Framework'e uygun şekilde hazırlayayım.

Hem veritabanı hem mantık katmanına oturur, sonra Blazor tarafına rahat geçeriz.

Hazırlayayım mı o kısmı?