# CRSSoft Staj Projesi Sunum Taslağı

## 1. Problem & Hedef

* Hukuk öğrencileri ve stajyer avukatlar için **karmaşık ceza hukuku senaryolarını anlamak zor**.
* Çözüm: **RAG (Retrieval-Augmented Generation)** tabanlı bir karar destek sistemi.
* Amaç: TCK maddelerini ve Yargıtay kararlarını hızlıca bulup, LLM ile özetleyerek kullanıcıya sunmak.

## 2. Çözümün Özeti

* **Serbest metin** ile vaka girişi
* **Embedding** ve **vektör veritabanı** (Qdrant) ile ilgili TCK ve Yargıtay kararlarını bulma
* **Yerel LLM (Gemma 3:4B)** ile bağlamlı analiz
* **React arayüzü** üzerinden sonuçları kullanıcıya sunma

## 3. Mimari Diyagram

**Bileşenler:** - Backend: .NET Core Web API - Frontend: React + Tailwind - LLM: Gemma 3:4B (lokal) - Embedding: nomic-embed-text- Vektör DB: Qdrant - Veri Kaynağı: TCK & Yargıtay kararları

A diagram of a data processing process

Description automatically generated

## 4. Sistem Akışı

1. Kullanıcı vaka metnini girer
2. Backend embedding üretir
3. Qdrant’tan en yakın TCK ve emsal kararlar bulunur
4. LLM’e bağlamlı prompt gönderilir
5. Yanıt üretilir → Frontend’de gösterilir

## 5. Öne Çıkan Özellikler

* Gerçek karar ve yasa maddeleriyle **kaynaklı cevaplar**
* Hakimin takdir alanlarına işaret etme (ör. haksız tahrik, meşru müdafaa)
* Eğitim ve staj süreçlerinde **karar destek aracı**

## 6. Demo Ekranları

* **Ekran 1:** Vaka girişi
* **Ekran 2:** TCK maddeleri + kısa açıklama
* **Ekran 3:** Emsal Yargıtay kararı özeti
* **Ekran 4:** Uyarı (Legal advice değildir)

## 7. Muhtemel Sorular

* Bu sistem hukuki danışmanlık mı veriyor? → Hayır, **eğitim ve karar destek**.
* Veri kaynakları güncel mi? → TCK & Yargıtay kararları, gelecekte otomatik güncelleme.
* Neden RAG? → Hallucination riskini azaltmak ve kaynak göstermek için.
* Yerel LLM neden? → Gizlilik, maliyet ve hız avantajı.
* Veriler nasıl korunuyor? → PII maskeleme, loglama kuralları.

## 8. Geliştirilebilecek Yönler

* Güncel mevzuat otomatik entegrasyonu (mevzuat.gov.tr)
* Kullanıcı deneyimi geliştirme (senaryo şablonları, confidence score)
* Daha büyük/hukuk alanına özel LLM entegrasyonu
* Farklı hukuk alanları (Medeni hukuk, ticaret hukuku vs.)
* Ölçüm ve kalite metrikleri (precision@k, recall, kullanıcı memnuniyeti)

## 9. Bu Projede Neler Öğrendim?

* **RAG mantığı:** Embedding, chunking, retrieval, context enjeksiyonu
* **.NET Core Web API:** Middleware, Swagger, endpoint tasarımı
* **LLM entegrasyonu:** Yerel LLM çağırma, prompt mühendisliği
* **Qdrant:** Vektör arama, koleksiyon yönetimi
* **Veri işleme:** PDF/JSON’dan yasa ve emsal çıkarımı
* **Etik bakış:** Legal advice değil, karar desteği

## 10. Kavram Kutucukları (Ek Açıklamalar)

* **PII Maskeleme:** Kişisel verileri (TC no, telefon, adres) gizlemek/anonimleştirmek. Örn: 532-\***-**45.
* **Hallucination:** Modelin gerçekte olmayan bilgi uydurması. Çözüm: RAG ile kaynaklı cevap.
* **Middleware:** Request–response arasında çalışan ara katman. (Loglama, hata yönetimi, kimlik doğrulama)
* **Swagger:** API’leri test ve dökümantasyon için kullanılan arayüz.
* **Endpoint:** API’de belirli bir işlevi temsil eden URL. Örn: POST /api/cases/analyze
* **Precision@k:** İlk k sonuçtan kaçının doğru olduğunu ölçer. (Doğruluk)
* **Recall:** Tüm doğru sonuçların ne kadarının bulunduğunu ölçer. (Kapsayıcılık)

## 11. Yol Haritası

* Kısa vadede: Kullanıcı dostu arayüz, otomatik mevzuat güncelleme
* Orta vadede: Farklı hukuk alanları için modüller
* Uzun vadede: Profesyonel hukuk yazılımlarıyla entegrasyon

## 12. Kapanış

* Hukuk öğrencileri ve stajyerler için **akıllı asistan**
* Kaynaklı, şeffaf ve öğretici sonuçlar
* Geleceğe açık, geliştirilebilir bir altyapı

## 13. Q/A

“Teşekkürler – sorularınızı alabilirim.”