

Teknik Case Study: 8D Problem Çözme Platformu (MVP)

1. Proje Özeti ve Senaryo

Şirketimiz, üretim hatlarında yaşanan problemleri takip etmek ve kök nedenlerini analiz etmek amacıyla "8D Problem Çözme Metodolojisini" dijitalleştirmektedir.

Senden, bu büyük projenin "**Problem Tanımlama (D1-D2)**" ve "**Kök Neden Analizi (D4- D5)**" süreçlerini simüle eden, **Full Stack** bir prototip (MVP) geliştirmeni bekliyoruz.

Bu çalışma, hem **Siemens iX** tasarım sistemine uyum yeteneğini hem de **PHP** ile ilişkisel veri yapılarını (Recursive/Tree Data) nasıl modellediğini görmemizi sağlayacaktır.

2. Tech Stack

• Frontend:

- Modern JS Framework React
- UI Library (Zorunlu): [Siemens iX Design System](#).

• Backend:

- Dil: **PHP** Native
- API: RESTful prensiplerine uygun JSON API.

• Veritabanı:

- MySQL veya relational herhangi bir database

3. Fonksiyonel Gereksinimler

Uygulamanın 2 temel bölümden oluşması beklenmektedir:

Bölüm A: Dashboard (Problem Listesi & Tanımlama)

1. Liste Ekranı: Mevcut problemlerin listelendiği bir tablo.

- *Sütunlar*: ID, Başlık, Sorumlu Ekip (D1), Durum (Açık/Kapalı), Tarih,
- vs. ◦ *Komponent*: Ag-Grid

2. Yeni Kayıt (Modal): "Yeni Problem Ekle" butonu ile açılan bir form.

- Başlık, Detaylı Açıklama (D2) ve Sorumlu Ekip bilgileri girilerek kaydedilmeli.
- *Siemens iX-Modal komponent kullanılması zorunludur.*

Bölüm B: Kök Neden Analizi (D4) ve Çözüm (D5) - Kritik Bölüm

Kullanıcı listeden bir probleme tıkladığında detay sayfasına gitmeli ve aşağıdakileri yapabilmelidir:

1. Dinamik Kök Neden Ağacı (The Root Cause Tree):

- Problemin altına "**Neden?**" soruları sorularak sebepler eklenebilmelidir.
- Eklenen her sebebin altına, **sınırsız derinlikte** alt sebepler (Child Nodes) eklenebilmelidir.
- **Örnek:** Makine Durdu (Parent) -> Sigorta Attı (Child) -> Aşırı Yüklenme (Grand-child).
- **Görselleştirme:** Bu yapı Frontend tarafında **hiyerarşik (girintili)** olarak gösterilmelidir. (Siemens iX komponentleri veya iç içe geçmiş listeler kullanılabilir).
- *5 neden analizi görselleştirilmesi olarak geçmektedir.*

2. Kök Neden ve Aksiyon Belirleme:

- Ağaçtaki herhangi bir dal/sebepe "**Kök Neden (Root Cause)**" olarak işaretlenebilmelidir.
- Kök neden olarak işaretlenen madde için bir input alanı açılmalı ve "**Kalıcı Çözüm (Action)**" girilmelidir.

4. Beklenen Teknik Yaklaşım (İpuçları)

Değerlendirme sırasında aşağıdaki noktalara dikkat edilecektir:

- **Veri Modeli (PHP):** Ağaç yapısını (Tree/Recursive Data) veritabanında veya veri kaynağında nasıl modellediğiniz önemlidir. (Örn: parent_id ilişkisi).
- **API Yapısı:** Frontend'in bu ağaç yapısını tüketebileceği uygun bir JSON formatı dönülmelidir.
- **Siemens iX Kullanımı:** Dokümantasyonun (ix.siemens.io) doğru okunup, uygun komponentlerin (Butonlar, Kartlar, Inputlar) standartlara göre kullanılması.

5. Teslimat

- Kodlarınızı bir **GitHub** (veya GitLab) reposuna yükleyiniz.
- Projenin lokalde nasıl çalıştırılacağını anlatan kısa bir **README.md** dosyası ekleyiniz

- Opsiyonel: Uygulamanın bir hostinge veya platforma deploy edilmiş halinin linki gönderilebilir.