

# Web Tabanlı Sosyal Ağ Analizi ve Görselleştirme Platformu

Kocaeli Üniversitesi, Bilişim Sistemleri Mühendisliği, Yazılım Geliştirme Laboratuvarı-I

Mehmet Enes Dinç  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
mehmetenes2004.dinc@gmail.com  
231307043

Ulaş Can Demirci  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
Kocaeli, Türkiye  
211307044

**Özet**—Bu proje, bireyler arasındaki karmaşık ilişkileri modellemek ve analiz etmek amacıyla geliştirilen web tabanlı bir Sosyal Ağ Analizi (SNA) yazılımıdır. Proje, Nesne Yönelimli Programlama (OOP) prensipleri temel alınarak JavaScript dili ile geliştirilmiş ve kullanıcı erişilebilirliğini maksimize etmek amacıyla Render bulut platformu üzerinde canlıya (production) alınmıştır. Sistem; kullanıcılara dinamik graf yönetimi imkanı sunarken, arka planda BFS, DFS, Dijkstra, A\* (A-Star), Degree Centrality ve Welsh-Powell renklendirme algoritmalarını çalıştırmaktadır. Dğüümler arası ağırlıklar, kullanıcı özniteliklerine dayalı matematiksel bir modelle dinamik olarak hesaplanmaktadır.

**Index Terms**—Sosyal Ağ Analizi, Graf Teorisi, Bulut Bilişim, Render, Algoritma Analizi, JavaScript.

## I. GİRİŞ

Günümüzde sosyal ağların analizi, topluluk yapılarının anlaşılması, bilgi yayılımının modellenmesi ve etkili aktörlerin tespiti açısından büyük önem taşımaktadır. Bu proje kapsamında, teorik graf algoritmalarının pratik bir uygulama üzerinde somutlaştırılması hedeflenmiştir[cite: 7].

Projenin temel amacı; kullanıcıların (düğümlerin) ve etkileşimlerin (kenarların) görsel olarak yönetilebildiği, farklı yol bulma ve analiz algoritmalarının sonuçlarının anlık olarak izlenebildiği interaktif bir platform oluşturmaktır. Geliştirilen uygulama, modern web teknolojileri kullanılarak platform bağımsız hale getirilmiş ve bulut tabanlı bir sunucu (Render) üzerinden kesintisiz erişime açılmıştır.

## II. SİSTEM MIMARISI VE TASARIM

### A. Yazılım Mimarisi ve OOP

Proje, katmanlı mimari ve Nesne Yönelimli Tasarım prensiplerine sıkı sıkıya bağlı kalınarak geliştirilmiştir[cite: 44]. Sistemdeki temel sınıflar şunlardır:

- **Node (Düğüm):** Kullanıcı verilerini (ID, İsim, Aktivite Puanı vb.) ve görsel koordinatları kapsüller.
- **Edge (Kenar):** İki düğüm arasındaki ilişkiyi ve bu ilişkinin maliyetini (ağırlık) temsil eder.
- **GraphManager:** Tüm graf yapısının durumunu (state) yönetir, ekleme/silme işlemlerini kontrol eder.

- **AlgorithmService:** Graf üzerinde çalışan algoritmaların soyutlandığı servis katmanıdır[cite: 45].

### B. Dinamik Ağırlık Hesaplama Modeli

Sistemde iki kullanıcı arasındaki bağın gücü (maliyeti), kullanıcıların öznitelik benzerliklerine göre belirlenir.  $i$  ve  $j$  düğümleri arasındaki ağırlık ( $W_{i,j}$ ), aşağıdaki formül ile hesaplanmaktadır:

$$W_{i,j} = 1 + \sqrt{(\Delta Aktivite)^2 + (\Delta Etkileşim)^2 + (\Delta Bağlantı)^2} \quad (1)$$

Bu formül sayesinde, benzer özelliklere sahip kullanıcılar arasındaki "mesafe" azalmakta, farklı profillerdeki kullanıcılar arasındaki mesafe artmaktadır.

## III. KULLANILAN ALGORİTMALAR

Projede, sosyal ağ analizi için literatürde kabul görmüş temel ve ileri düzey algoritmalar entegre edilmiştir[cite: 33].

### A. Arama ve Gezinme Algoritmaları

**BFS (Breadth-First Search):** Çizge üzerinde genişlik öncelikli arama yaparak, bir kullanıcının doğrudan ve dolaylı tüm bağlantılarını katman katman keşfeder. **DFS (Depth-First Search):** Derinlik öncelikli arama stratejisi ile ağın en uç noktalarına kadar ulaşılabilirlik analizi yapar[cite: 34].

### B. En Kısa Yol Algoritmaları

**Dijkstra Algoritması:** Ağırlıklı graflarda, başlangıç düğümünden diğer tüm düğümlere olan en kısa yolları hesaplar. Dinamik ağırlık formülümüzdeki maliyetleri baz alır[cite: 35]. **A\* (A-Star) Algoritması:** Dijkstra'nın sezgisel (heuristic) bir varyasyonudur. Hedefe yönelik tahmini bir maliyet fonksiyonu kullanarak arama uzayını daraltır ve performansı artırır.

**Degree Centrality (Merkezlilik):** Ağdaki en popüler ve etkili kullanıcıları tespit etmek için düğüm derecelerini hesaplar[cite: 37]. **Welsh-Powell Renklendirme:** Ağdaki komşu düğümlerin aynı renge sahip olmamasını garanti ederek, çizmeyi minimum sayıda renk ile boyar. Bu, ayrık toplulukların görselleştirilmesinde kullanılmıştır[cite: 38].

Tablo I  
KULLANILAN ALGORİTMALARIN TEORİK KARMAŞIKLIKLARI

Algoritma	Zaman Karmaşıklığı	Kullanım Alanı
BFS	$O(V + E)$	Bağlantı Keşfi
Dijkstra	$O(E + V \log V)$	En Kısa Yol
Welsh-Powell	$O(V^2 + E)$	Renklendirme
Degree Centrality	$O(V + E)$	Etki Analizi

#### IV. DAĞITIM VE CANLI ORTAM (DEPLOYMENT)

Geliştirilen uygulamanın erişilebilirliği ve sürdürülebilirliği, modern DevOps pratikleri kullanılarak sağlanmıştır.

##### A. Render Bulut Platformu

Uygulama, statik web sitesi barındırma ve otomatik dağıtım özellikleri nedeniyle **Render** platformu üzerinde canlıya alınmıştır. Bu tercih şu avantajları sağlamıştır:

- **Sürekli Entegrasyon (CI/CD):** GitHub deposu ile Render entegre edilmiştir. GitHub'a gönderilen (push) her kod güncellemesi, otomatik olarak algılanmakta, derlenmekte ve canlı sunucuda güncellenmektedir[cite: 19].
- **Yüksek Erişilebilirlik:** Uygulama, global CDN (İçerik Dağıtım Ağı) üzerinden sunularak, kullanıcıların coğrafi konumundan bağımsız olarak yüksek hızda erişim sağlanmıştır.
- **SSL Güvenliği:** Tüm veri trafiği HTTPS protokolü üzerinden şifreli olarak gerçekleştirilmektedir.

#### V. SONUÇLAR

Yapılan testler sonucunda; uygulamanın 50-100 düğümlü orta ölçekli graflarda dahi algoritmaları milisaniyeler mertebesinde çalıştırdığı gözlemlenmiştir. Dinamik ağırlık hesaplama modülü sayesinde, kullanıcılar arasındaki ilişki maliyetleri başarılı bir şekilde simüle edilmiş ve Dijkstra algoritması bu maliyetlere göre en optimum rotayı çizmiştir.

Render üzerindeki canlı sürüm, farklı cihaz ve tarayıcılarda test edilmiş, "Responsive" (duyarlı) tasarım sayesinde mobil ve masaüstü platformlarda sorunsuz çalıştığı doğrulanmıştır.

#### TEŞEKKÜR

Bu proje, Kocaeli Üniversitesi Teknoloji Fakültesi Bilişim Sistemleri Mühendisliği Bölümü'nde verilen Yazılım Geliştirme Laboratuvarı-I dersi gereksinimleri doğrultusunda hazırlanmıştır.

#### KAYNAKLAR

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," 3rd ed. MIT Press, 2009.
- [2] M. Newman, "Networks: An Introduction," Oxford University Press, 2010.
- [3] Render Documentation, "Static Site Deployment," 2025. [Online]. Available: <https://render.com/docs>