

ANADOLU ÜNİVERSİTESİ

Açıköğretim Fakültesi

AR-GE YÖNETİM DASHBOARD'I

Yapay Zekâ Destekli Geliştirme Süreci ile
Teknik Dokümantasyon ve Akademik Rapor

Hazırlayan: Sefa Emre Öncü

Kurum: Anadolu Üniversitesi — Açıköğretim Fakültesi

Versiyon: 3.0 | Tarih: 28 Şubat 2026

Özet

Bu çalışmada, Anadolu Üniversitesi Açıköğretim Fakültesi bünyesinde geliştirilen Ar-Ge Yönetim Dashboard'ının teknik mimarisi, yazılım geliştirme süreci, işlevsel özellikleri ve kurumsal katkıları akademik bir çerçevede kapsamlı biçimde ele alınmaktadır. Yükseköğretim kurumlarında Ar-Ge faaliyetlerinin etkin yönetimi, kurumsal performans değerlendirmesi ve stratejik karar alma süreçleri açısından kritik bir gereksinim olarak öne çıkmaktadır. Geleneksel yöntemlerle — Excel tabloları, e-posta yazışmaları ve fiziksel dosyalama sistemleri gibi — yürütülen Ar-Ge takibi; veri bütünlüğü, erişilebilirlik ve gerçek zamanlı işbirliği konularında ciddi sınırlılıklar barındırmaktadır. Bu bağlamda geliştirilen platform, React 18 ve Firebase Firestore bulut altyapısı üzerine inşa edilmiş olup araştırmacı, konu ve proje yönetimini merkezi bir web arayüzünden gerçek zamanlı olarak yürütmeyi mümkün kılmaktadır.

Geliştirme sürecinde, üretken yapay zekâ modellerinin yazılım mühendisliğine entegrasyonunu konu alan güncel bir yaklaşım olan 'Vibe Coding' metodolojisini benimsenmiştir (Karpathy, 2025). Bu metodoloji çerçevesinde, yaklaşık 7.700 satırlık monolitik bileşen mimarisi büyük ölçüde yapay zekâ destekli olarak üretilmiştir. Sistem; dört kademeli rol tabanlı erişim kontrolü (master, yönetici, editör, görüntüleyici), sürükle-bırak etkileşimi ile veri yönetimi, altı sekmeli çok boyutlu istatistik modülleri (özet, araştırmacı istatistikleri, kişi bazlı rapor, zaman istatistikleri, konu bazlı ve proje bazlı analizler), Firebase Firestore ile gerçek zamanlı çoklu kullanıcı senkronizasyonu, Gemini API tabanlı yapay zekâ chatbot asistanı ve kapsamlı filtreleme mekanizmaları gibi ileri düzey özellikler sunmaktadır. Ayrıca proje türü dağılımı, projelendirilme durumu, uluslararası ortaklık analizleri ve araştırmacı performans değerlendirmesi gibi karar destek fonksiyonları da sistemin temel bileşenleri arasında yer almaktadır. Bu çalışma, yükseköğretim kurumlarında Ar-Ge faaliyetlerinin dijital yönetimi için ölçülebilir, sürdürülebilir ve tekrarlanabilir bir referans model ortaya koymaktadır.

Anahtar Kelimeler: Ar-Ge Yönetimi, Vibe Coding, React Dashboard, Firebase Firestore, Yapay Zekâ Destekli Yazılım Geliştirme

Abstract

This study comprehensively examines the technical architecture, software development process, functional features, and institutional contributions of the R&D Management Dashboard developed at Anadolu University, Faculty of Open Education. Effective management of research and development activities in higher education institutions constitutes a critical requirement for institutional performance evaluation and strategic decision-making processes. Traditional methods of R&D tracking — including spreadsheets, email correspondence, and physical filing systems — present significant limitations in terms of data integrity, accessibility, and real-time collaboration. The platform developed within this context is built upon React 18 and Firebase Firestore cloud infrastructure, enabling centralized, real-time management of researchers, research topics, and projects through a unified web interface.

Throughout the development process, the 'Vibe Coding' methodology — a contemporary approach addressing the integration of generative AI models into software engineering — was adopted (Karpathy, 2025). Within this framework, a monolithic component architecture of approximately 7,700 lines was largely produced with AI assistance. The system offers

advanced capabilities including four-tier role-based access control (master, administrator, editor, viewer), drag-and-drop data management interactions, six-tab multi-dimensional statistics modules (summary, researcher statistics, person-based reports, time statistics, topic-based and project-based analyses), real-time multi-user synchronization via Firebase Firestore, a Gemini API-powered AI chatbot assistant, and comprehensive filtering mechanisms. Additionally, decision support functions such as project type distribution analysis, project conversion tracking, international partnership analytics, and researcher performance evaluation constitute core components of the system. This work presents a scalable, sustainable, and replicable reference model for the digital management of R&D activities in higher education institutions.

Keywords: R&D Management, Vibe Coding, React Dashboard, Firebase Firestore, AI-Assisted Software Development

1. Giriş ve Genel Bakış

Bu rapor, Anadolu Üniversitesi Açıköğretim Fakültesi bünyesinde geliştirilen Ar-Ge Yönetim Dashboard'ının teknik mimarisini, yapay zekâ destekli geliştirme sürecini, uygulama kararlarını, karşılaşılan sorunları ve çözüm yaklaşımlarını akademik çerçevede açıklamaktadır. Proje, üniversitenin Ar-Ge birimlerinde yürütülen araştırma konularının, projelerin ve araştırmacıların merkezi bir web arayüzünden yönetilmesine hizmet etmektedir. Geliştirme süreci boyunca üretken yapay zekâ (Generative AI) modellerinin yazılım geliştirmeye entegrasyonunu konu alan güncel bir yaklaşım olan 'Vibe Coding' metodolojisi benimsenmiştir (Karpathy, 2025; Anthropic, 2025).

1.1 Projenin Amacı ve Motivasyonu

Yükseköğretim kurumlarında Ar-Ge faaliyetlerinin etkin yönetimi, kurumsal performans ve akademik üretkenlik açısından kritik önem taşımaktadır. Geleneksel yöntemlerle (Excel tabloları, e-posta yazışmaları, fiziksel dosyalama) yürütülen Ar-Ge takibi; veri bütünlüğü, erişilebilirlik ve gerçek zamanlı işbirliği konularında ciddi sınırlılıklar barındırmaktadır. Bu proje, söz konusu sınırlılıkları aşmak üzere modern web teknolojileri ve bulut altyapısı kullanılarak merkezi, gerçek zamanlı ve çok kullanıcılı bir Ar-Ge yönetim platformu geliştirmeyi amaçlamaktadır. Platform sayesinde birden fazla yöneticinin eş zamanlı olarak veri girişi ve görüntüleme yapabilmesi, kurumsal Ar-Ge verilerinin tutarlılığının güvence altına alınması ve karar alma süreçlerinin hızlandırılması hedeflenmektedir.

1.2 Problem Durumu

Türkiye'deki yükseköğretim kurumlarının büyük çoğunluğu, Ar-Ge faaliyetlerini takip etmek için yapılandırılmış dijital sistemlerden yoksundur. Araştırma konularının hangi araştırmacılara atandığı, projelerin hangi konularla ilişkili olduğu, bütçe dağılımları ve zaman çizelgeleri gibi kritik bilgiler yoğunlukla dağınık dosyalarda veya bireysel notlarda tutulmaktadır. Bu durum, veri kaybı riskini artırmakta, kurumsal hafızayı zayıflatmakta ve yönetim kademesinin bütüncül bir görünüm elde etmesini güçleştirmektedir.

Anadolu Üniversitesi Açıköğretim Fakültesi özelinde, Ar-Ge birimine bağlı çok sayıda araştırmacı, konu ve proje bulunmaktadır. Birden fazla yöneticinin eş zamanlı olarak bu verilere erişip güncelleme yapması gerekliliği, geleneksel dosya tabanlı yaklaşımları işlevsiz kılmaktadır. Araştırmacı-konu-proje arasındaki karmaşık ilişki ağı, düz tablo yapılarında yeterince temsil edilememektedir. Araştırma konularının öncelik sıralaması, kategorilere göre dağılımı, proje türü bazlı analiz, araştırmacı bazında iş yükü dağılımı ve fikir sahipliği takibi gibi yönetsel ihtiyaçlar da mevcut araçlarla karşılanamamaktadır.

1.3 Kapsam

Bu çalışma, yukarıda tanımlanan problemlere çözüm üzere tasarlanan Ar-Ge Yönetim Dashboard'ının teknik mimarisini, kullanılan teknolojileri ve yapay zekâ destekli geliştirme sürecini kapsamaktadır. Dashboard; araştırmacı yönetiminden proje takibine, gerçek zamanlı senkronizasyondan yedekleme sistemine kadar geniş bir işlevsellik yelpazesi sunmaktadır. Platformun kapsadığı temel modüller aşağıda sıralanmaktadır:

- Araştırmacı Yönetimi: Kişi kartları, akademik unvan, eğitim durumu, ilgi alanları, proje türü istatistikleri
- Konu Yönetimi: Araştırma konuları, öncelik/durum/kategori atamaları, araştırmacı ilişkilendirme, fikir sahipliği

- Proje Yönetimi: Konulara bağlı projeler, bütçe, zaman çizelgesi, proje türü, çapraz senkronizasyon
- Gerçek Zamanlı Senkronizasyon: Firebase Firestore ile bulut tabanlı çoklu admin desteği
- İstatistik ve Analitik: Özeti, kişi, zaman, kurum bazlı grafikler; sıralama tablosu
- Yedekleme Sistemi: JSON dışa/İçe aktarım, Firestore yedekleme, 30 günlük otomatik hatırlatma
- Yapılandırma Yönetimi: Roller, durumlar, öncelikler, kategoriler, proje türleri

2. Literatür Taraması

Bu bölüm, projenin dayandığı kuramsal çerçeveyi ve ilgili alandaki mevcut çalışmaları incelemektedir. Yükseköğretimde Ar-Ge yönetimi, yapay zekâ destekli yazılım geliştirme ve bulut tabanlı gerçek zamanlı sistemler konularında literatürdeki temel çalışmalara ve kavramsal temellere degeinilmektedir.

2.1 Yükseköğretimde Ar-Ge Yönetimi

Yükseköğretim kurumları bilgi üretiminin ve yenilikçiliğinin temel merkezleri olarak kabul edilmektedir. Araştırma faaliyetlerinin sistematik biçimde yönetilmesi, kurumsal Ar-Ge performansını doğrudan etkilemektedir. Özellikle Türkiye'de üniversitelerin TÜBİTAK proje sayıları, SCI yayın üretkenlikleri ve patent başvuruları gibi metriklerle değerlendirilmesi, Ar-Ge yönetim sistemlerine olan ihtiyacı artırmaktadır. Dijital dönüşüm bağlamında yükseköğretim kurumları, iç süreçlerini modernize etme baskısıyla karşı karşıya bulunmaktadır. Web tabanlı yönetim panelleri bu açığı kapatmaya yönelik etkin çözümler sunmaktadır (Altbach vd., 2019; YÖK, 2023).

2.2 Yapay Zekâ ve Yazılım Geliştirme

Büyük Dil Modelleri (LLM), yazılım mühendisliği alanında devrim niteliğinde dönüşümlere yol açmıştır. OpenAI'nın GPT serisi, Anthropic'in Claude modeli ve Google'ın Gemini'si gibi üretken YZ modelleri, kod üretimi, hata ayıklama, kod incelemesi ve dokümantasyon gibi görevlerde geliştiricilere önemli katkılar sağlamaktadır (Chen vd., 2021; Vaswani vd., 2017). Bu modellerin yazılım geliştirme sürecine entegrasyonu, hem bireysel üretkenliği artırmakta hem de daha önce uzmanlık gerektiren görevlerin daha geniş kitleler tarafından yerine getirilmesini mümkün kılmaktadır.

2.2.1 Vibe Coding

Andrej Karpathy tarafından 2025 yılında popülerleştirilen 'Vibe Coding' kavramı, geliştiricinin yapay zekâ ile doğal dilde iletişim kurarak kod üretmesini ifade etmektedir (Karpathy, 2025). Bu yaklaşımda geliştirici; gereksinimleri, mimarileri ve çözüm stratejilerini doğal dilde tanımlamakta, YZ modeli ise bu tanımları çalışır koda dönüştürmektedir. Geliştirici, üretilen kodu inceleyerek onaylamakta, düzeltme talimatları vermekte veya yönlendirme yapmaktadır. Bu paradigma değişimi, yazılım geliştirme sürecinde insan rolünün yeniden tanımlanması anlamına gelmektedir.

2.2.2 AI Agent (Yapay Zekâ Ajanı)

Yapay zekâ ajanları, belirli görevleri otonom biçimde gerçekleştirebilen, çevresini algılayabilen ve kararlara göre eylem serisini planlayabilen yazılım varlıklarıdır. Yazılım geliştirmede agent'lar; dosya okuma/yazma, terminal komutu çalışma, web arama ve kod analizi gibi araçları kullanarak karmaşık görevleri adım adım yerine getirebilmektedir (Wang vd., 2024). Geleneksel kod tamamlama araçlarından farklı olarak, YZ ajanları birden fazla dosyayı aynı anda analiz edebilmekte, bağamlar arası karışım yapabilmekte ve çok adımlı görevleri planlayarak uygulayabilmektedir.

2.2.3 Üretken YZ (Generative AI / GenAI)

Metin, kod, görsel ve ses gibi içerikleri üretebilen yapay zekâ sistemleri, üretken YZ olarak sınıflandırılmaktadır. Yazılım geliştirmede GenAI; kod üretimi, UI tasarıımı, dokümantasyon hazırlama, test senaryosu oluşturma ve refaktöring önerileri gibi geniş bir yelpazede

kullanılmaktadır (Brown vd., 2020). Üretken YZ'nin yazılım mühendisliğine etkisi, yalnızca kod yazmayı hızlandırmakla sınırlı kalmamakta; aynı zamanda tasarım kararlarının tartışılması, alternatif mimarilerin değerlendirilmesi ve teknik dokümantasyonun otomatik üretilmesi gibi bilişsel süreçlere de katkıda bulunmaktadır.

2.2.4 Kural Tabanlı ve LLM Tabanlı Yaklaşımların Karşılaştırması

Geleneksel kural tabanlı sistemler, önceden tanımlanmış if-then kuralları ve karar ağaçları ile çalışmaktadır. Bu sistemler deterministik, öngörlülebilir ve kontrol edilebilir olmakla birlikte; esneklik ve doğal dil anlayışı açısından sınırlıdır. LLM tabanlı sistemler ise doğal dil anlayışı sayesinde esnek, bağlam duyarlı ve yaratıcı yanıtlar üretebilmektedir. Bu projede her iki yaklaşım da farklı katmanlarda kullanılmıştır: Dashboard'in kendi iş mantığı kural tabanlı olarak tasarlanmış; geliştirme süreci ise LLM tabanlı ajanlarla yürütülmüştür (Russell & Norvig, 2021).

2.3 Yükseköğretimde Yapay Zekâ Kullanımı

Yükseköğretim kurumlarında YZ kullanımı; öğrenci destek sistemleri, uyarlanabilir öğrenme platformları, idari süreç otomasyonu ve araştırma yönetimi gibi alanlarda hızla yaygınlaşmaktadır (Zawacki-Richter vd., 2019). Bu proje, yükseköğretimde YZ kullanımının farklı bir boyutunu ortaya koymaktadır: kurumsal yazılım geliştirme sürecinin kendisinde yapay zekâdan yararlanma. Bir üniversite yönetim aracının, YZ ajanları ile birlikte geliştirilmesi; hem ürünün hem de sürecin YZ ile ilişkisini somutlaştırmakta ve bu alanda uygulamalı bir örnek oluşturmaktadır.

3. Teknoloji Yığını ve Mimari

Bu bölümde, Ar-Ge Yönetim Dashboard'ının geliştirilmesinde kullanılan teknolojiler, bileşen mimarisi ve veri modeli detaylı biçimde açıklanmaktadır. Teknoloji seçimleri; gerçek zamanlı çoklu kullanıcı desteği, hızlı prototipleme ve kolay dağıtım gereksinimlerine göre yapılmıştır (Google, 2024; Meta, 2024).

3.1 Kullanılan Teknolojiler

Katman	Teknoloji	Açıklama
Frontend	React 18	Hooks tabanlı SPA, fonksiyonel bileşenler
Stil	Tailwind CSS	Utility-first CSS çerçevesi, responsive tasarım
Build	Vite	ESM tabanlı hızlı bundler, HMR desteği
Veritabanı	Firebase Firestore v9	NoSQL bulut DB, gerçek zamanlı snapshot
Hosting	Firebase Hosting	CDN destekli küresel dağıtım
İkonlar	Lucide React	30+ ikon bileşeni (SVG tabanlı)
Kimlik Doğrulama	Özel Uygulama	Basit parola tabanlı, 4 rol seviyesi
YZ Ajansı	Claude Opus 4 (Anthropic)LM tabanlı AI agent, vibe coding	

3.2 Bileşen Mimarisi

Uygulama, monolitik tek dosya mimarisi kullanmaktadır. Dashboard.jsx dosyası yaklaşık 7.650 satırından oluşmakta olup tüm UI bileşenlerini, iş mantığını, state yönetimini ve Firestore senkronizasyonunu barındırmaktadır. Bu mimari tercih, vibe coding metodolojisinin doğal bir sonucudur: YZ ajansı tek dosya üzerinde çalışarak bağlamı kaybetmeden hızlı iterasyonlar gerçekleştirebilmektedir. Aşağıda dosya yapısı özetlenmektedir:

Dosya	Satır	İşlev
Dashboard.jsx	~7.650	Ana bileşen: UI + state + Firestore sync
firebase.js	~30	Firebase konfigürasyonu, memoryLocalCache
AuthContext.jsx	~60	Kimlik doğrulama: 4 rol yönetimi
LoginPage.jsx	~80	Giriş sayfası: kullanıcı adı + şifre
App.jsx	~25	Rota yönetimi: auth kontrolü
main.jsx	~10	Uygulama giriş noktası: React DOM

3.3 Firestore Veri Modeli

Tüm veriler Firestore'da 'arge' koleksiyonu altında 11 doküman halinde tutulmaktadır. Her doküman, belirli bir veri tipini temsile etmekte ve items veya data alanı içermektedir. Bu yapı, Firestore'un doküman boyutu sınırlamalarına uygun olup aynı zamanda onSnapshot ile tek seferde tüm koleksiyonun dinlenmesine olanak tanımaktadır.

Doküman	Yapı	Açıklama
researchers	{ items: [...] }	Tüm araştırmacı kayıtları
topics	{ items: [...] }	Araştırma konuları
projects	{ items: [...] }	Projeler ve bağlı konular
quicklinks	{ items: [...] }	Hızlı erişim bağlantıları
cfg_roles	{ data: {...} }	Araştırmacı rolleri
cfg_statuses	{ data: {...} }	Konu/proje durum tanımları
cfg_priorities	{ data: {...} }	Öncelik seviyeleri
cfg_ptypes	{ items: [...] }	Proje türleri
cfg_categories	{ items: [...] }	Konu kategorileri
cfg_degrees	{ items: [...] }	Akademik dereceler
cfg_edustatus	{ items: [...] }	Eğitim durumları

3.4 Varlık İlişki Modeli

Sistemdeki üç ana varlık (araştırmacı, konu, proje) arasında çok-çok (many-to-many) ilişkiler bulunmaktadır. Bu ilişkiler, her varlığın kendi dokümanı içinde referans dizileri ile temsil edilmektedir. İlişki modeli, çift yönlü senkronizasyon mekanizmasının temelini oluşturmaktadır.

Araştırmacı ↔ Konu: Bir konu birden fazla araştırmaciya sahip olabilmektedir. Her ilişki rol ve isOwner (fikir sahibi) bayrağı taşımaktadır. Araştırmacı bir konudan çıkarıldığında, bağlı projelerdeki varlığı da gözden geçirilmektedir.

Konu ↔ Proje: Bir proje en az bir konuya bağlı olmak zorundadır; bu, sistemin temel iş kurallarından biridir. Bir konu birden fazla projede yer alabilmekte ve projeye konudan gelen araştırmacılar otomatik olarak eklenmektedir.

Araştırmacı ↔ Proje: Araştırmacılar projeye hem doğrudan hem de bağlı konular üzerinden katılabilmektedir. Konudan gelen araştırmacılar, konu-proje ilişkisi kesildiğinde otomatik olarak temizlenmektedir.

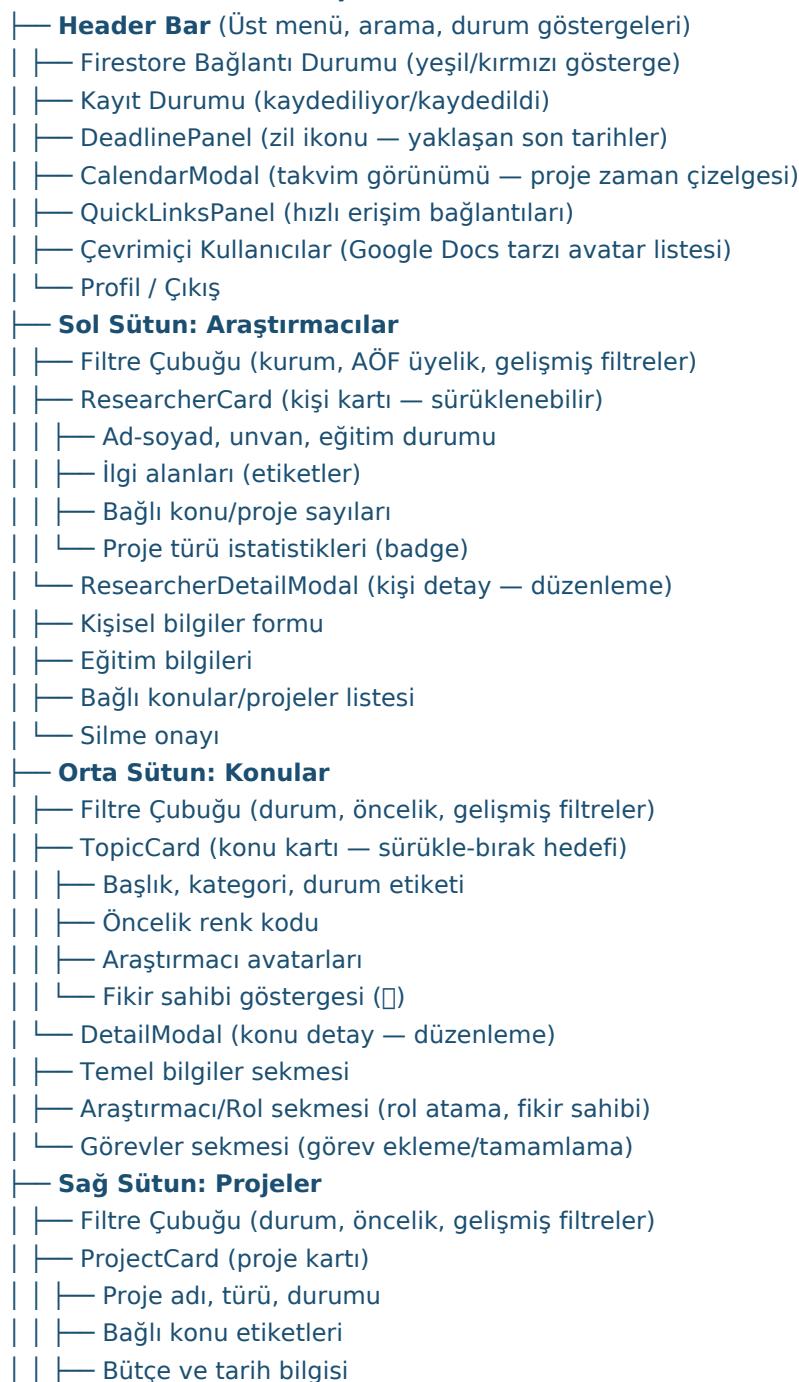
4. Uygulama Bileşen Ağacı ve Özellikler

Bu bölümde, Dashboard uygulamasının bileşen hiyerarşisi ağaç yapısı olarak sunulmakta ve her bir bileşenin işlevi açıklanmaktadır. Uygulama, 20'den fazla özelleştirilmiş React bileşeninden oluşmakta olup tamamı Dashboard.jsx dosyası içinde tanımlanmıştır.

4.1 Bileşen Hiyerarşisi (Ağaç Yapısı)

Aşağıdaki şema, uygulamadaki tüm bileşenleri ve aralarındaki hiyerarşik ilişkileri göstermektedir. Kök bileşen olan ArGeDashboard, tüm state yönetimini ve Firestore senkronizasyonunu üstlenmekte; alt bileşenler ise belirli UI ve iş mantığı sorumluluklarını taşımaktadır.

ArGeDashboard (Ana Bileşen — ~7.650 satır)



- └ Araştırmacı avatarları + fikir sahipleri
- └ DetailModal (proje detay — düzenleme)
- └ Temel bilgiler + bağlı konular
- └ Araştırmacı/Rol sekmesi
- └ Görevler sekmesi
- └ **Modallar**
 - └ AddItemModal (araştırmacı/konu/proje ekleme formu)
 - └ SettingsModal (yapilandırma yönetimi)
 - └ Roller, Durumlar, Öncelikler
 - └ Kategoriler, Proje Türleri
 - └ Akademik Dereceler, Eğitim Durumları
 - └ StatsModal (istatistik ve grafikler — 6 sekme)
 - └ Özet sekmesi (genel istatistik, pasta/çubuk grafik)
 - └ Araştırmacı İst. sekmesi (unvan dağılımı, proje türü)
 - └ Kişi Bazlı Rapor sekmesi (araştırmacı detay)
 - └ Zaman İst. sekmesi (yıl/ay bazlı trend)
 - └ Konu Bazlı sekmesi (durum dağılımı, proje türü)
 - └ Proje Bazlı sekmesi (uluslararası, ülke/kurum analizi)
 - └ LeaderboardModal (sıralama tablosu)
 - └ TableViewModal (tablo görünümü + CSV dışa aktarım)
 - └ CalendarModal (takvim — proje zaman çizelgesi)
- └ **Yardımcı Bileşenler**
 - └ Badge, ProgressBar, Avatar, InfoRow
 - └ FilterDropdown, RoleSelectPopup
 - └ TaskItem, Toast
 - └ SimplePieChart, SimpleBarChart, SimpleLineChart
 - └ ArGeChatbot (kural tabanlı sohbet asistanı)
- └ **Yedekleme Paneli** (yalnızca master)
 - └ JSON indirme / yükleme
 - └ Firestore yedek alma
 - └ 30 gün uyarı banner'ı

4.2 Kart Bileşenleri: Oluşturma, Düzenleme ve Silme

Her varlık türü (araştırmacı, konu, proje) için özelleştirilmiş kart bileşeni tasarlanmıştır. Kartlar, ilgili varlığın en önemli bilgilerini kompakt bir formatta sunmakta ve CRUD (oluşturma, okuma, güncelleme, silme) işlemlerini desteklemektedir.

4.2.1 Kişi Kartı (ResearcherCard)

Her araştırmacı için kompakt bir kart görüntülenmektedir. Kart üzerinde ad-soyad, akademik unvan, eğitim durumu, ilgi alanları (etiketler hâlinde), bağlı konu/proje sayıları ve proje türü bazlı istatistikler yer almaktadır. Kartın sol kenarında sürükleme tutamağı bulunmakta olup bu tutamak ile araştırmacı konu kartlarının üzerine sürüklenebilmektedir.

İşlem	Tetikleyici	Sonuç
Oluşturma	'+' butonu → AddItemModal	Yeni araştırmacı kaydı oluşturulur
Görüntüleme	Kart tıklama	ResearcherDetailModal açılır
Düzenleme	Modal içi form alanları	Bilgiler güncellenir → Firestore'a yazılır

İşlem	Tetikleyici	Sonuç
Silme	Modal içi 'Sil' butonu	Onay sonrası araştırmacı silinir
Sürükle-Bırak	Kart → Konu kartı üzerine	Araştırmacı konuya (ve projeye) eklenir

4.2.2 Konu Kartı (TopicCard)

Her araştırma konusu için renkli kart görüntülenmektedir. Kartın üst kısmında öncelik renk kodu (kırmızı: acil, sarı: yüksek, mavi: normal, gri: düşük) ile durum etiketi gösterilmektedir. Orta kısımda konu başlığı ve kategori etiketi, alt kısımda ise araştırmacı avatarları ve fikir sahibi göstergesi yer almaktadır. Konu kartları, sürükle-bırak hedefi olarak da işlev görmektedir.

İşlem	Tetikleyici	Sonuç
Oluşturma	'+' butonu → AddItemModal	Yeni konu oluşturulur
Düzenleme	Kart tıklama → DetailModal	Sekmeli düzenleme arayüzü açılır
Rol Atama	DetailModal → Roller sekmesi	Araştırmacıya rol ve fikir sahibi atanır
Görev Yönetimi	DetailModal → Görevler sekmesi	Görev eklenir, tamamlanır, silinir
Proje Oluşturma	Konu kartı → 'Projelendir'	Konudan yeni proje türetilir

4.2.3 Proje Kartı (ProjectCard)

Proje kartları konu kartlarına benzer yapıda olup ek bilgiler içermektedir: proje türü, başlangıç/bittiş tarihi, bütçe bilgisi ve bağlı konu sayısı. Projenin genel durumu renk koduyla, fikir sahipleri ise özel ikon ile belirtilmektedir. Proje kartlarında ayrıca 'İptal Et' butonu bulunmakta olup bu buton ile proje tamamlanmadan sonlandırılabilmektedir.

İşlem	Tetikleyici	Sonuç
Oluşturma	'+' butonu → AddItemModal	En az 1 konu seçilerek proje oluşturulur
Düzenleme	Kart tıklama → DetailModal	Proje bilgileri, bağlı konular düzenlenir
Konu Ekleme	DetailModal → Konu seçici	Projeye konu bağlanır + araştırmacılar senkronize edilir
Konu Çıkarma	DetailModal → 'Çıkar' butonu	Konu projeden çıkar; özgü araştırmacılar temizlenir
İptal Etme	Kart → 'İptal' butonu	Proje durumu 'iptal' olarak güncellenir

4.3 İstatistik ve Analitik Modülü (StatsModal)

İstatistik modalı altı sekmeden oluşmekte olup her sekme farklı bir analiz perspektifi sunmaktadır. Modal başında beş adet filtre bulunmaktadır: araştırmacı, durum, proje türü, yıl ve AÖF üyelik filtresi. Filtreler uygulandığında tüm sekmelerdeki istatistikler otomatik olarak güncellenmektedir. Grafikler SimplePieChart, SimpleBarChart ve SimpleLineChart bileşenleri ile oluşturulmaktadır.

Sekme	Açıklama
Özet	Genel sayılar, durum dağılımları, araştırmacı istatistikleri, görev tamamlama, unvan dağılımı
Araştırmacı İst.	Konulardaki/projelerdeki araştırmacı sayıları, unvan dağılımı, aktivite tablosu
Kişi Bazlı Rapor	Seçilen araştırmacının konuları, projeleri, rolleri, görev tamamlama oranı
Zaman İstatistikleri	Yıl/ay bazlı konu ve proje dağılımı, trend çizgi grafikleri, detay tablosu
Konu Bazlı	Konu durumu dağılımı, aylık başlangıç/bitiş grafikleri, detay tablosu
Proje Bazlı	Uluslararası ortaklık, proje türü dağılımı, ülke/kurum analizi, drill-down

4.3.1 Özet Sekmesi

Özet sekmesi, sistemdeki tüm verilerin kuşbakışı görünümünü sunmaktadır. Dört adet genel sayı kartında toplam araştırmacı, konu, proje ve toplam bütçe (₺) gösterilmektedir. Konu durumu dağılımı (önerilen, aktif, tamamlanan) ve proje durumu dağılımı ayrı kart gruplarıyla sunulmaktadır. Araştırmacı istatistikleri bölümünde toplam kişi, AÖF üyesi, PI deneyimli, konusu olan/olmayan ve fikir sahibi sayıları 6 kartlık ızgara düzeneinde görüntülenmektedir. Proje türü dağılımı dinamik kartlarla, görev tamamlama oranı ilerleme çubuğuyla ve unvan dağılımı (Prof.Dr., Doç.Dr., Dr.Öğr.Üyesi vb.) çubuk grafik ile gösterilmektedir. İki adet pasta grafik konu ve proje durumlarının görsel dağılımını sunmaktadır.

4.3.2 Araştırmacı İstatistikleri Sekmesi

Bu sekmede araştırmacıların konular ve projelerdeki dağılımı detaylı biçimde sunulmaktadır. Konulardaki araştırmacılar bölümünde önerilen, aktif ve tamamlanan konulardaki benzersiz araştırmacı sayıları ayrı kartlarda gösterilmektedir. Projelerdeki araştırmacılar bölümü aynı yapıyı proje bazlı sunmaktadır. Unvan dağılımı çubuk grafik ile görselleştirilmektedir. Araştırmacı Aktivite Tablosu'nda her araştırmacı için konu ve projelerdeki önerilen/aktif/tamamlanan sayıları ile görev tamamlama oranı satır satır listelenmektedir.

4.3.3 Kişi Bazlı Rapor Sekmesi

Dropdown menüden seçilen tek bir araştırmacının detaylı profili sunulmaktadır. Kişinin ismi, unvanı, kurumu ve birimi başlık alanında görüntülenmektedir. Konu kartlarında önerilen, aktif ve tamamlanan konu sayıları; proje kartlarında aynı bilgiler proje bazlı gösterilmektedir. Görev tamamlama oranı ilerleme çubuğuyla yüzdelik olarak sunulmaktadır. Rol dağılımı bölümünde kişinin farklı konulardaki rolleri renkli badge'ler ile listelenmiştir. Son bölgelerde kişinin tüm konuları (durum, başlık, rol) ve tüm projeleri (durum, başlık, tür) tablo halinde sunulmaktadır.

4.3.4 Zaman İstatistikleri Sekmesi

İki görünüm modu sunulmaktadır: yıl bazlı ve ay bazlı. Yıl bazlı görünümde sistemdeki tüm yıllar için konu ve proje dağılımları çubuk grafiklerle gösterilmektedir. Ay bazlı görünümde seçilen yılın 12 ayı için aynı metrikler hesaplanmaktadır. Her görünümde altı adet çubuk

grafik (önerilen/aktif/tamamlanan × konu/proje) ve iki adet trend çizgi grafiği (toplam konu ve toplam proje trendi) yer almaktadır. Detay tablosunda yıl/ay bazlı tüm sayılar satır satır sunulmaktadır.

4.3.5 Konu Bazlı Sekmesi

Konu durumu dağılımını pasta grafikle, konuların aylık başlangıç ve bitiş tarihlerini çubuk grafiklerle görselleştirmektedir. Detay tablosunda her durum için konu sayısı ve yüzdelik oranı sunulmaktadır.

4.3.6 Proje Bazlı Sekmesi

En kapsamlı sekmedir. Uluslararası ortak projeler bölümünde yurt dışı ortağı olan projeler tespit edilerek toplam/önerilen/aktif/tamamlanan sayıları ulusal projelerle karşılaştırılmaktadır. Proje durumu pasta grafiği ve proje türü çubuk grafiği temel dağılımları göstermektedir. Projelerin aylık başlangıç dağılımı çizgi grafikle, detay tablosunda durum/sayı/yüzde/bütçe bilgileri sunulmaktadır.

Ülke ve Kurum Dağılımı alt bölümü, projelerin coğrafi ve kurumsal analizini sunmaktadır. Dört özet kartında tekil ülke sayısı, ülke bilgili proje sayısı, tekil kurum sayısı ve işbirliği ülke sayısı gösterilmektedir. Yürütücü ülke ve ortak ülke dağılımları pasta ve çubuk grafiklerle, durum bazlı ülke dağılımı üç ayrı pasta grafikle (önerilen/aktif/tamamlanan) görselleştirilmektedir. Ülke Detay Tablosu'nda her ülke için önerilen/aktif/tamamlanan proje sayıları, yürütücü ve ortak toplamları satır satır sunulmaktadır. Aynı yapı kurum bazlı olarak da tekrarlanmaktadır: yürütücü kurum dağılımı, ortak kurum dağılımı, durum bazlı kurum dağılımı ve kurum detay tablosu. Kurum bazlı drill-down özelliği ile seçilen bir kurumun toplam proje sayısı, durum kartları, ülke dağılımı ve ortak kurum dağılımı detaylı biçimde incelenebilmektedir.

4.4 Özeti İstatistik Paneli

Ana sayfanın araştırmacı sütununda genişletilebilir/daraltılabılır bir istatistik paneli bulunmaktadır. Bu panel, 6 kartlık ızgara düzeneinde aşağıdaki metrikleri sunmaktadır:

Metrik	Açıklama
Toplam Kişi	Sistemde kayıtlı tüm araştırmacı sayısı
AÖF Üyesi	Açıköğretim Fakültesi mensubu araştırmacı sayısı
PI Deneyimli	Proje yürütücülüğü (PI) deneyimi olan araştırmacı sayısı
Konusu Olan	En az bir araştırma konusuna atanmış araştırmacı sayısı
Konusu Olmayan	Henüz herhangi bir konuya atanmamış araştırmacı sayısı
Fikir Sahibi	En az bir konuda veya projede fikir sahibi olarak işaretlenmiş araştırmacı sayısı

Proje türü dağılımı da ayrı bir alt bölümde dinamik ızgara olarak gösterilmektedir. Her proje türü için adet bilgisi renkli kartlar ile sunulmaktadır.

4.5 Diğer Uygulama Özellikleri

4.5.1 Sürükle-Bırak (Drag & Drop) Sistemi

HTML5 Drag and Drop API'si üzerine inşa edilen bu sistem, araştırmacı kartlarının konu kartları üzerine sürüklentimesine olanak tanımaktadır. Bırakma gerçekleştiğinde RoleSelectPopup açılmakta; bu popup'ta araştırmacının konudaki rolü ve fikir sahibi olup

olmadığı belirlenmektedir. Seçim tamamlandığında araştırmacı hem konuya hem de bağlı projelere otomatik olarak eklenmektedir.

4.5.2 Tablo Görünümü (TableViewModal)

Kart görünümüne alternatif olarak tablo biçiminde veri görüntüleme seçeneği sunulmaktadır. Üç sekmeden oluşmaktadır (araştırmacılar, konular, projeler). Tabloda sıralama, filtreleme ve CSV dışa aktarım özellikleri bulunmaktadır. Konu tablosunda 'Fikir Sahibi' sütunu yer almaktadır.

4.5.3 Sıralama Tablosu (LeaderboardModal)

Araştırmacıları çeşitli kriterlere göre sıralayan ve puanlayan bu modal, konu sayısı, proje katılımı ve görev tamamlama oranı gibi metrikleri hesaplayarak bir liderlik tablosu oluşturmaktadır. AÖF üyelik filtresi ve sıralama kriteri seçimi desteklenmektedir.

4.5.4 Takvim (CalendarModal)

Proje zaman çizelgelerini aylık takvim görünümünde sunan bu modal, yaklaşan son tarihleri vurgulamakta ve proje başlangıç/bittiş tarihlerini görsel olarak işaretlemektedir. Tarihe tıklandığında ilgili projenin detaylarına erişilebilmektedir.

4.5.5 Hızlı Erişim Bağlantıları (QuickLinksPanel)

Yöneticilerin sık başvurduğu dış kaynaklara (TÜBİTAK portalı, YÖK veritabanı vb.) tek tıkla erişim sağlayan panel, Firestore'da saklanmakta ve tüm kullanıcılar tarafından görüntülenebilmektedir. Bağlantılar eklenebilmekte, düzenlenebilmekte ve silinebilmektedir.

4.5.6 Son Tarih Paneli (DeadlinePanel)

Zil ikonuna tıklandığında açılan dropdown menü, yaklaşan proje son tarihlerini listelemektedir. Tarih hesaplaması otomatik olarak yapılmakta ve kalan gün sayısına göre renk kodu atanmaktadır (kırmızı: acil, sarı: yakın, yeşil: uzak).

4.5.7 Kural Tabanlı Chatbot (ArGeChatbot)

Dashboard bünyesinde basit bir kural tabanlı sohbet asistanı bulunmaktadır. Bu asistan, araştırma konusu önerisi ve araştırmacı eşlestirmesi gibi görevlerde kullanıcıya yardımcı olmaktadır. LLM tabanlı değil, deterministik kurallarla çalışmaktadır.

4.5.8 Ayarlar Paneli (SettingsModal)

Yapılabilir parametrelerin yönetildiği bu panel; roller, konu/proje durumları, öncelik seviyeleri, konu kategorileri, proje türleri, akademik dereceler ve eğitim durumlarını kapsamaktadır. Her yapılandırma öğesi eklenebilmekte, düzenlenebilmekte ve silinebilmektedir. Değişiklikler Firestore'a anında yansıtılmakta ve tüm bağlı istemcilerde gerçek zamanlı olarak güncellenmektedir.

4.6 Yetkilendirme ve Rol Sistemi

Sistem, dört kademeli bir yetkilendirme modeli uygulamaktadır. Her kullanıcı rolü, belirli işlemleri gerçekleştirme yetkisine sahiptir. Bu model, kurumsal hiyerarşiyi yansıtacak şekilde tasarlanmıştır.

Rol	Yetkiler
master	Tüm işlemler + yedekleme + sistem ayarları + zorunlu yayına

Rol	Yetkiler
admin	Tüm CRUD işlemleri + ayar düzenleme + manuel senkronizasyon
editor	Ekleme ve düzenleme yetkileri, silme kısıtlı
user	Yalnızca görüntüleme (salt okunur erişim)

5. Senkronizasyon Mimarisi

Senkronizasyon, bu projenin en kritik teknik bileşenidir. Birden fazla yöneticinin aynı anda sistemi kullanabilmesi, veri çakışmalarının önlenmesi ve değişikliklerin anlık olarak tüm istemcilere yansıtılması gerekmektedir (Google, 2024). Bu bölümde, Firebase Firestore tabanlı gerçek zamanlı senkronizasyon mekanizması detaylandırılmaktadır.

5.1 Okuma Mekanizması (onSnapshot)

Her 11 doküman için bir onSnapshot dinleyicisi çalışmaktadır. Gelen snapshot'taki JSON, lastJson.current referansı ile karşılaştırılmakta; farklısa state güncellenmekte, aynıysa işlem atlanmaktadır. Bu karşılaştırma mekanizması, sonsuz yaz-oku döngüsünü engellemenin temel anahtarıdır. lastJson.current referansı, hem yazma hem okuma tarafında güncellenerek iki yönlü koruma sağlamaktadır.

5.2 Yazma Mekanizması (Debounced setDoc)

State değiştiğinde 500ms debounce süresi sonrasında Firestore'a setDoc ile yazılmaktadır. Yazma öncesinde lastJson.current güncellenerek onSnapshot'in kendi yazdığı veriyi yeniden işlemesi engellenmektedir. Debounce mekanizması, ardışık düzenlemelerde gereksiz yazma operasyonlarını birleştirerek Firestore kota kullanımını optimize etmektedir.

5.3 Çapraz Varlık Senkronizasyonu

Konu, proje ve araştırmacı varlıkları arasında çift yönlü senkronizasyon sağlanmıştır. handleUpdateItem fonksiyonu, bir konu güncellenirken eski ve yeni araştırmacı listelerini karşılaştırarak eklenen ve çıkarılan araştırmacıları tespit etmektedir. Çapraz senkronizasyonun doğruluğu, 9 farklı kontrol noktası üzerinden denetlenmiş ve tüm noktalar güvenli olarak teyit edilmiştir.

5.4 Çevrimiçi Kullanıcı Takibi (Presence)

Google Docs tarzı bir canlı gösterge sistemi uygulanmıştır. Her oturum açık kullanıcı, Firestore'daki presence dokümanına kaydedilmekte ve tüm istemcilerde renkli avatarlar ile görüntülenmektedir. 6 farklı renk paleti kullanılarak eş zamanlı çalışan yöneticiler birbirinden ayırt edilebilmektedir.

5.5 Özel Firestore Konfigürasyonu

experimentalForceLongPolling: true — Kurumsal ağlarda güvenlik duvarları tarafından engellenen WebSocket bağlantıları yerine HTTP long polling kullanılmaktadır.

memoryLocalCache() — IndexedDB tabanlı önbelleğin neden olduğu kuyruk sorunları ortadan kaldırılmaktadır.

6. Yapay Zekâ Destekli Geliştirme Süreci

Bu bölüm, projenin geliştirilmesinde kullanılan yapay zekâ destekli metodolojileri, insan-YZ işbirliğinin somut uygulamalarını ve bu süreçten elde edilen çıkarımları detaylandırmaktadır (Anthropic, 2025).

6.1 İnsan + YZ Kodlama Akış Diyagramı

Aşağıda, her bir özelliğin geliştirilmesinde izlenen iteratif İnsan + YZ kodlama süreci adım adım gösterilmektedir. Bu akış, çevik yazılım geliştirme metodolojileri ile benzerlikler taşımakta; ancak YZ ajanının anlık kod üretme kapasitesi sayesinde çok daha hızlı iterasyonlara olanak tanımaktadır (Beck vd., 2001; Karpathy, 2025).

ADIM 1 — Gereksinim Bildirimi (İnsan)

İnsan geliştirici, istediği özelliği Türkçe doğal dilde açıklamaktadır.

Örnek: 'Kişi kartlarına proje türü istatistiklerini ekle'

↓

ADIM 2 — Kod Analizi ve Plan (YZ Ajanı)

YZ ajanı mevcut kodu okumakta, etkilenen dosyaları belirlemekte ve değişiklik planını sunmaktadır.

Çıktı: Etkilenen bileşenler, eklenmesi/değiştirilmesi gereken satırlar

↓

ADIM 3 — Kod Üretimi (YZ Ajanı)

YZ ajanı planı uygulayarak kod değişikliklerini gerçekleştirermektedir.

React bileşenleri, Firestore entegrasyonu, Tailwind CSS stilleri üretilmektedir.

↓

ADIM 4 — Doğrulama (YZ Ajanı)

Üretilen kod Babel parser ile ayırtırılarak söz dizimi doğrulanmaktadır.

Çıktı: 'PARSE OK' veya hata detayı

↓

ADIM 5 — İnceleme ve Test (İnsan)

İnsan geliştirici kodu incelemekte, tarayıcıda test etmekte ve geri bildirim vermektedir.

↓ (Düzelme gerekiyorsa ADIM 1'e dönülmektedir)

ADIM 6 — Onay ve Dağıtım (İnsan + YZ)

Özellik onaylandığında Firebase Hosting'e dağıtılmaktadır.

Rapor otomatik olarak güncellenmektedir.

6.2 İnsan ve YZ Rol Dağılımı

Görev Alanı	İnsan Geliştiricinin Rolü	YZ Ajanının Rolü
Gereksinim	Özellikin doğal dilde tanımaması	Gereksinimleri koda dönüştürmek
Mimari	Teknoloji seçimi, veri modeli kararları	Mevcut kodu analiz edip plan sunmak
Kod Üretimi	Üretilen kodu incelemek, onaylamak	React, Firestore, CSS kodlarını yazmak

Görev Alanı	İnsan Geliştiricinin Rolü	YZ Ajanının Rolü
Hata Ayıklama	Sorunları raporlamak, test etmek	Kök neden analizi ve çözüm uygulamak
UI/UX	Renk, düzen, etkileşim kararları	Tailwind CSS ile uygulamak
Kalite Kontrol	Tarayıcıda son kullanıcı testi	Kod审计, syntax doğrulama
Dokümantasyon	İçerik ve dil kontrolü	Rapor yazımı (Text, DOCX, PDF)

6.3 YZ Kullanım Alanları Tablosu

Alan	YZ Türü	Açıklama
Kod Üretimi	LLM Agent (Claude)	React, Firestore, Tailwind kodları生成済み
Hata Ayıklama	LLM Agent (Claude)	Closure bug, sync döngüsü çözümlenmiştir
Kod Denetimi	LLM Agent (Claude)	Çapraz senkronizasyon审计が実行済み
UI Tasarımı	LLM (Vibe Coding)	Kart, modal, tablo bileşenleri tasarlanmıştır
Dokümantasyon	LLM Agent (Claude)	Bu rapor YZ desteğiyle hazırlanmıştır
İş Mantığı	Kural Tabanlı	Validasyonlar, iş kuralları deterministik
Auth Sistemi	Kural Tabanlı	Şifre kontrolü, rol atama kurallarla çalışmaktadır
Chatbot	Kural Tabanlı	Sohbet asistanı if-then kurallarıyla çalışmaktadır

7. Kritik Hata Çözümleri ve Teknik Dersler

Geliştirme sürecinde karşılaşılan kritik hatalar, hem sistemin güvenilirliğini test etmiş hem de yapay zekâ destekli geliştirme sürecinin hata ayıklama kapasitesini somut biçimde ortaya koymuştur. Bu bölümde temel hatalar, kök neden analizleri ve çözüm yaklaşımları sunulmaktadır.

7.1 Initial Push Closure Bug'ı (Veri Kaybı)

Bu hata, projenin en kritik ve öğretici teknik sorunudur. JavaScript'in closure mekanizmasının React hooks ile etkileşiminden kaynaklanan bu hata, üretim ortamında gerçek veri kaybına yol açmıştır. useEffect(() => {...}, []) içindeki markReady callback'i, JavaScript closure mekanizması gereği mount anındaki boş state değerlerini yakalamıştır. Bu değerler Firestore'a yazılıncı gerçek verinin üzerine geçmiştir. Çözüm olarak tüm initial push kodu kaldırılmıştır; onSnapshot dinleyicisi zaten doküman yoksa varsayılan değerleri yazmaktadır.

7.2 Sonsuz Yaz-Oku Döngüsü

onSnapshot ile okunan verinin state'e yazılması, state değişikliğinin debounced write'i tetiklemesi ve yazılan verinin onSnapshot'ı tekrar tetiklemesi şeklinde sonsuz bir döngü oluşmuştur. lastJson.current karşılaştırma patterni ile onSnapshot'ın kendi yazdığı veriyi tekrar işlemesi engellenmiştir. Bu pattern, gerçek zamanlı sistemlerde sıkça karşılaşılan yapısal bir sorunun çözümünü temsil etmektedir.

7.3 handleUpdateItem Yeniden Yazımı

Çapraz varlık senkronizasyonundaki eksiklikler, bir konudan çıkarılan araştırmacının bağlı projelerde kalmaya devam etmesi sorununa yol açmıştır. Eski ve yeni konu araştırmacı listeleri karşılaştırılarak tam yeniden yazım gerçekleştirilmiştir. Bu hata, ilişkisel veri modellerinde kaskad güncellemenin önemini somut biçimde ortaya koymuştur.

7.4 Firestore Bağlantı Sorunları

Sorun	Kök Neden	Çözüm
Kota Aşımı	Spark planı okuma/yazma limitleri Firebase Blaze planına yükseltilmiştir	
IndexedDB Kuyruk	Varsayılan önbellek birikmesi	memoryLocalCache() uygulanmıştır
WebSocket Engeli	Üniversite güvenlik duvarı	experimentalForceLongPolling uygulanmıştır

8. İş Kuralları, Yedekleme ve Kalite Güvencesi

8.1 Temel İş Kuralları

Proje-Konu Zorunlu İlişki: Her projenin en az bir konuya bağlı olması zorunludur. Projenin tek konusu çıkarılamamakta; bunun yerine 'Önce projeyi iptal edin' uyarısı gösterilmektedir.

Araştırmacı Temizlik Kuralları: Konu projeden çıkarıldığında, o konuya özgü araştırmacılar projeden otomatik olarak temizlenmektedir. Başka bağlı konularda bulunan araştırmacılar korunmaktadır.

Fikir Sahibi Takibi: Her konu ve projede bir veya daha fazla araştırmacı 'fikir sahibi' olarak işaretlenebilmektedir. Bu bilgi, konu-proje senkronizasyonuna dâhildir ve otomatik olarak taşınmaktadır.

8.2 Yedekleme Sistemi (Üç Katmanlı)

Katman 1 — JSON İndir/Yükle: Tüm veriler JSON formatında yerel bilgisayara indirilebilmektedir. Bu yedek, çevrimdışı veri kurtarma imkânı sağlamaktadır.

Katman 2 — Firestore Yedekleme: Veriler Firestore'da ayrı yedek koleksiyonuna kopyalanmaktadır (yalnızca master rolü).

Katman 3 — Otomatik Hatırlatma: 30 günden uzun süre yedekleme yapılmamışsa sarı uyarı banner'ı gösterilmektedir.

8.3 Doğrulama ve Kalite Güvencesi

Veri kaybı riskinin sistematik biçimde denetlenmesi amacıyla kapsamlı bir kod审计i gerçekleştirilmiştir. Bu audit, yapay zekâ ajanı tarafından 9 farklı kontrol noktası üzerinden yürütülmüş ve tüm noktalar güvenli olarak teyit edilmiştir.

#	Kontrol Noktası	Durum	Sonuç
1	Initial Push kodu	Kaldırılmıştır	GÜVENLİ
2	markReady fonksiyonu	Salt okunur	GÜVENLİ
3	onSnapshot dinleyicisi	LastJson kontrolü	GÜVENLİ
4	Debounced write	500ms gecikme	GÜVENLİ
5	handleUpdateItem	Tam yeniden yazım	GÜVENLİ
6	handleDeleteTopic	Temizlik eklenmiştir	GÜVENLİ
7	handleRemoveTopicFromProject	Temizlik eklenmiştir	GÜVENLİ
8	handleRoleSelect	Proje sync eklenmiştir	GÜVENLİ
9	setState çağrıları	Boş dizi yazımı yoktur	GÜVENLİ

9. Sonuç ve Tartışma

9.1 Gerçekleştirilen Çalışmanın Özeti

Bu çalışmada, Anadolu Üniversitesi Açıköğretim Fakültesi'nin Ar-Ge faaliyetlerini merkezi olarak yönetmek üzere kapsamlı bir web tabanlı dashboard geliştirilmiştir. Sistem; araştırmacı, konu ve proje yönetimini, gerçek zamanlı çoklu kullanıcı senkronizasyonunu, çapraz varlık tutarlığını, çok katmanlı yedekleme mekanizmalarını, istatistik ve analistik araçlarını kapsamaktadır. Yaklaşık 7.650 satır React kodundan oluşan uygulama, 20'den fazla bileşen, 11 Firestore dokümanı ve 4 kademeli yetkilendirme modeli ile donatılmıştır.

Projenin en özgün boyutu, geliştirme sürecinin tamamında yapay zekâ ajanlarının aktif katılımcı olarak yer almazıdır. Claude Opus 4 modeli; kod üretiminden hata ayıklamaya, çapraz senkronizasyon auditinden bu raporun yazılmasına kadar geliştirme sürecinin her aşamasında etkin rol üstlenmiştir. Bu deneyim, 'vibe coding' kavramının gerçek dünya koşullarında nasıl uygulanabileceğine dair somut ve detaylı bir vaka çalışması niteliğindedir.

9.2 Alana Katkı

Bu çalışma, birden fazla disiplinin kesim noktasında önemli katkılar sunmaktadır. Yükseköğretimde Ar-Ge yönetimi alanında, dağınık ve senkronize olmayan veri yönetimi sorununa bulut tabanlı gerçek zamanlı bir çözüm geliştirilmiştir. Yapay zekâ destekli yazılım geliştirme alanında ise 'vibe coding' metodolojisinin 7.650+ satırlık karmaşık bir uygulamada başarıyla kullanılabileceği gösterilmiştir.

Çalışmanın özellikle önemli bir katkısı, LLM tabanlı YZ ajanının yalnızca basit kod üretimi değil, aynı zamanda karmaşık hata ayıklama (closure bug gibi), çapraz modül denetimi ve sistematik kalite güvencesi görevlerini de başarıyla yerine getirebildiğinin gösterilmesidir. Bu bulgu, yazılım mühendisliğinde YZ'nin rolüne ilişkin güncel tartışmalara empirik bir katkı sağlamaktadır.

Kural tabanlı ve LLM tabanlı yaklaşımın aynı projede farklı katmanlarda birlikte kullanılması — uygulamanın iç mantığında deterministik kurallar, geliştirme sürecinde ise üretken YZ — hibrit bir model ortaya koymaktadır. Bu model, güvenilirlik ve esneklik arasındaki dengenin nasıl sağlanabileceğine dair pratik bir referans oluşturmaktadır.

9.3 Sınırlılıklar ve Kısıtlar

Çalışmanın bazı sınırlılıkları bulunmaktadır. Monolitik dosya mimarisi (tek dosyada ~7.650 satır), bakım ve genişletme açısından zorluklar yaratabilmektedir. Kimlik doğrulama sistemi basit parola tabanlı olup kurumsal güvenlik standartları için yeterli olmayabilmektedir. Ayrıca vibe coding sürecinin tekrarlanabilirliği ve farklı projelere uyarlanabilirliği henüz sistematik olarak incelenmemiştir. Bu sınırlılıklar, gelecek çalışmalar için iyileştirme alanlarını işaret etmektedir.

9.4 Gelecek Yönelimler

- Modüler Yapıya Geçiş: Dashboard.jsx'in daha küçük, yeniden kullanılabilir bileşenlere bölünmesi
- Firebase Authentication: SSO, OAuth ve çok faktörlü doğrulama desteği
- Yayın Takip Modülü: Araştırmacıların akademik yayınlarının izlenmesi, DOI entegrasyonu
- Otomatik Raporlama: İstatistik raporlarının periyodik olarak PDF/Excel formatında üretimi

- Bildirim Sistemi: E-posta ve push notification ile değişiklik bildirimi
- LLM Entegrasyonu: Araştırma eğilimi analizi, işbirliği önerisi, kaynak tahsis optimizasyonu

9.5 Kapanış

Yapay zekâ destekli yazılım geliştirme, henüz erken aşamalarında olmasına karşın hızla olgunlaşan ve dönüştürücü potansiyel taşıyan bir alan olarak öne çıkmaktadır. Bu proje, bir üniversitenin gerçek bir operasyonel ihtiyacını karşılamak üzere, insan geliştiricinin vizyonu ile yapay zekâ ajanının teknik kapasitesinin bir araya getirilmesiyle hayatı geçirilmiştir. Sonuç olarak ortaya çıkan ürün, yalnızca bir yönetim aracı değil; aynı zamanda insan-YZ işbirliğinin yazılım mühendisliğindeki olanaklarını sergileyen uygulamalı bir vaka çalışmasıdır.

Yükseköğretim kurumlarının dijital dönüşüm süreçlerinde yapay zekâ destekli araçların ve metodolojilerin benimsenmesi, hem kurumsal verimlilik hem de akademik yenilikçilik açısından büyük potansiyel taşımaktadır. Bu çalışma, söz konusu potansiyelin somut bir örneği olarak alana katkı sağlamayı amaçlamaktadır. Gelecekte bu deneyimlerden yola çıkarak, vibe coding metodolojisini daha geniş kapsamlı projelerde sistematik olarak incelenmesi ve kurumsallaşması hedeflenmektedir.

Bu rapor 28 Şubat 2026 tarihinde güncellenmiştir. Raporun en güncel hâli her sistem güncellemesinde Text (.md), Word (.docx) ve PDF formatlarında yeniden üretilmektedir.

Kaynakça ve Referanslar

Aşağıdaki kaynaklar APA 7 (American Psychological Association, 7. baskı) formatına uygun olarak listelenmiştir. Metin içi atıflarda yazar soyadı ve yıl bilgisi kullanılmıştır.

- Altbach, P. G., Reisberg, L., & Rumbley, L. E. (2019). *Trends in global higher education: Tracking an academic revolution*. Brill. <https://doi.org/10.1163/9789004406155>
- Anthropic. (2025). *Claude: AI assistant technical documentation*.
<https://docs.anthropic.com/en/docs/about-claude/models>
- Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). *Manifesto for agile software development*. <https://agilemanifesto.org/>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877-1901. <https://doi.org/10.48550/arXiv.2005.14165>
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. O., Kaplan, J., ... & Zaremba, W. (2021). Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
<https://doi.org/10.48550/arXiv.2107.03374>
- Google. (2024). *Cloud Firestore documentation*. <https://firebase.google.com/docs/firestore>
- Karpathy, A. (2025, Şubat 4). *Vibe coding* [Blog yazısı]. X (Twitter).
<https://x.com/karpathy/status/1886192184808149383>
- Meta. (2024). *React documentation*. <https://react.dev/>
- Russell, S. J., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4. baskı). Pearson.
<https://aima.cs.berkeley.edu/>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 5998-6008. <https://doi.org/10.48550/arXiv.1706.03762>
- Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., ... & Wen, J. (2024). A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6), 186345. <https://doi.org/10.1007/s11704-024-40231-1>
- YÖK. (2023). *Yükseköğretim kurumları araştırma ve geliştirme faaliyetleri istatistikleri*. Yükseköğretim Kurulu. <https://www.yok.gov.tr/universitelerimiz/istatistikler>
- Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education. *International Journal of Educational Technology in Higher Education*, 16(1), 39. <https://doi.org/10.1186/s41239-019-0171-0>