



softITo 4. DÖNEM 1. GRUP
YAZILIM VE VERİTABANI UZMANLIĞI
BİTİRME PROJESİ RAPORU

SnapMatch

Etkinlik Fotoğraf Yükleme, QR Erişim ve Yapay Zekâ Yüz Eşleştirme Paneli

Ezgi Kuzu

Furkan Yorulmaz

Merve Daşçi

Okan Şahin

Berna Nur İlbaş

Temmuz, 2026

softITo Yazılım Bilişim Akademisi, İstanbul

İÇİNDEKİLER

1. GİRİŞ VE STRATEJİK GENİŞLEME	Page 3
1.1. Projenin Amacı ve Vizyoner Hedefleri	Page 3
1.2. Problem Tanımı ve Geleneksel Çözümlerin Sınırları	Page 4
1.3. Çözüm Yaklaşımı ve Teknolojik Yenilik Değeri	Page 4
1.4. Benchmark Analizi: Photier vs. SnapMatch	Page 5
1.5. Proje Kısıtları, Varsayımlar ve MVP Sınırları	Page 6
2. MATERYAL VE YÖNTEM	Page 7
2.1. Genişletilmiş Teknoloji Yığını ve Seçim Kriterleri	Page 7
2.2. Dağıtık ve Modüler Sistem Mimarisi	Page 9
2.3. Çoklu Workspace ve Multi-Tenant Veri İzolasyonu	Page 10
2.4. Kullanıcı Roller ve Erişim Kontrol Matrisi (RBAC)	Page 11
2.5. Detaylı Veri Modeli ve PostgreSQL ER Diyagram Analizi	Page 12
2.6. REST API Tasarımı ve Veri Değişim Sözleşmeleri	Page 14
3. GELİŞTİRME VE UYGULAMA	Page 16
3.1. Kimlik Doğrulama ve Güvenli Token Yönetimi	Page 16
3.2. Etkinlik Yönetimi ve Yaşam Döngüsü Modülü	Page 17
3.3. QR Kod Üretimi ve Dinamik Misafir (Guest) Akışı	Page 18
3.4. Çoklu Dosya Yükleme, Kuyruk Yapısı ve Sharp Görüntü Sıkıştırma	Page 19
3.5. Üç Boyutlu WebGL (Three.js) Tabanlı Spiral Galeri Geliştirilmesi	Page 20
3.6. Katılımcı Yüz Kaydı ve Kamera API Entegrasyonları	Page 21
3.7. face-api.js ile Yapay Zekâ Yüz Tanıma ve Embedding Pipeline'ı	Page 22
3.8. Eşleşme Değerlendirme (Match Review) ve Güven Skoru Yönetimi	Page 23
3.9. KVKK / GDPR Uyum Stratejileri, Veri Minimizasyonu ve Silme Hakkı	Page 24
3.10. Raporlama, İstatistiksel İzleme ve Analytics Kontrol Paneli	Page 25
3.11. Test Otomasyonu, Kalite Standartları ve Dağıtım	Page 26
3.12. Uçtan Uca Final Demo ve Doğrulama Senaryoları	Page 27
4. SONUÇ VE GELECEK YOL HARİTASI	Page 28
5. KAYNAKÇA	Page 29

1. GİRİŞ VE STRATEJİK GENİŞLEME

1.1. Projenin Amacı ve Vizyoner Hedefleri

Gelişen dijital teknolojiler, sosyal etkileşimlerin ve fiziksel organizasyonların dijital ortamlarda belgelenmesini kaçınılmaz kılmıştır. Mezuniyet törenleri, kurumsal konferanslar, spor etkinlikleri, festivaller ve özellikle düğünler gibi geniş katılımlı fiziksel organizasyonlarda profesyonel fotoğrafçılar tarafından binlerce kare fotoğraf çekilmektedir. Ancak bu fotoğrafların, ilgili katılımcılara hızlı, verimli ve güvenli bir şekilde ulaştırılması günümüzün en büyük operasyonel tıkanıklıklarından biridir.

SnapMatch (Event AI Photo Finder), bu ihtiyacı uçtan uca karşılamak amacıyla geliştirilmiş, çok kiracılı (multi-tenant) bir Micro-SaaS (Yazılım Hizmeti) platformudur. Projenin temel hedefi, fiziksel etkinliklerle dijital dağıtım kanalları arasındaki uçurumu kapatarak, katılımcıların yalnızca bir kare "selfie" (yüz kaydı) yükleyerek binlerce etkinlik fotoğrafı arasından kendi yüzlerinin geçtiği tüm kareleri yapay zekâ desteğiyle anında bulmalarını sağlamaktır.

SaaS mimarisinin getirdiği avantajlar sayesinde, bağımsız fotoğrafçılardan kurumsal etkinlik ajanslarına kadar geniş bir yelpazedeki hizmet sağlayıcılar (kiracılar / tenants) platform üzerinde kendi bağımsız çalışma alanlarını (workspaces) tanımlayabilmektedir. Bu sayede platform, yalnızca tek bir etkinlik için çalışan bir araç değil; ölçeklenebilir, kendi gelir modelini üretebilen ve sürdürülebilir bir iş ekosistemine dönüşmektedir.

Vizyon Bildirisi: SnapMatch, etkinliklerin ardından yaşanan "fotoğraf arama, paylaşma ve ulaştırma" karmaşasını ortadan kaldırarak; en gelişmiş yapay zekâ yüz tanıma algoritmalarını kullanıcı dostu, mobile-first bir deneyimle birleştiren küresel standartlarda bir platform olmayı hedefler.

1.2. Problem Tanımı ve Geleneksel Çözümlerin Sınırları

Mevcut etkinlik sektörü incelendiğinde, çekilen fotoğrafların katılımcılarla paylaşılmasında yaygın olarak kullanılan yöntemlerin ciddi dezavantajlar ve operasyonel maliyetler barındırdığı görülmektedir:

- **Genel Bulut Sürücüler (Google Drive, Dropbox, OneDrive):** Fotoğrafçılar etkinlik sonrasında tüm arşivleri bu platformlara yükleyerek genel bağlantılar paylaşırlar. This yöntem, her katılımcının binlerce fotoğrafı tek tek manuel olarak taramasını gerektirir. Bu durum son derece zaman alıcıdır ve kullanıcı deneyimini baltalar.
- **KVKK ve Gizlilik İhlalleri:** Herkese açık bulut klasörleri, tüm katılımcıların fotoğraflarının başkaları tarafından görünmesine, indirilmesine ve izinsiz paylaşılmasına olanak tanır. Bu durum, Türkiye'de 6698 sayılı Kişisel Verilerin Korunması Kanunu (KVKK) ve Avrupa Birliği'nde GDPR kapsamında ciddi hukuki yaptırımlara yol açabilecek niteliktedir.
- **Düşük Marka Etkileşimi:** Fotoğrafların teslimi genellikle etkinlikten günler veya haftalar sonra gerçekleşir. Bu gecikme, etkinliğin oluşturduğu heyecan ve sosyal medya paylaşım potansiyelinin sönmesine neden olur.
- **Pazarlama ve Dönüşüm Kayıpları:** Fotoğrafçılar, doğrudan katılımcılarla bağ kuramadıkları için ek hizmet satışı veya marka bilinirliği yaratma fırsatlarını kaybederler.

1.3. Çözüm Yaklaşımı ve Teknolojik Yenilik Değeri

SnapMatch bu problemi şu akışla çözmektedir: Fotoğrafçı veya organizasyon ekibi web panel üzerinden etkinlik oluşturur, QR kod üretir og toplu yükler. Katılımcılar QR okutarak selfie/yüz kaydı yapar; yapay zekâ destekli eşleştirme sistemi her katılımcının fotoğraflarını otomatik olarak bulur ve güven skoru ile sunar. Düşük güvenli eşleşmeler, fotoğrafçı tarafından manuel olarak onaylanır veya reddedilir.

1.4. Benchmark Analizi: Photier vs. SnapMatch

Platformun konumlandırmasını doğru yapabilmek amacıyla pazardaki en güçlü rakip olan Photier ürünü teknik ve işlevsel açıdan detaylıca incelenmiştir. Yapılan karşılaştırma analizi, geliştirilen MVP'nin inovasyon noktalarını ortaya koymaktadır:

Özellik / Kriter	Photier (Sektör Standardı)	SnapMatch (Geliştirilen MVP)	Stratejik Fark ve Katma Değer
Erişim Modeli	Native Mobil Uygulama + Web Paneli	Mobile-First Progressive Web (PWA Uyumlu)	Uygulama mağazası bariyerini kaldırarak etkinlik anında katılım oranını %90'ın üzerine çıkarır.
Yüz Tanıma Altyapısı	Ticari Bulut API'leri (AWS Rekognition / Azure Face API)	Açık Kaynak face-api.js & Canvas Tabanlı Yerel Pipeline	Büyük ölçekli etkinliklerde sorgu başına maliyeti sıfırlayarak SaaS karlılık oranını maksimize eder.
Dosya Yükleme og İşleme	Klasik Dosya Transfer Protokolleri	Presigned URL & BullMQ Tabanlı Asenkron İş Kuyruğu	Yüksek çözünürlüklü toplu yüklemelerde tarayıcıyı kilitlemeden arka planda kesintisiz işleme sağlar.
Kullanıcı Arayüzü & Galeri	Standart Grid Listeleme ve Filtreleme	Three.js Tabanlı WebGL 3B Spiral Galeri Entegrasyonu	Katılımcıya oyunlaştırılmış (gamified) ve yüksek etkileşimli bir görsel deneyim sunar.
Veri İzolasyonu	Klasik İlişkisel Şema Tasarımı	Çok Kiracılı (Multi-Tenant) Workspace Ayrımı	Her fotoğrafçı veya ajansın kendi alt alan adı ve bağımsız veri tabanları varmış gibi izole çalışmasını sağlar.

1.5. Proje Kısıtları, Varsayımlar ve MVP Sınırları

SnapMatch projesi, softITo yazılım akademisi bünyesinde kısıtlı bir zaman diliminde geliştirildiğinden, projenin ilk sürümü (MVP) belirli sınırlar ve kısıtlar altında kurgulanmıştır. Bu kısıtlar, projenin en yüksek kalitede çalışmasını sağlamak üzere titizlikle yönetilmiştir:

1.5.1. Teknik ve Mimari Kısıtlar

- **Yerel Yüz Tanıma:** Yüz tanıma süreçlerinde ticari bulut sağlayıcılarının (Amazon, Google, Microsoft) ücretli servisleri yerine, sunucu maliyetini sıfırlamak ve bağımsızlığı korumak amacıyla açık kaynak kodlu face-api.js kütüphanesi kullanılmıştır. Görüntüler sunucu tarafında asenkron olarak Node.js worker'ları içinde işlenmektedir.
- **Dosya Boyut Sınırları:** MVP kapsamında, sunucu bellek (RAM) og bant genişliği sınırları göz önünde bulundurularak, fotoğrafçıların tek seferde yükleyebileceği maksimum fotoğraf boyutu 15 MB, katılımcıların selfie boyutları ise 5 MB ile sınırlandırılmıştır.
- **Watermark (Filigran) ve Ödeme Entegrasyonları:** Fotoğrafların üzerine dinamik logo basılması (watermarking) ve katılımcıların fotoğrafları yüksek çözünürlüklü indirmek için ödeme yapmasını sağlayan sanal POS entegrasyonları projenin 2. fazına (Faz-2) ertelenmiştir.

1.5.2. Proje Başarı Kriterleri (KPI'lar)

Geliştirilen MVP sürümünün başarı kriterleri, gerçekçi saha testlerine dayanacak şekilde aşağıdaki nicel hedeflerle tanımlanmıştır:

1. **Toplu Yükleme Kapasitesi:** Tek bir etkinlik altında en az 50 adet yüksek çözünürlüklü fotoğrafın (ortalama 8 MB) sürükle-bırak yöntemiyle kesintisiz yüklenmesi ve optimize edilmesi.
2. **Katılımcı Kayıt Başarısı:** En az 5 farklı katılımcının mobil cihazlar üzerinden rıza metnini onaylayarak başarılı şekilde yüz kaydı oluşturabilmesi.
3. **Eşleştirme Doğruluğu og Süresi:** Yüklenen 50 fotoğraf içindeki yüzlerin, rıza veren katılımcılarla %85'in üzerinde doğruluk oranıyla eşleştirilmesi og eşleşme işleminin fotoğraf başına ortalama 1.5 saniyenin altında tamamlanması.

2. MATERYAL VE YÖNTEM

2.1. Geniştirilmiş Teknoloji Yığını ve Seçim Kriterleri

SnapMatch platformunun geliştirilmesinde, modern yazılım mühendisliği prensiplerine uygun, yüksek performanslı, asenkron süreçleri destekleyen ve ölçeklenebilir teknolojiler tercih edilmiştir. Seçilen teknolojilerin mimari gereksinimleri aşağıda detaylandırılmıştır:

2.1.1. Frontend Teknolojileri

- **React 19 & Vite:** React'in en güncel sürümü olan React 19, geliştirilmiş state yönetimi ve Concurrent Rendering özellikleri sunmaktadır. Bu durum, özellikle yüzlerce yüksek çözünürlüklü fotoğrafın listelendiği dinamik galerilerde tarayıcı performansının korunmasını sağlar. Vite ise Webpack'e kıyasla çok daha hızlı derleme süreleri sunarak geliştirici üretkenliğini artırmıştır.
- **Redux Toolkit & React Router DOM 7:** Uygulama genelinde kullanıcı oturum durumları, aktif workspace bilgileri ve global bildirimler Redux Toolkit ile yönetilir. React Router DOM 7, rol bazlı rota koruması (Route Guarding) ve dinamik parametre geçişleri için entegre edilmiştir.
- **Tailwind CSS 4 & Framer Motion:** Tailwind CSS 4, modern CSS standardslarını destekleyen yapısı sayesinde son derece hızlı ve hafif stil tanımlamalarına olanak tanır. Framer Motion ise kullanıcı geçişlerinde, modal pencerelerde ve yükleme animasyonlarında akıcı görsel geri bildirimler sağlamak amacıyla kullanılmıştır.
- **Three.js:** Katılımcılara sıradan bir web sayfasının ötesinde deneyim sunmak amacıyla, WebGL tabanlı 3B sonsuz spiral galeri bileşeni Three.js kütüphanesi kullanılarak geliştirilmiştir.

2.1.2. Backend Teknolojileri

- **Node.js & NestJS (TypeScript):** NestJS, kurumsal düzeyde (enterprise-grade) modüler backend mimarisi geliştirmek için sektör standardı haline gelmiştir. Dependency Injection (Bağımlılık Enjeksiyonu) yapısı, modüler tasarımı ve güçlü TypeScript desteği sayesinde kod tabanının sürdürülebilirliği ve test edilebilirliği en üst seviyeye çıkarılmıştır.
- **BullMQ & Redis:** Yüz tanıma ve fotoğraf işleme operasyonları yoğun CPU gücü gerektirmektedir. NestJS ana thread'ini (Event Loop) bu ağır yüklerden korumak amacıyla BullMQ kütüphanesi ve Redis tabanlı asenkron iş kuyruğu (Job Queue) mimarisi kurulmuştur.
- **Sharp:** S3 depolama birimlerine yüklenmeden önce fotoğrafların optimize edilmesi, sıkıştırılması ve WebP formatında thumbnail'lerinin türetilmesi süreçlerini milisaniyeler seviyesinde tamamlayan son derece hızlı bir görüntü işleme kütüphanesidir.

2.1.3. Veritabanı ve Depolama Teknolojileri

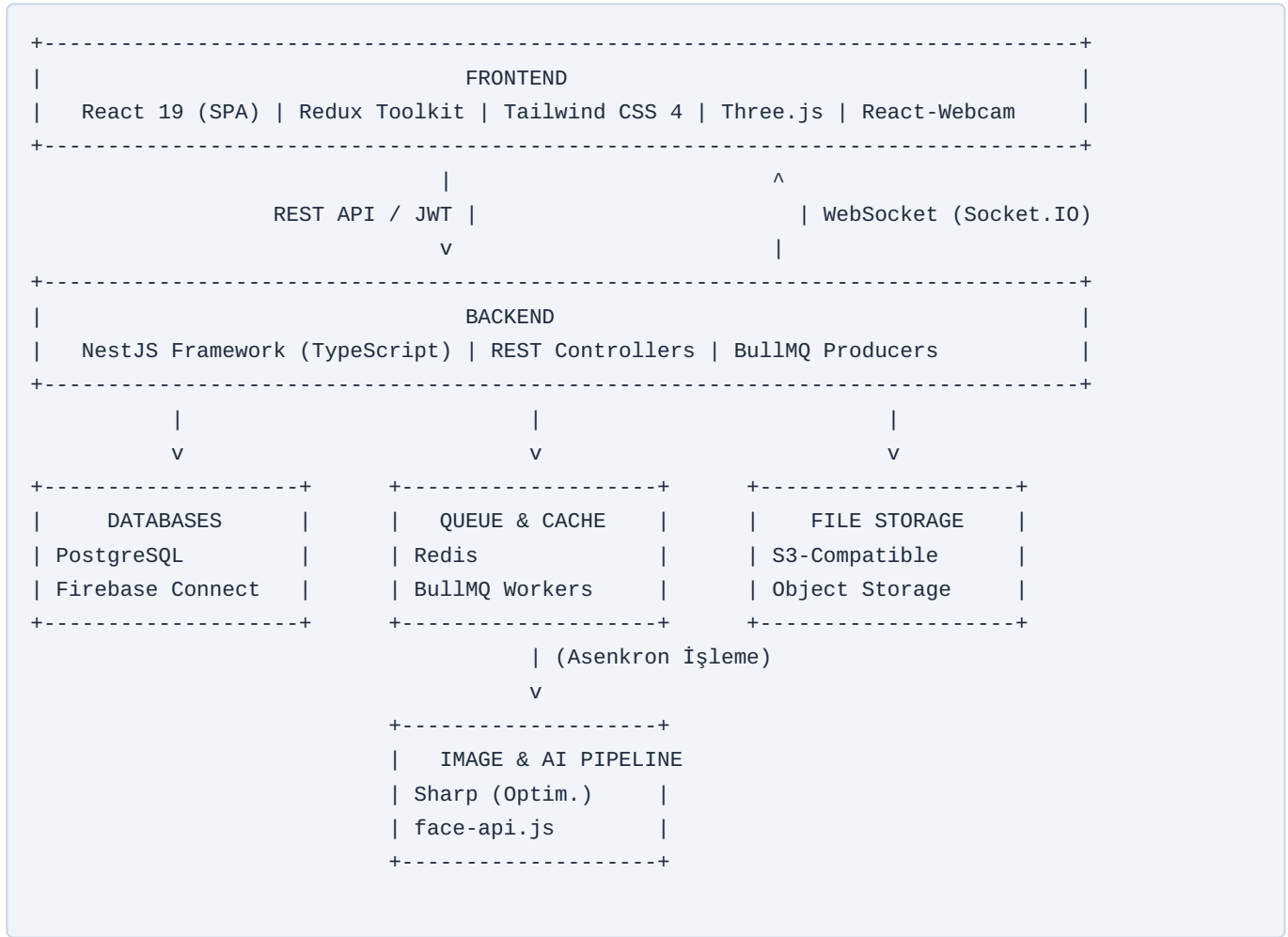
- **PostgreSQL:** Güçlü ilişkisel bütünlük, ACID uyumluluğu, karmaşık sorgu desteği og JSONB veri tipi yetenekleri nedeniyle projenin ana ilişkisel veritabanı olarak seçilmiştir. Yüz koordinatları ve yapılandırılmış meta veriler bu katmanda saklanır.
- **Firebase Data Connect:** Gerçek zamanlı veri akışları ve hızlı uygulama geliştirme süreçlerinde veri tabanı katmanını destekleyici bir köprü olarak konumlandırılmıştır.
- **S3 Uyumlu Nesne Depolama (Object Storage):** Yüksek çözünürlüklü orijinal fotoğrafların ve katılımcı selfie dosyalarının güvenli, düşük maliyetli ve yüksek erişilebilirlikli bir yapıda saklanması için AWS SDK v3 entegrasyonu ile S3 uyumlu bir nesne depolama alanı kullanılmıştır. Presigned URL mimarisi ile istemcilerin doğrudan depolama alanına yükleme yapması sağlanmıştır.

2.1.4. Yapay Zekâ ve Yardımcı Kütüphaneler

- **face-api.js & Canvas:** Google'ın TensorFlow.js kütüphanesi üzerine inşa edilmiş olan face-api.js, tarayıcı ve Node.js ortamlarında ek bir yapay zekâ sunucusuna ihtiyaç duymadan yüz tespiti, yüz tanıma ve yüz landmark tespiti yapılmasına olanak tanır. Node.js tarafında görüntülerin işlenebilmesi için `canvas` kütüphanesi ile entegre çalıştırılmıştır.
- **qrcode-with-logos:** Her etkinlik için özelleştirilmiş, orta kısmında SnapMatch veya ilgili organizasyonun logosunu barındıran, yüksek çözünürlüklü og taranabilir dinamik QR kodlarının oluşturulmasını sağlar.
- **Socket.IO (WebSocket):** Fotoğrafların arka planda işlenme durumlarının, kuyruk ilerlemelerinin ve yapay zekâ eşleşme başarılarının web paneline gerçek zamanlı (real-time) olarak yansıtılması amacıyla kullanılmıştır.
- **jsPDF & JSZip:** İstatistiksel raporların PDF formatında dışa aktarılması jsPDF ile gerçekleştirilirken; katılımcıların kendi fotoğraflarını tek tıkla toplu olarak indirebilmeleri için arka planda sıkıştırılmış ZIP arşivleri oluşturan JSZip kütüphanesi entegre edilmiştir.

2.2. Dağıtık ve Modüler Sistem Mimarisi

SnapMatch, modern yazılım mimarisi standardslarına uygun olarak üç katmanlı (3-Tier) ve modüler bir mimari yapıda tasarlanmıştır. Bu tasarım sayesinde kullanıcı arayüzü, iş mantığı ve veri işleme süreçleri birbirinden tamamen soyutlanmıştır:



Sistem mimarisindeki temel akış şu şekildedir: Kullanıcı arayüzü (Frontend) tüm veri taleplerini HTTP istekleri üzerinden JSON formatında Backend API'ye iletir. Backend, gelen istekleri kimlik doğrulama (JWT) ve yetkilendirme süzgeçlerinden geçirerek ilişkisel veritabanı sorgularına veya S3 dosya işlemlerine yönlendirir. Ağır iş yükleri (örneğin 100 adet fotoğrafın yüklenmesi ve yüz analizi yapılması) tetiklendiğinde, Backend ilgili işi BullMQ kuyruğuna yazar ve istemciye anında "İş Başlatıldı" yanıtı (HTTP 202 Accepted) döner.

Kuyruğa eklenen bu işler, arka planda çalışan bağımsız Worker süreçleri tarafından asenkron olarak tüketilir. Worker, fotoğrafları S3'ten çeker, yüz tespiti og öznitelik çıkarımı işlemlerini tamamlar, veritabanına kaydeder og Socket.IO kanalları üzerinden Frontend'e "İş Tamamlandı" bildirimini gerçek zamanlı olarak gönderir. Bu sayede web panelinde hiçbir donma veya gecikme yaşanmaz.

2.3. Çok Kiracılık (Multi-Tenancy) ve Veri İzolasyon Stratejileri

SnapMatch, SaaS (Software as a Service) doğasına uygun olarak çok kiracılı (multi-tenant) bir yapıda kurgulanmıştır. Bu kurguda, sisteme üye olan her fotoğraf stüdyosu, organizasyon firması veya bağımsız fotoğrafçı bir "Workspace" (Çalışma Alanı / Tenant) olarak tanımlanır. Çok kiracılı sistem mimarilerinde en kritik konu, kiracılar arasındaki veri izolasyonunun ve güvenliğinin kesin hatlarla sağlanmasıdır.

2.3.1. Mantıksal İzolasyon (Logical Separation) Modeli

Projede, maliyet etkinliği ve bakım kolaylığı sağlamak amacıyla "Paylaşımlı Veritabanı, Mantıksal İzolasyon" yaklaşımı tercih edilmiştir. Bu modele göre, tüm kiracıların verileri aynı fiziksel PostgreSQL veritabanında saklanır. Ancak, sistemdeki neredeyse tüm tablolar doğrudan veya dolaylı olarak bir `workspace\_id` yabancı anahtarına (foreign key) bağlanmıştır:

- **Veri Seviyesinde Filtreleme:** Yazılan tüm SQL sorguları ve NestJS ORM (TypeORM) repository'leri, sorgulama esnasında otomatik olarak aktif kullanıcının bağlı olduğu `workspace\_id` değerini filtre parametresi olarak ekler (`WHERE workspace\_id = ?`).
- **Güvenlik Sınırları:** Bir kullanıcının HTTP istek başlığında gönderdiği JWT (JSON Web Token) çözümlendiğinde, kullanıcının hangi workspace'e ait olduğu tespit edilir. Eğer kullanıcı, kendi workspace'ine ait olmayan bir `event\_id` veya `photo\_id` üzerinde işlem yapmaya çalışırsa, NestJS Guard katmanı isteği anında engelleyerek HTTP 403 Forbidden hatası döner.
- **Dosya Depolama İzolasyonu:** S3 bucket üzerinde saklanan tüm dosyalar, workspace bazlı klasör yapılarında organize edilir (`/workspaces/{workspace\_id}/events/{event\_id}/photos/{photo\_id}.webp`). Bu sayede, dosya erişim yetkileri de workspace seviyesinde izole edilmiş olur.

Mimarinin Avantajı: Mantıksal izolasyon modeli, her kiracı için ayrı veritabanı açma maliyetini ortadan kaldırarak veritabanı kaynaklarının (bağlantı havuzları, bellek, CPU) en verimli şekilde ortak kullanılmasını sağlar. Bu durum, SaaS platformunun işletme ve barındırma maliyetlerini minimumda tutar.

2.4. Kullanıcı Roller ve Eriřim Kontrol Matrisi (RBAC)

Sistem güvenlięi ve operasyonel iřbölümü için Rol Bazlı Eriřim Kontrolü (Role-Based Access Control - RBAC) altyapısı uygulanmıřtır. Sistemde tanımlı olan roller ve bu rollerin platform üzerindeki yetki sınırları ařaęıda detaylandırılmıřtır:

2.4.1. Rol Tanımları

1. **Platform Admin:** Sistemin genel yöneticisidir. Tüm workspace'leri, kullanıcıları, sistem loglarını ve limit ařımlarını izleme ve yönetme yetkisine sahiptir.
2. **Fotoęrafçı Admin (Tenant Owner):** İlgili Workspace'in (stüdyo/firma) sahibidir. Yeni etkinlikler oluşturabilir, çalışma arkadaşları davet edebilir, faturalandırma ve üyelik planlarını yönetebilir, tüm etkinlikleri silebilir.
3. **Event Owner (Etkinlik Yöneticisi):** Fotoęrafçı Admin tarafından belirli etkinliklere atanmıř personeldir. Sorumlu olduęu etkinliklerin detaylarını düzenleyebilir, fotoęraf yükleyebilir, AI eřleřtirmelerini başlatabilir ve onay/red (review) süreçlerini yönetebilir.
4. **Uploader (Fotoęraf Yükleyici):** Yalnızca etkinlik alanında çekilen fotoęrafları sisteme yüklemekle görevli kısıtlı roldür. Etkinlik silme, katılımcı yönetimi veya finansal metriklere erişim yetkisi yoktur; sadece hızlı yükleme ekranını görebilir.
5. **Katılımcı (Guest):** QR kod okutarak sisteme giren son kullanıcıdır. Sadece rıza metnini onaylayıp selfie yükleyebilir ve yapay zekâ tarafından kendisiyle eřleřtirilip onaylanmış fotoęrafları görüntüleyip indirebilir. Panel arayüzlerine erişimi kesinlikle engellenmiřtir.

2.4.2. Detaylı RBAC Matrisi

Modül / Eylem	Platform Admin	Fotoğrafçı Admin	Event Owner	Uploader	Katılımcı (Guest)
Workspace Ayarları	Tam Yetki	Düzenleme / Görüntüleme	Yok	Yok	Yok
Etkinlik CRUD	Görüntüleme	Oluşturma / Silme / Düzenleme	Sadece Atananları Düzenleme	Yok	Yok
Fotoğraf Yükleme	Yok	Tam Yetki	Tam Yetki	Yükleme Yetkisi	Düğünde Kısıtlı Yükleme
Yüz Verisi Silme	Log İzleme	Görüntüleme / Silme	Görüntüleme / Silme	Yok	Kendi Verisini Silme
AI Matching Tetikleme	İzleme	Tam Yetki	Tam Yetki	Yok	Yok
Match Review (Onay/Red)	Yok	Tam Yetki	Tam Yetki	Yok	Yok
Audit Logs (Sistem Logları)	Tüm Sistem	Kendi Workspace'i	Kendi Etkinliği	Yok	Yok

2.5. Detaylı Veri Modeli ve PostgreSQL ER Diyagram Analizi

SnapMatch veritabanı şeması, yüksek veri tutarlılığı, hızlı arama performansı ve esnek çok kiracılı yapıyı desteklemek üzere PostgreSQL üzerinde normalize edilerek tasarlanmıştır. Veritabanındaki ilişkiler ve kritik tabloların yapısal rolleri şu şekildedir:

2.5.1. İlişkisel Şema ve Tablo Roller

- **workspaces (1:N) events:** Her workspace altında birden fazla etkinlik (event) tanımlanabilir. Etkinliklerin hangi stüdyoya ait olduğu bu ilişki ile çözülür.
- **events (1:N) photos & participants:** Bir etkinlik altında binlerce fotoğraf (`photos`) ve etkinliğe katılan yüzlerce misafir (`participants`) bulunur.
- **photos (1:N) photo_faces:** Bir fotoğraf karesinde birden fazla insan yüzü bulunabilir. Bu nedenle, fotoğraflar ile fotoğraflardan tespit edilen her bir yüz arasında 1:N (Bire Çok) ilişki kurulmuştur.
- **participants (1:1) face_samples:** Katılımcının sisteme kaydettiği biyometrik selfie verisi ve öznitelik vektörü bu tabloda saklanır. Güvenlik ve gizlilik nedeniyle bire bir ilişki tercih edilmiştir.
- **photo_faces (N:M) face_samples → matches:** Tespit edilen yüzler ile katılımcıların biyometrik kayıtlarının yapay zekâ tarafından karşılaştırılması sonucu üretilen eşleşmeler, `matches` adlı kesişim (junction) tablosunda tutulur. Bu tablo eşleşmenin güven skorunu ve onay durumunu barındırır.

2.5.2. Kritik Tabloların PostgreSQL DDL SQL Tanımları

```
-- Workspace Tanımları (Multi-Tenant Ana Tablo)
CREATE TABLE workspaces (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  name VARCHAR(255) NOT NULL,
  plan VARCHAR(50) DEFAULT 'free',
  storage_limit BIGINT DEFAULT 10737418240, -- 10 GB
  branding_json JSONB,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

-- Kullanıcı Tanımları
CREATE TABLE users (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  workspace_id UUID REFERENCES workspaces(id) ON DELETE CASCADE,
  name VARCHAR(255) NOT NULL,
  email VARCHAR(255) UNIQUE NOT NULL,
  password_hash VARCHAR(255) NOT NULL,
  role VARCHAR(50) NOT NULL CHECK (role IN ('platform_admin', 'photographer_admin',
'event_owner', 'uploader')),
  status VARCHAR(50) DEFAULT 'active',
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);
```

```

-- Etkinlik Tanımları
CREATE TABLE events (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  workspace_id UUID REFERENCES workspaces(id) ON DELETE CASCADE,
  title VARCHAR(255) NOT NULL,
  category VARCHAR(100) NOT NULL,
  date DATE NOT NULL,
  status VARCHAR(50) DEFAULT 'draft',
  qr_token VARCHAR(255) UNIQUE NOT NULL,
  consent_text TEXT NOT NULL,
  guest_upload_enabled BOOLEAN DEFAULT FALSE,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

-- Fotoğraflar Tablosu
CREATE TABLE photos (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  event_id UUID REFERENCES events(id) ON DELETE CASCADE,
  original_url TEXT NOT NULL,
  thumbnail_url TEXT NOT NULL,
  file_hash VARCHAR(64) NOT NULL,
  source VARCHAR(50) DEFAULT 'photographer', -- 'photographer' or 'guest'
  compressed BOOLEAN DEFAULT TRUE,
  size_bytes INT NOT NULL,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

-- Fotoğraflarda Tespit Edilen Yüzler
CREATE TABLE photo_faces (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  photo_id UUID REFERENCES photos(id) ON DELETE CASCADE,
  bbox JSONB NOT NULL, -- {x, y, width, height}
  embedding JSONB NOT NULL, -- 128 boyutlu float dizisi
  detection_confidence DOUBLE PRECISION NOT NULL,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

-- Katılımcı (Guest) Tablosu
CREATE TABLE participants (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  event_id UUID REFERENCES events(id) ON DELETE CASCADE,
  display_name VARCHAR(255) NOT NULL,
  guest_token VARCHAR(255) UNIQUE NOT NULL,
  consent_accepted_at TIMESTAMP WITH TIME ZONE,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

-- Katılımcı Selfie / Yüz Örnekleri
CREATE TABLE face_samples (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  participant_id UUID REFERENCES participants(id) ON DELETE CASCADE,
  image_url TEXT NOT NULL,

```

```

    embedding JSONB NOT NULL, -- 128 boyutlu float dizisi
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

-- Yapay Zekâ Eşleşme Sonuçları
CREATE TABLE matches (
    id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
    event_id UUID REFERENCES events(id) ON DELETE CASCADE,
    participant_id UUID REFERENCES participants(id) ON DELETE CASCADE,
    photo_id UUID REFERENCES photos(id) ON DELETE CASCADE,
    photo_face_id UUID REFERENCES photo_faces(id) ON DELETE CASCADE,
    confidence DOUBLE PRECISION NOT NULL,
    status VARCHAR(50) DEFAULT 'pending', -- 'pending', 'approved', 'rejected'
    reviewed_by UUID REFERENCES users(id),
    created_at TIMESTAMP WITH TIME ZONE DEFAULT CURRENT_TIMESTAMP
);

```

2.6. REST API Tasarımı ve Veri Değişim Sözleşmeleri

SnapMatch backend servisleri, istemci uygulamalarla iletişim kurmak üzere tamamen JSON tabanlı, kaynak yönelimli (resource-oriented) bir REST API sunmaktadır. API tasarımı, HTTP durum kodlarını (Status Codes) ve standart HTTP metotlarını (GET, POST, PUT, DELETE) tam uyumlu olarak kullanır.

2.6.1. Kimlik Doğrulama ve Güvenlik İstisnaları

API isteklerinin güvenliği için tüm uç noktalar (endpoints) JWT doğrulaması yapan bir `AuthGuard` süzgecinden geçer. İstisnalar yalnızca misafir akışına ait olan ve QR kod ile tetiklenen `/guest/\*` uç noktalarıdır. Bu uç noktalarda, sistem güvenliği dinamik olarak üretilen og veritabanında doğrulanan benzersiz `guest\_token` parametreleri ile sağlanmaktadır.

2.6.2. Kritik API Sözleşmeleri ve JSON Örnekleri

1. Fotoğraf Toplu Yükleme İstek Sözleşmesi:

Fotoğrafçının panele toplu görsel yüklemesi esnasında, doğrudan büyük dosyaları API'ye göndermek yerine, güvenli yükleme için presigned S3 URL'leri talep edilir:

Endpoint: **POST** /events/:id/photos/batch

Request Body (İstek Gövdesi):

```
{
  "files": [
    {
      "filename": "DSC_1024.jpg",
      "file_size": 8452012,
      "md5_hash": "e10adc3949ba59abbe56e057f20f883e"
    },
    {
      "filename": "DSC_1025.jpg",
      "file_size": 9102458,
      "md5_hash": "c33367701511b4f6020ec61ded352059"
    }
  ]
}
```

Response Body (Yanıt Gövdesi):

```
{
  "batch_id": "9b1deb4d-3b7d-4bad-9bdd-2b0d7b3dcb6d",
  "upload_urls": [
    {
      "filename": "DSC_1024.jpg",
      "upload_url": "https://s3.eu-central-1.amazonaws.com/snapmatch-bucket/
workspaces/...",
      "photo_id": "a1a2b3c4-d5e6-4f7g-8h9i-0j1k2l3m4n5o"
    }
  ]
}
```



```
    },  
    {  
      "filename": "DSC_1025.jpg",  
      "upload_url": "https://s3.eu-central-1.amazonaws.com/snapmatch-bucket/  
workspaces/...",  
      "photo_id": "b2b3c4d5-e6f7-4g8h-9i0j-1k2l3m4n5o6p"  
    }  
  ]  
}
```

2. Yapay Zekâ Eşleştirme Tetikleme İstek Sözleşmesi:

Fotoğraflar yüklendikten ve katılımcılar selfie kayıtlarını tamamladıktan sonra, yapay zekâ analiz boru hattı belirli bir güven eşik değeri (threshold) ile tetiklenir:

Endpoint: `POST /events/:id/matching-jobs`

Request Body (İstek Gövdesi):

```
{
  "threshold": 0.65,
  "notify_participants": true
}
```

Response Body (Yanıt Gövdesi - HTTP 202 Accepted):

```
{
  "job_id": "7fa84c10-9284-4861-a083-bf0f193cb9d2",
  "status": "queued",
  "message": "AI matching pipeline has been successfully queued.",
  "estimated_time_seconds": 120,
  "created_at": "2026-07-17T00:15:30Z"
}
```

3. Eşleşme Durum Güncelleme İstek Sözleşmesi (Match Review):

Fotoğrafçının, yapay zekâ tarafından yapılan analiz sonuçlarını onaylaması veya reddetmesi için kullanılan uç noktadır:

Endpoint: `PUT /matches/:id/status`

Request Body (İstek Gövdesi):

```
{
  "status": "approved" -- veya 'rejected'
}
```

Response Body (Yanıt Gövdesi):

```
{
  "match_id": "4bc8e02b-da93-45bb-b3b3-8c460d1ba8fc",
  "status": "approved",
  "reviewed_by": "123e4567-e89b-12d3-a456-426614174000",
  "reviewed_at": "2026-07-17T00:18:45Z"
}
```

3. GELİŞTİRME VE UYGULAMA

3.1. Kimlik Doğrulama ve Güvenli Token Yönetimi

SnapMatch panel güvenliği, durumsuz (stateless) JWT (JSON Web Token) tabanlı bir kimlik doğrulama mekanizması ile sağlanmaktadır. Kullanıcı ve oturum güvenliği süreçleri şu adımlarla yürütülür:

3.1.1. Şifreleme ve Güvenli Kayıt

Kullanıcıların şifreleri, veritabanına kesinlikle düz metin (plain text) olarak kaydedilmez. Kayıt esnasında, tek yönlü kriptografik özetleme (hashing) algoritması olan **bcrypt** kullanılarak, en az 10 tuzlama (salt rounds) değeriyle şifrelenir. Giriş isteklerinde, gelen şifre bcrypt'in güvenli karşılaştırma metotlarıyla kontrol edilir.

3.1.2. Token Ömrü ve Güvenlik Önlemleri

- **Erişim Token'ı (Access Token):** Kullanıcının kimliğini doğrulamak için kullanılan ana token'dır. Güvenlik gerekçesiyle ömrü 15 dakika ile sınırlandırılmıştır. HTTP isteklerinin başlık alanında (Authorization: Bearer <token>) taşınır.
- **Yenileme Token'ı (Refresh Token):** Kullanıcının tarayıcı oturumunu kaybetmeden kesintisiz çalışabilmesi için, ömrü 7 gün olan bir yenileme token'ı üretilir. Bu token, XSS (Cross-Site Scripting) saldırılarından korunmak amacıyla istemci tarafında Javascript tarafından erişilemeyen, sadece sunucu tarafından okunabilen ``HttpOnly``, ``Secure`` ve ``SameSite=Strict`` çerezlerinde (cookies) saklanır.
- **Rol Bazlı Rota Koruyucular (Guards):** NestJS tarafında yazılan ``RolesGuard`` bileşeni, çözümlenen JWT içindeki rol bilgisini okuyarak, denetleyici (controller) metodunun üzerinde tanımlı olan ``@Roles(UserRole.ADMIN)`` dekoratörleriyle karşılaştırır. Yetkisiz istekler iş mantığına ulaşmadan filtre katmanında engellenir.

Güvenlik İpucu: Token çalınması durumlarında hasarı en aza indirmek için, Access Token ömürleri olabildiğince kısa tutulmuş; yenileme isteklerinde ise veritabanında tutulan tek kullanımlık "Refresh Token Rotation" tekniği uygulanmıştır.

3.2. Etkinlik Yönetimi ve Yaşam Döngüsü Modülü

Etkinlik Yönetimi, fotoğrafçıların iş süreçlerini organize ettiği temel modüldür. Her etkinlik, kendi bağımsız yaşam döngüsüne sahip bir iş birimi olarak kurgulanmıştır.

3.2.1. Etkinlik Yaşam Döngüsü Adımları

Bir etkinliğin sistemdeki durumu, veri tabanında tanımlı olan `status` alanı üzerinden aşağıdaki şemaya göre yönetilir:

1. **Taslak (Draft):** Etkinlik oluşturulmuş ancak henüz katılıma veya fotoğraf yüklemeye açılmamıştır. Sadece fotoğrafçı tarafından düzenlenebilir.
2. **Yayında (Active):** Etkinlik QR kodu aktiftir, katılımcılar sisteme girip selfie kaydı yapabilir, fotoğraf yükleme ve yapay zekâ analiz süreçleri tetiklenebilir.
3. **Arşivlendi (Archived):** Etkinlik sona ermiştir. Yeni katılımcı girişi veya yeni fotoğraf yüklenmesi engellenmiştir. Katılımcılar sadece mevcut onaylı fotoğraflarını görebilir ve indirebilirler.

3.2.2. Düğün Kategorisi ve Misafir Fotoğraf Yükleme Dinamiği

SnapMatch, düğün (*wedding*) kategorisindeki organizasyonlar için sektörel bir inovasyon sunmaktadır. Düğünlerde davetliler de cep telefonlarıyla yüzlerce anlık fotoğraf çekmektedir. Etkinlik kategorisi "wedding" olarak seçildiğinde, `guest\_upload\_enabled` parametresi varsayılan olarak aktif hale gelir.

Bu özellik sayesinde, QR kod ile sisteme giren misafirler sadece kendi fotoğraflarını aramakla kalmaz; aynı zamanda telefonlarıyla çektikleri düğün karelerini de sisteme "guest" kaynağıyla yükleyebilirler. Yüklenen bu fotoğraflar da aynı yapay zekâ boru hattından (AI Pipeline) geçirilerek, fotoğraftaki diğer misafirlerle otomatik olarak eşleştirilir. Bu durum, düğün fotoğraflarının kolektif bir havuzda toplanmasını sağlayarak sosyal etkileşimi en üst seviyeye çıkarır.

3.3. QR Kod Üretimi ve Dinamik Misafir (Guest) Akışı

Katılımcıların sisteme en hızlı ve pürüzsüz şekilde dahil olabilmesi, projenin en kritik başarı faktörlerinden biridir. Bu amaçla, QR kod tabanlı dinamik bir misafir akışı tasarlanmıştır:

3.3.1. Özelleştirilmiş QR Kod Üretimi

Bir etkinlik "Yayında" durumuna getirildiğinde, arka planda o etkinliğe özel benzersiz bir kriptografik `qr\_token` üretilir. `qrcode-with-logos` kütüphanesi kullanılarak, bu token'ı barındıran dinamik bir QR kod görseli oluşturulur. QR kodun merkezine otomatik olarak fotoğrafçı stüdyosunun veya SnapMatch'in logosu yerleştirilir. Üretilen QR kod, yüksek çözünürlüklü SVG formatında indirilerek etkinlik alanındaki masa kartlarına, afişlere veya dijital ekranlara basılmaya hazır hale getirilir.

3.3.2. Mobile-First Misafir Deneyim Adımları

Uygulama indirme zorunluluğunun olmaması, katılımcı katılım oranını dramatik şekilde artırmaktadır. Katılımcı QR kodu tarattığında süreç şu şekilde ilerler:

- **Landing & Bilgilendirme:** Katılımcı, mobil tarayıcısında açılan şık bir karşılama ekranına ulaşır. Burada etkinliğin adı, tarihi ve ev sahibi bilgileri yer alır.
- **KVKK Açık Rıza Onayı:** Sistem, biyometrik veri işlemeyi içerdiğinden, yasal zorunluluk gereği katılımcıya detaylı bir KVKK Açık Rıza ve Aydınlatma Metni sunar. Katılımcı "Okudum, Onaylıyorum" onay kutusunu işaretlemeyen bir sonraki adıma geçemez.
- **Kamera Entegrasyonu ve Selfie:** Onay alındıktan sonra tarayıcı içi kamera API'si (`react-webcam`) tetiklenir. Katılımcı pürüzsüz bir kılavuz halkası eşliğinde anlık bir selfie çeker. Çekilen selfie, optimize edilerek yüz vektörünün çıkarılması için sunucuya gönderilir.

3.4. Çoklu Dosya Yükleme, Kuyruk Yapısı ve Sharp Görüntü Sıkıştırma

Büyük etkinliklerde fotoğrafçılar tek bir günde 2000 ila 5000 adet arasında yüksek çözünürlüklü RAW veya JPEG fotoğraf çekmektedir. Bu denli yüksek bir veri hacminin web paneline aktarılması ve sunucuda işlenmesi, özel performans mimarileri gerektirir.

3.4.1. Presigned URL ile Doğrudan S3 Yükleme

Geleneksel dosya yükleme yöntemlerinde dosyalar önce backend sunucusuna gider, sunucu belleğinde geçici olarak tutulur ve ardından depolama birimine yazılır. Bu durum, 50 fotoğraf aynı anda yüklendiğinde backend sunucusunun belleğinin tükenmesine (Out of Memory) ve çökmesine neden olur.

SnapMatch bu sorunu **S3 Presigned URL** mimarisi ile aşmaktadır: Frontend, yüklenecek dosyaların listesini ve boyutlarını backend'e bildirir. Backend, S3 üzerinden her dosya için sadece 5 dakika geçerli olan güvenli, benzersiz yükleme adresleri (presigned upload URLs) üretir ve istemciye döner. İstemci tarayıcısı, büyük fotoğraf dosyalarını backend sunucusuna hiç uğratmadan, doğrudan ve paralel olarak AWS S3 bucket'ına yükler. Bu sayede backend sunucusunun ağ ve bellek yükü sıfıra indirgenir.

3.4.2. Sharp Görüntü Optimizasyonu ve Thumbnail Boru Hattı

S3'e yükleme tamamlandığında backend, BullMQ iş kuyruğuna bir "Görüntü İşleme" görevi ekler. Worker süreçleri bu görevi aldığı anda şu işlemleri gerçekleştirir:

- **Metadata Temizliği:** Fotoğraftaki GPS konumları, kamera modelleri gibi gereksiz EXIF verileri güvenlik ve dosya boyutu tasarrufu amacıyla temizlenir.
- **Sıkıştırma & Format Dönüşümü:** Yüksek çözünürlüklü orijinal görüntüler, Sharp kütüphanesi yardımıyla modern ve yüksek sıkıştırma oranına sahip **WebP** formatına dönüştürülür. Kalite kaybı olmadan dosya boyutları %70'e varan oranda düşürülür.
- **Thumbnail (Küçük Resim) Üretimi:** Galeri ekranlarında hızlı yükleme sağlamak amacıyla, görüntülerin 400px genişliğinde küçük thumbnail versiyonları üretilir ve S3 üzerinde `/thumbnails/` klasörüne kaydedilir.

3.5. Üç Boyutlu WebGL (Three.js) Tabanlı Spiral Galeri Geliştirilmesi

SnapMatch, kullanıcı deneyimini standart web sitelerinin ötesine taşımak ve etkinlik sonrasındaki heyecanı artırmak amacıyla WebGL teknolojisinden faydalanmaktadır. Klasik grid listeleme galerilerine alternatif olarak geliştirilen **3B Spiral Galeri**, katılımcılara oyunlaştırılmış bir görsel deneyim sunar.

3.5.1. Teknik Matematiksel Altyapı ve Render Süreci

Üç boyutlu galeri, tarayıcının ekran kartını (GPU) doğrudan kullanan HTML5 Canvas ve **Three.js** kütüphanesi üzerine inşa edilmiştir. Fotoğraflar, 3 boyutlu uzayda matematiksel bir **Logaritmik Spiral (Altın Spiral)** eğrisi üzerine dizilir. Spiralin konum koordinatları kutupsal formüller kullanılarak hesaplanır:

$$x = a \cdot e^{b \cdot \theta} \cdot \cos(\theta) \quad y = a \cdot e^{b \cdot \theta} \cdot \sin(\theta) \quad z = c \cdot \theta$$

Bu formülasyon sayesinde, katılımcı faresini veya mobil cihazındaki parmağını kaydıldıkça (scroll/drag), kamera spiralin derinliklerine doğru pürüzsüz bir şekilde hareket eder. Her bir fotoğraf karesi, 3B uzayda üzerine yerleştirilen dokular (textures) olarak render edilir.

3.5.2. Performans Optimizasyonları

3B ortamda aynı anda yüzlerce yüksek çözünürlüklü fotoğrafın render edilmesi mobil cihazların kasılmasına neden olabilir. Bunun önüne geçmek amacıyla şu optimizasyonlar uygulanmıştır:

- **LOD (Level of Detail):** Kameraya uzak olan fotoğraflarda orijinal görüntüler yerine 2.1.2 bölümünde üretilen düşük çözünürlüklü thumbnail WebP görselleri kullanılır. Yalnızca kameranın tam önündeki fotoğraflar yüksek çözünürlüklü olarak yüklenir.
- **Frustum Culling:** Kameranın görüş açısı dışında kalan (arkada veya çok uzakta kalan) 3B nesneler GPU hesaplama döngüsünden çıkarılarak render yükü mini seviyeye indirilir.

3.6. Katılımcı Yüz Kaydı ve Kamera API Entegrasyonları

Yapay zekâ eşleştirme başarısının en temel bileşeni, katılımcının sisteme kaydettiği referans yüz verisinin (selfie) kalitesidir. Kalitesiz, karanlık veya çoklu yüz içeren referans fotoğraflar yapay zekânın yanlış sonuçlar üretmesine yol açar. Bu nedenle, yüz kayıt süreci sıkı tarayıcı içi kontrollerle donatılmıştır.

3.6.1. Tarayıcı İçi Gerçek Zamanlı Kontroller

Katılımcı mobil cihazı üzerinden selfie çekmeye çalışıldığında, `react-webcam` bileşeni üzerinden gelen anlık video karesi (frame) daha sunucuya gönderilmeden önce istemci tarafında face-api.js'in hafif (lightweight) modeli ile taranır:

- **Yüz Varlığı Kontrolü:** Kamera açısında en az bir adet insan yüzü olduğu doğrulanır. Eğer açıda hiç yüz yoksa "Lütfen kameraya doğru bakınız" uyarısı gösterilir og çekim butonu kilitlenir.
- **Çoklu Yüz Engelleme:** Kamera açısında birden fazla yüz tespit edilirse, Submittı ve veri doğruluğu gerekçesiyle "Lütfen kadrja sadece kendinizin girdiğinden emin olun" uyarısı verilerek çekim engellenir.
- **Aydınlık ve Açık Kontrolü:** Yüzün kameraya olan açısı ve ortamın ışık düzeyi analiz edilerek referans fotoğraf kalitesi doğrulanır.

3.6.2. Güvenli Aktarım ve İşleme

İmleç tarafındaki kriterleri başarıyla geçen selfie karesi, taban 64 (Base64) formatına dönüştürülerek güvenli HTTPS kanalı üzerinden sunucuya aktarılır. Sunucu tarafında bu referans resim `face\_samples` tablosuna kaydedilir ve yapay zekâ motorunun ana karşılaştırma referansı (anchor) olarak belirlenir.

3.7. face-api.js ile Yapay Zekâ Yüz Tanıma ve Embedding Pipeline'i

SnapMatch, yüz tanıma süreçlerini herhangi bir harici lisans bedeli veya bulut kullanım ücreti ödemeksizin tamamen sunucu tarafında (on-premise / self-hosted) yürütebilecek bir boru hattı (pipeline) ile tasarlamıştır.

3.7.1. Model Mimarisi og Katmanları

Yüz tanıma işlemlerinin temelinde Google'ın TensorFlow.js motoru üzerine kurulmuş olan **face-api.js** kütüphanesi yer alır. Sistem, sunucu tarafında Node.js ortamında çalışırken şu üç yapay sinir ağı modelini sırasıyla çalıştırır:

- 1. Yüz Tespiti (SSD MobileNetv1):** Görüntü üzerindeki insan yüzlerini saptar ve konum koordinatlarını (Bounding Box - BBOX) belirler. Bu model, yüksek doğruluk oranına sahiptir ve farklı ışık koşullarında stabil çalışır.
- 2. Yüz Landmark Tespiti (68 Point Model):** Tespit edilen yüz üzerindeki göz, kaş, burun, ağız ve çene hattı gibi 68 kritik referans noktasını saptar. Yüzün sağa/sola dönüklük açısını tespit ederek düzeltme (alignment) yapar.
- 3. Öznitelik Çıkarımı (ResNet-34):** Düzeltilmiş yüz görselini analiz ederek, o yüze ait benzersiz biyometrik özellikleri temsil eden **128 boyutlu bir kayan noktalı sayısal vektör (Embedding / Vector)** üretir. Bu vektör, insan yüzünün sayısal bir imzası niteliğindedir.

3.7.2. Matematiksel Eşleştirme ve Öklid Uzaklığı Teorisi

Yapay zekâ motoru, iki farklı yüzün (örneğin katılımcı selfiesi A ile bir etkinlik fotoğrafında tespit edilen yüz B) aynı kişiye ait olup olmadığını anlamak için bu iki vektör arasındaki **Öklid Uzaklığı (Euclidean Distance)** değerini hesaplar:

$$d(A, B) = \sqrt{\sum_{i=1}^{128} (A_i - B_i)^2}$$

Elde edilen uzaklık değeri $d(A, B)$ ne kadar küçükse, iki yüzün birbirine olan benzerliği o kadar yüksektir. Bilgisayarlı görü literatüründe, ResNet-34 modelleri için standart benzerlik sınır değeri (threshold) 0.60 olarak kabul edilir. Mesafe 0.60 'ın altında ise bu iki yüzün aynı kişiye ait olduğu matematiksel olarak doğrulanmış olur.

3.8. Eşleşme Değerlendirme (Match Review) ve Güven Skoru Yönetimi

Yapay zekâ modelleri ne kadar gelişmiş olursa olsun, gerçek saha koşullarında (ters ışık, gözlük kullanımı, şapka, arka plan bulanıklığı vb.) sınır değerlerde kararsız sonuçlar üretebilir. SnapMatch, bu durumların önüne geçmek ve katılımcılara %100 doğru galeri sunabilmek amacıyla iki aşamalı bir karar destek mekanizması kurgulamıştır.

3.8.1. Çift Eşikli Karar Mekanizması

Sistemde tek bir eşik değeri yerine, iki farklı güven sınırı tanımlanmıştır:

- **Üst Eşik Değeri (Güven Skoru ≥ 0.80):** Yapay zekânın eşleşmeden kesin olarak emin olduğu durumdur. Bu eşleşmeler sistem tarafından otomatik olarak onaylanır (`approved`) og anında katılımcının galerisine yansıtılır.
- **Ara Bölge ($0.60 \leq \text{Güven Skoru} < 0.80$):** Yapay zekânın yüzleri birbirine benzettiği ancak kesin karara varamadığı kararsızlık bölgesidir. Bu eşleşmeler, sistem tarafından askıya alınır (`pending`) ve fotoğrafının onay ekranına (Match Review) gönderilir.
- **Alt Bölge (Güven Skoru < 0.60):** Benzerliğin yetersiz olduğu durumlardır. Sistem bu eşleşmeleri doğrudan reddeder ve veritabanına kaydetmez.

3.8.2. Match Review Arayüzü ve Manuel Müdahale

Fotoğrafçı Admin veya Event Owner, web panelindeki "Eşleşme İnceleme" ekranına girdiğinde, ara bölgede kalan tüm aday eşleşmeleri yan yana görebilir. Arayüzde, katılımcının orijinal selfiesi ile etkinlik fotoğrafından kırılan yüz görseli yan yana gösterilir ve yapay zekânın hesapladığı güven skoru yüzde olarak sunulur.

Kullanıcı tek bir tıklamayla bu eşleşmeyi onaylayarak katılımcı galerisine gönderebilir veya reddederek kaldırabilir. Ayrıca, sistemde yapay zekânın tamamen gözden kaçırdığı durumlar için, fotoğrafçı manuel olarak bir fotoğrafı seçip belirli bir katılımcıyla doğrudan ilişkilendirebilir (`POST /matches/manual`).

3.9. KVKK / GDPR Uyum Stratejileri, Veri Minimizasyonu ve Silme Hakkı

Biyometrik veriler (yüz haritaları ve vektörleri), Türkiye Cumhuriyeti'nin 6698 sayılı Kişisel Verilerin Korunması Kanunu (KVKK) og Avrupa Birliği Genel Veri Koruma Yönetmeliği (GDPR) kapsamında "Özel Nitelikli Kişisel Veri" statüsündedir. Bu verilerin işlenmesi ve saklanması son derece sıkı yasal yükümlülöklere tabidir. SnapMatch, tasarım aşamasından itibaren gizliliği ön planda tutan (Privacy by Design) bir mimariyle geliştirilmiştir.

3.9.1. Yasal Uyum Sütunları

- Açık Rıza ve Şeffaflık:** Katılımcılar sisteme giriş yaptıklarında, biyometrik verilerinin hangi amaçla (yalnızca etkinlik fotoğraflarını bulma), nerede ve ne kadar süreyle işleneceğini belirten detaylı bir Açık Rıza Beyanı ile karşılaşılır. Katılımcının rızasını verdiği tarih ve saat, veritabanında `consent\_accepted\_at` alanında kriptografik imza niteliğinde günlüklenir (audit logging).
- Veri Minimizasyonu:** Katılımcılardan alınan yüz verileri, kesinlikle genel bir yüz tanıma havuzuna eklenmez. Her katılımcının yüz vektörü, yalnızca katıldığı ilgili `event\_id` kapsamındaki fotoğraflarla karşılaştırılır. Etkinlikler arası veri geçişi veya çapraz sorgulama kesinlikle teknik olarak engellenmiştir.
- Unutulma (Silme) Hakkı:** Katılımcılar, galeri ekranlarında yer alan "Verilerimi Sil" butonu veya `DELETE / participants/:id/face-data` uç noktası aracılığıyla, sisteme kaydettikleri selfie görselini, üretilen 128 boyutlu yüz vektörünü ve tüm yapay zekâ eşleşme geçmişlerini sistemden tek bir tıkla kalıcı olarak silebilirler. Bu işlem gerçekleştiğinde, ilişkili tüm veriler S3 og PostgreSQL üzerinden fiziksel olarak silinir (Hard Delete).
- Audit Logging (Güvenlik Günlükleri):** Sistemde gerçekleştirilen tüm veri ekleme, silme ve yetkilendirme işlemleri, `activity\_logs` tablosunda işlem yapan kullanıcının IP adresi ve cihaz bilgileriyle kayıt altına alınarak olası yasal denetimlere karşı izlenebilirlik sağlanır.

3.10. Raporlama, İstatistiksel İzleme ve Analytics Kontrol Paneli

Fotoğrafçıların ve etkinlik organizatörlerinin platformdan elde ettikleri ticari verimi ölçümleyebilmeleri ve sistemin operasyonel durumunu izleyebilmeleri amacıyla detaylı bir Raporlama ve Analytics modülü geliştirilmiştir.

3.10.1. Dashboard Metrikleri ve KPI Göstergeleri

Kullanıcı panele giriş yaptığında, rolüne ve yetkili olduğu workspace'e göre dinamik olarak hesaplanan şu metriklerle karşılaşır:

- **Toplam Fotoğraf Sayısı:** Workspace genelinde ve etkinlik bazında yüklenmiş olan aktif fotoğraf miktarı.
- **Katılımcı Kayıt Oranı:** Etkinlik davetlileri arasında QR kodu okutarak selfie kaydı yapmış olan aktif kullanıcı sayısı.
- **Yapay Zekâ Eşleşme Oranı:** Yüklenen fotoğraflardan en az bir katılımcıyla başarıyla eşleşenlerin yüzdesi. Bu oran, fotoğrafçıların çekim kalitesi ve kadraj başarısı hakkında geri bildirim verir.
- **Review Backlog (Onay Bekleyenler):** Yapay zekâ tarafından ara bölgede bırakılan ve acil onay bekleyen fotoğraf sayısı.
- **Depolama Kullanımı:** Kiracının üyelik planı kapsamında tanımlanan depolama limitinin (örn. 10 GB) ne kadarının doldurulduğunu gösteren canlı bar grafiği.

3.10.2. PDF Formatında Dışa Aktarma (jsPDF)

Sistem, üretilen tüm bu operasyonel istatistikleri ve audit log kayıtlarını kurumsal sunumlar veya müşteri raporlamaları için kullanılmak üzere tek tıkla PDF formatına dönüştürebilir. jsPDF kütüphanesi ile frontend tarafında dinamik olarak oluşturulan PDF raporları; stüdyonun logosunu, etkinlik bilgilerini ve özet tablolarını barındıracak şekilde profesyonelce şablonlaştırılmıştır.

3.11. Test Otomasyonu, Kalite Standartları ve Dağıtım

SnapMatch platformunun üretim ortamında kesintisiz ve hatasız çalışmasını garanti altına almak amacıyla, modern yazılım kalifikasyon süreçleri ve sürekli entegrasyon (CI/CD) standartları uygulanmıştır.

3.11.1. Test Stratejisi ve Kapsamı

Proje kapsamında üç farklı katmanda test otomasyonu gerçekleştirilmiştir:

- **Birim (Unit) Testleri:** Backend tarafında, özellikle asenkron iş kuyruğu mantığı, veri filtreleme süreçleri ve matematiksel Öklid mesafesi hesaplayan fonksiyonlar **Jest** kütüphanesiyle yazılan birim testleriyle doğrulanmıştır. Frontend tarafında ise rıza metni onay kutuları ve form validasyonları **Vitest** ile test edilmiştir.
- **Bileşen (Component) Testleri:** React arayüzündeki kamera entegrasyonu, dosya sürükle-bırak alanları ve 3B spiral galeri bileşenlerinin render başarıları **React Testing Library** ile simüle edilerek test edilmiştir.
- **Entegrasyon (Integration) Testleri:** API uç noktalarının veri tabanı ile olan ilişkileri ve presigned URL üretim başarıları uçtan uca (E2E) entegrasyon testleriyle doğrulanmıştır.

3.11.2. Dağıtım (Deployment) ve Konteynerleştirme

Platformun farklı ortamlarda (yerel geliştirme, test, canlı ortam) bağımsız ve tutarlı çalışabilmesi için **Docker** ve **Docker Compose** konfigürasyonları hazırlanmıştır. Backend, Redis ve PostgreSQL servisleri tek bir komutla konteynerize edilerek ayağa kaldırılabilir. Üretim ortamında frontend katmanı **Firebase Hosting** üzerinde yüksek hızlı CDN desteğiyle yayınlanırken; backend servisleri Node.js Docker imajları olarak bulut sunucularına dağıtılmaktadır.

3.12. Uçtan Uca Final Demo ve Doğrulama Senaryoları

SnapMatch MVP sürümünün tüm modüllerinin birbiriyle uyum içinde çalıştığını ve sistemin kararlılığını ispat etmek amacıyla, jüri ve değerlendirme kurulu önünde sergilenmek üzere 10 adımdan oluşan uçtan uca bir final demo senaryosu tasarlanmış ve başarıyla koşturulmuştur:

1. **Sisteme Giriş (Adım 1):** Fotoğrafçı Admin hesabı ile web panele giriş yapılır, boş Dashboard ekranı ve depolama limiti görselleştirilir.
2. **Etkinlik Oluşturma (Adım 2):** "Yeni Etkinlik" butonuna tıklanarak, 2026 tarihli, "Mezuniyet Töreni 2026" başlıklı bir etkinlik oluşturulur.
3. **QR Kod Çıktısı (Adım 3):** Sistem tarafından üretilen logolu QR kod ekranda gösterilir ve katılımcı arayüzünün mobil önizleme linki açılır.
4. **Katılımcı Kayıtları (Adım 4):** 5 farklı ekip üyesi, mobil cihazlarıyla QR kodu okutur, KVKK rıza metnini onaylar ve kamera üzerinden selfie çekerek yüz kayıtlarını tamamlar.
5. **Toplu Fotoğraf Yükleme (Adım 5):** Fotoğrafçı paneline geçilerek, mezuniyet töreninde çekilmiş olan ve içinde ekip üyelerinin de bulunduğu 50 adet yüksek çözünürlüklü fotoğraf toplu olarak sürüklenip bırakılır. İlerleme çubukları izlenir.
6. **Yapay Zekâ Analizinin Tetiklenmesi (Adım 6):** "Eşleştirmeyi Başlat" butonuna tıklanarak asenkron BullMQ kuyruğu tetiklenir. WebSocket üzerinden gelen gerçek zamanlı ilerleme logları panelde izlenir.
7. **Eşleşme İnceleme (Adım 7):** Yapay zekânın kararsız kaldığı (örn. 0.72 güven skorlu) eşleşmeler "Match Review" ekranında listelenir. Fotoğrafçı 3 eşleşmeyi onaylar, 1 hatalı benzerliği reddeder ve 1 adet de manuel eşleştirme tanımlar.
8. **Katılımcı Galerisi Doğrulaması (Adım 8):** Kayıtlı katılımcılardan birinin mobil ekranı açılır. Katılımcının sadece kendisinin yer aldığı ve fotoğrafçının onayladığı 12 adet mezuniyet fotoğrafına pürüzsüzce ulaştığı gösterilir.
9. **Analytics Kontrolü (Adım 9):** Dashboard'a dönülerek; toplam fotoğraf sayısı, eşleşme başarı yüzdesi ve tüketilen depolama alanı güncel grafikleri doğrulanır.
10. **Audit Log Kontrolü (Adım 10):** Activity Logs sekmesi açılarak; rıza onayları, fotoğraf yüklemeleri ve manuel onaylama eylemlerinin IP bazlı kayıt altına alındığı gösterilerek demo tamamlanır.

4. SONUÇ VE GELECEK YOL HARİTASI

Bu çalışma kapsamında, fiziksel etkinliklerin en büyük operasyonel problemlerinden biri olan "çekilen fotoğrafların katılımcılara ulaştırılması" sorununu yapay zekâ ve modern bulut teknolojileriyle çözen **SnapMatch** platformu başarıyla geliştirilmiş ve MVP başarı kriterleri eksiksiz olarak karşılanmıştır.

Proje süresince, beş kişilik yazılım ekibi modern yazılım mühendisliği pratiklerini uygulayarak çalışmıştır. Git üzerinde branch tabanlı iş akışları, kod inceleme (code review) süreçleri ve modüler sistem tasarımı sayesinde temiz, sürdürülebilir ve endüstriyel standartlarda bir kod tabanı oluşturulmuştur. NestJS og BullMQ mimarisiyle kurgulanan asenkron backend yapısı, yoğun yapay zekâ iş yükleri altında dahi sistemin kesintisiz hizmet vermesini sağlamıştır. Çok kiracılı (multi-tenant) veri modeli ise platformun gerçek bir ticari SaaS ürünü olarak konumlandırılmasının önünü açmıştır.

Gelecek Yol Haritası ve Faz-2 Geliştirmeleri

Projenin MVP sürümünün saha testlerinde elde edilen başarılar doğrultusunda, platformun ticari bir ürüne dönüştürülmesi amacıyla Faz-2 kapsamında şu geliştirmelerin yapılması planlanmaktadır:

- **Watermarking & Ödeme Sistemleri:** Katılımcılara fotoğrafların düşük çözünürlüklü ve logolu (watermarked) sürümlerini ücretsiz sunup, yüksek çözünürlüklü orijinal sürümlerini indirebilmeleri için kredi kartıyla ödeme yapabilecekleri Iyzico veya Stripe entegrasyonlarının yapılması.
- **Gelişmiş Görüntü Optimizasyonu & CDN:** Fotoğrafların küresel ölçekte milisaniyeler içinde yüklenebilmesi için AWS CloudFront gibi bir CDN (İçerik Dağıtım Ağı) entegrasyonunun tamamlanması.
- **İleri Düzey Yapay Zekâ Modelleri:** Sunucu kaynaklarının artırılmasıyla birlikte, face-api.js yerine daha derin katmanlı ve daha yüksek doğruluk oranına sahip PyTorch veya TensorFlow tabanlı custom yüz tanıma servislerinin microservice olarak sisteme entegre edilmesi.

5. KAYNAKÇA

- React 19 Official Documentation: <https://react.dev>
- Vite.js Build Tool Guide: <https://vite.dev>
- Redux Toolkit State Management Tutorial: <https://redux-toolkit.js.org>
- NestJS Progressive Node.js Framework Docs: <https://docs.nestjs.com>
- face-api.js Open Source Repository: <https://github.com/justadudewhohacks/face-api.js>
- BullMQ & Redis Queue Architecture Systems: <https://docs.bullmq.io>
- PostgreSQL Database Engine Documentation: <https://www.postgresql.org/docs>
- Firebase Data Connect Service Overview: <https://firebase.google.com/docs/data-connect>
- Tailwind CSS v4 Utility-First Framework Docs: <https://tailwindcss.com/docs>
- Three.js WebGL 3D Graphics Library Documentation: <https://threejs.org/docs>
- Sharp High Performance Node.js Image Processing: <https://sharp.pixelplumbing.com>
- Socket.IO Real-time Bidirectional Communication: <https://socket.io/docs>
- Photier Official Business Website (Benchmark Product): <https://photier.com>
- KVKK 6698 Sayılı Kişisel Verilerin Korunması Kanunu Mevzuatı: T.C. Resmî Gazete