



BLM4531 Ağ Tabanlı Teknolojiler ve Uygulamaları

PROJE TANITIM DOKÜMANTASYONU

HABİBE ALEYNA TAŞDEMİR - 22290583

1. Proje Tanımı (Özet)

Proje Adı: Kişisel Dijital Hafıza & Özet

Amaç: Kullanıcının günlük yaşamından parçaları (fotoğraf, sesli günlük, metin notu, dinlediği şarkı ve opsiyonel konum) **tek yerde** toplayıp **zaman içinde anlamlandırmak**; ay/yıl bazında görsel ve metinsel **otomatik özetler** üretmek.

Neden? (Problem Tanımı): Fotoğraflar galeriye, notlar farklı uygulamalara, müzikler ayrı platformlara dağınık. Geriye dönüp bakınca “Bu ay/ yıl ne yaptım, nasıl hissettim?” sorusunun cevabını çıkarmak zor. Bu ürün, **tek bir zaman çizelgesinde** tüm parçaları birleştirir ve **özetler** üretir.

Değer Önerisi: - **Tek akış:** Farklı formatlardaki anılar tek zaman çizelgesinde. -

Hatırlatmalı kayıt: Günlük tek tıkla ekleme → düşük sürtünme. - **Anlamlandırma:** Aylık/ yıllık özet, görsel kolaj, istatistikler. - **Gizlilik:** Veriler kişisel; paylaşım opsiyonel ve süreli.

Hedef Kullanıcılar: Öğrenciler, genç profesyoneller, içerik üreticileri ve günlük tutmak isteyip zaman bulamayan herkes.

Kilit Senaryolar: 1) Kullanıcı, gün sonunda 10–30 sn’lik **sesli günlük** kaydı ekler; ertesi gün transkript ve duygu etiketi hazır. 2) Gün içinde çektiği **fotoğrafları** seçip tek seferde yükler; otomatik tarih/etiket önerileri alır. 3) Akşam dinlediği **şarkıyı** ekler (manuel veya Spotify/Last.fm ile); ay sonu “en çok dinlediğin tür/ sanatçı” özetine yansır. 4) Ay sonunda **özet sayfasında** adet grafikleri, en aktif günler, etiket bulutu ve mini görsel kolajı görür; isterse PDF/PNG dışa aktarır.

Öne Çıkan Özellikler: - Zaman çizelgesi (gün/ hafta/ ay) + tür/ etiket filtreleri - Foto/ ses/ metin/ şarkı girişi (tek + buton akışı) - Aylık/ yıllık **istatistik ve kolaj** üretimi - (Ops.) Ses → metin transkripsiyonu ve **duygu etiketi** - Paylaşılabilir özet bağlantısı (süreli, salt-okunur)

Farklılaştırıcılar: Minimal arayüz, türler arası tek akış, çok hızlı kayıt akışı, özelleştirilebilir özet şablonları.

Kapsam Notu: Bu doküman **yalnızca Web istemcisini** kapsar. Aynı API’yi kullanacak **mobil Flutter** istemcisi ayrı bir dokümanda ele alınacaktır.

Kısıtlar & Varsayımlar: - Web istemci, tüm veriyi **REST API** üzerinden tüketecek (doğrudan DB yok). - Dosya yüklemeleri API’ye **multipart** ile yapılır; imzalı URL/stream ile görüntülenir. - Gizlilik/KVKK: Varsayılan tüm içerikler **özeldir**. Paylaşım açıkça kullanıcı aksiyonu ile ve süreli token’la yapılır.

Başarı Ölçütleri (MVP): - günde ≥ 1 kayıt ekleyen aktif kullanıcı yüzdesi (D1/D7 tutunma) - aylık özet oluşturma oranı - yükleme hatası oranı $< \%1$; ilk render < 2 sn (API hariç)

2. Kapsam

- **Web Uygulaması (ASP.NET Core)**
 - Kimlik doğrulama, içerik ekleme (foto/ses/metin/şarkı), zaman çizelgesi, filtreleme/arama, özet ekranları, ayarlar.
 - UI teknolojisi seçenekleri: **Razor Pages** (minimal, server-rendered) veya **Blazor Server/WebAssembly** (component tabanlı). Bu projede **Razor Pages + Partial Views/Tag Helpers** önerilmektedir.
- **API Entegrasyonu**
 - Web istemcisi **REST API** tüketir. Backend/API geliştirmesi sonraki fazdadır; bu fazda **Mock API** ile geliştirme yapılacaktır (WireMock.NET veya Minimal API stub).
- **Veritabanı**
 - Web katmanı **doğrudan DB erişmeyecek**, tüm veriyi API'den çekecektir. (Repository/Service adapter interface'leri ile ayrıştırma)
- **Depolama**
 - Dosya yüklemeleri **API üzerinden** yapılır; web frontu yalnızca **multipart** istek hazırlar.

3. Roller ve Kullanıcı Tipleri

- **Bireysel Kullanıcı:** İçerik oluşturur, düzenler, siler, özet raporlarını görüntüler.
- **Yönetici (opsiyonel):** Kural/limit tanımları, kullanıcı yönetimi, sistem metrikleri.

4. İşlevsel Gereksinimler

1. **Kimlik Doğrulama**
 - E-posta/şifre ile kayıt-giriş (JWT).
 - Google/Apple OIDC (opsiyonel).
2. **İçerik Ekleme**
 - **Fotoğraf** yükleme (jpg/png, boyut limiti: varsayılan 10 MB).
 - **Sesli günlük** yükleme/kayıt (m4a/mp3, süre limiti: varsayılan 60 sn).
 - **Metin notu** (markdown desteği opsiyonel).
 - **Dinlenen şarkı** kaydı (manuel giriş veya Spotify/Last.fm bağlama).
 - **Konum** (enlem/boylam, opsiyonel).
3. **Zaman Çizelgesi (Timeline)**
 - Gün/Hafta/Ay görünümü, medya önizlemeleri.
 - Etiket ve içerik türüne göre filtreleme.

4. Arama ve Etiketleme

- Başlık/açıklama/etikete göre arama.
- Etiket CRUD.

5. Aylık/Yıllık Özet

- İçerik istatistikleri (adet, en çok kullanılan etiketler, en yoğun günler, vb.).
- (Ops.) **Transkript + Duygu Analizi**: Sesli günlük → yazıya çevirme ve duygu (pozitif/nötr/negatif) skoru.
- Görsel rapor/panel; PDF/PNG dışa aktarım.

6. Paylaşım (opsiyonel)

- Özet raporu özel link ile paylaşma (salt okunur, süresi kısıtlı token).

7. Bildirimler

- Günlük hatırlatma (e-posta/web push) — saat konfigüre edilebilir.

5. İşlevsel Olmayan Gereksinimler

- **Performans**: API yanıt süresi p50 < 200 ms (metin işlemleri); dosya upload hariç.
- **Güvenlik**: JWT, HTTPS mecburi, dosya tip kontrolü, boyut limitleri, rate limit (IP başına 60 req/dk varsayılan).
- **Ölçeklenebilirlik**: Stateless API + bulut depolama; arka plan işler için queue.
- **Günlükleme/İzleme**: Serilog + OpenTelemetry (ops.).
- **Yedekleme**: DB günlük yedek, dosya replikasyonu (ops.).

6. Mimari

• Katmanlar

- **Pages/Views**: Razor Pages (*.cshtml) – sayfa yönlendirme, SEO dostu URL'ler.
- **ViewModels/DTOs**: API şemasıyla birebir uyumlu view modeller.
- **Services (HttpClient Adapters)**: IAuthService, IEntryService, ITagService, ISummaryService, IUploadService – yalnızca API çağırır.
- **Caching**: MemoryCache (kısa süreli), tarayıcı cache policy (ETag/If-None-Match).

• Navigasyon/Rotalar

- / (Dashboard/Timeline), /entries/new, /entries/{id}, /tags, /summaries/monthly, /summaries/yearly, /settings, /auth/login, /auth/register.

• Bileşenler

- **EntryCard, MediaPreview** (image/audio), **TagChips, DateRangePicker, UploadDropzone, SummaryCharts** (Chart.js/Chartist yerine plotly.js ops.)
- **Durum Yönetimi**
 - Sayfa başına ViewModel + TempData; uzun oturum verileri için LocalStorage (yalnız **JWT refresh yok**, yalnızca access token kısa ömürlü ve HttpOnly cookie kullanılacak).
- **Mock API ile Geliştirme**
 - appsettings.Development.json → ApiBaseUrl mock sunucuya işaret eder.

7. Veri Modeli (Taslak ER)

Varlıklar - User(Id, Email, PasswordHash, DisplayName, CreatedAt) - Entry(Id, UserId, Type(enum: Photo, Audio, Text, Track), Title, Note, CreatedAt, Latitude?, Longitude?) - Media(Id, EntryId, StoragePath, ContentType, SizeBytes, DurationSec?) - Tag(Id, Name) - EntryTag(EntryId, TagId) - TrackMeta(Id, EntryId, Provider(enum: Manual, Spotify, LastFm), Artist, Title, Album, ExternalId?) - Transcript(Id, EntryId, Text, Language, Confidence) - Sentiment(Id, EntryId, Score(-1..1), Label) - MonthlySummary(Id, UserId, Year, Month, JsonBlob, GeneratedAt) - YearlySummary(Id, UserId, Year, JsonBlob, GeneratedAt)

Ana İlişkiler - User 1..* Entry - Entry 0..* Media / Transcript / Sentiment / TrackMeta - Entry .. Tag (EntryTag)

8. API Sözleşmesi (Tüketim Tarafı – Web’in Beklentisi)

Not: Backend/API ayrı fazda geliştirilecek; aşağıdaki sözleşme web istemcisinin **beklediği** endpoint’leri tanımlar. Geliştirme sürecinde **Swagger/OpenAPI** ile senkron tutulacaktır.

Kimlik - POST /api/auth/register — {email, password} - POST /api/auth/login — {email, password} ⇒ HttpOnly cookie’de access token (XSRF token ayrı header) - POST /api/auth/logout

Entry - GET /api/entries?from&to&type&tag&page&pageSize - POST /api/entries — multipart: {type, title?, note?, latitude?, longitude?, files[]} - GET /api/entries/{id} - PUT /api/entries/{id} — {title?, note?, tags[]} - DELETE /api/entries/{id}

Etiketler - GET /api/tags - POST /api/tags — {name} - DELETE /api/tags/{id}

Özetler - GET /api/summaries/monthly?year&month - GET /api/summaries/yearly?year

Medya Akışı - GET /api/media/{mediaId} — Signed URL veya streaming

Hata Formatı

```
{
  "type": "https://tools.ietf.org/html/rfc9110#section-15",
  "title": "Validation failed",
  "status": 400,
  "errors": { "title": ["Required"] }
}
```

9. Güvenlik

- **JWT** access token (15–60 dk) + refresh token.
- **HTTPS** zorunlu, CORS whitelist.
- Dosya upload'larında **MIME/Signature** kontrolü, boyut limiti, antivirüs taraması (ops.).
- **Role-based** veya policy-based yetkilendirme (kullanıcı yalnızca kendi verisini görür).

10. Hata Yönetimi & Doğrulama

- Global exception middleware (ProblemDetails)
- FluentValidation ile DTO doğrulamaları
- Standart hata kodları: 400/401/403/404/413/429/500

11. Test Planı (Web)

- **Birim Testleri (UI/Services):**
 - Service adapter'ları için HttpClient fake (RichardSzalay.MockHttp).
 - ViewModel validasyonları (FluentValidation) – client-side + server-side.
- **Entegrasyon (UI + Mock API):**
 - Playwright ile e2e: login → entry ekleme → timeline görünümü → özet grafikleri.
- **Erişilebilirlik (A11y):**
 - Axe-core ile temel WCAG 2.1 AA kontrolleri, klavye ile gezinme, kontrast.
- **Performans:**
 - Lighthouse (PWA opsiyonel), p95 interaction < 200 ms (API hariç), ilk render < 2s.

12. DevOps & Dağıtım

- **Proje Yapısı**

- src/Web (ASP.NET Core Razor Pages)
- src/Web/Services (API adapter'ları)
- tests/Web.Tests, tests/Web.E2E
- **CI:** GitHub Actions – build, test, Playwright headless e2e.
- **Artefakt:** Docker image (Kestrel) + Nginx reverse proxy (ops.).
- **Konfigürasyon:** appsettings.{env}.json, ApiBaseUrl, CookiePolicy.

13. Yol Haritası

Hafta 1 — Dokümantasyon & UI Tasarım - Bu teknik şartname, wireframe'ler, bileşen envanteri

Hafta 2 — Proje İskeleti & Auth - Razor Pages projesi, layout, tema - Login/register sayfaları, HttpOnly cookie, antiforgery

Hafta 3 — Timeline & Entry Ekleme - Entry listesi (pagination, filtre), yeni entry modal/form (foto/ses/metin/şarkı) - UploadDropzone + progress bar (multipart)

Hafta 4 — Detay & Etiketler - Entry detay sayfası, TagChips, etiket filtreleri, arama

Hafta 5 — Özetler & Grafikler - Monthly/Yearly özet sayfaları, SummaryCharts (Chart.js/Plotly)

Hafta 6 — Erişilebilirlik & Performans - A11y düzeltmeleri, Lighthouse skorları, lazy-load, CDN ayarları

Hafta 7 — Test & Sertleştirme - E2E senaryoları, hata sayfaları (ProblemDetails render), global error UX

Hafta 8 — Yayın - Docker image, Nginx, ortam değişkenleri ile ApiBaseUrl geçişi, smoke test

14. Teslimatlar

- Teknik şartname (bu doküman)
- Wireframe paketleri (Timeline, Entry Form, Özetler, Ayarlar)
- API tüketim contract'ı (OpenAPI referansı)
- Kaynak kod, CI pipeline
- E2E test raporu (Playwright)
- Kullanıcı kılavuzu: kurulum, login, içerik ekleme, özet görüntüleme

15. Riskler & Varsayımlar

- Ses transkripsiyonu ve duygu analizi için 3P servis ihtiyacı (API maliyeti).
- Dosya depolamada maliyet ve kota yönetimi.
- Spotify/Last.fm OAuth akışı için geliştirici hesapları gerekir.
- Kişisel veri gizliliği (KVKK/GDPR) — paylaşım linkleri süreli ve iptal edilebilir olmalı.

16. Tercih Edilen Yığın (Web – Özet)

- **Dil/Framework:** .NET 9 / ASP.NET Core Razor Pages
- **UI:** TailwindCSS + Partial Views; komponent: TagHelpers
- **Grafikler:** Chart.js veya Plotly (bundle boyutu dikkate alınarak)
- **Auth:** Cookie-based (HttpOnly) + antiforgery token
- **HTTP:** HttpClient + Polly (retry/backoff) – sadece idempotent GET’lerde retry
- **Ayarlar:** ApiBaseUrl, MaxUploadSizeMB, AllowedMimeTypes

17. Kabul Kriterleri (Web)

- Kullanıcı web arayüzünden **foto/ses/metin/şarkı** girişlerinden en az birini ekleyebilmeli.
- **Timeline** sayfası tarih aralığı ve etiket ile filtrelenebilmelidir.
- **Aylık özet** sayfası seçilen ay için istatistikleri **grafik** olarak gösterebilmelidir.
- Tüm istekler **kimlik doğrulaması** gerektirmeli, HttpOnly cookie kullanılmalıdır.
- Dosya yükleme sırasında **progress** görünmeli, boyut/tip doğrulamaları istemci tarafında yapılmalıdır.
- Erişilebilirlik kontrolünde temel **WCAG 2.1 AA** maddeleri karşılanmalıdır.