Ygg Proje Dokümantasyonu

# Proje Amacı ve Sorun Tanımı

Ygg, Pomodoro Tekniği ile kişiselleştirilmiş müziği birleştirerek insanların daha iyi odaklanmasını sağlamayı amaçlayan bir verimlilik uygulamasıdır. Pomodoro Tekniği, çalışma süresini (genellikle 25 dakika) kısa molalarla bölerek odaklanmayı artırmaya yönelik popüler bir zaman yönetimi yöntemidir. Ygg, odaklanmayı sürdürme ve motivasyonu yüksek tutma konusunda yaşanan zorlukları gidermek amacıyla geliştirilmiştir. Birçok kişi, çalışma sırasında odaklanmayı artırmak için müzik dinler, ancak zamanlayıcı ve müzik uygulamaları arasında geçiş yapmak dikkat dağıtıcı olabilmektedir. Ygg, müzik akışı entegrasyonunu Pomodoro zamanlayıcısına doğrudan ekleyerek, kullanıcıların minimum dikkat dağıtımıyla çalışma ritimlerini korumasını sağlar.

Özet: Projenin amacı, zaman yönetimi ile müziğin uyumunu sağlayarak verimliliği ve odaklanmayı artırmaktır.

Çözdüğü Sorun: Ygg, dijital dikkat dağıtıcılarla dolu bir dünyada odaklanmayı sürdürme sorununu ele alır. Geleneksel Pomodoro zamanlayıcıları yalnızca zamanı ölçerken, Ygg entegre müzik desteği sayesinde kullanıcıların favori müziklerini dinleyerek çalışma seanslarını yönetmelerine olanak tanır. Ayrıca, seansların analiziyle verimlilik artışına yönelik geri bildirim sunar.

Benzersiz Özellikler ve Yenilikler:  
• Müzik Servisi Entegrasyonu: Spotify, Apple Music ve YouTube Music hesaplarının entegrasyonu sayesinde kullanıcılar, kişisel çalma listelerini kullanabilir.  
• Veri Analizi Destekli Odak Çalma Listeleri: Kullanıcının geçmiş verilerine göre akıllı öneriler sunar.  
• Verimlilik Analitiği: Seans verileri grafikler ve istatistiklerle sunularak kullanıcının performansı takip edilir.  
• Özelleştirilebilir Çalışma Ortamı: Tema düzenleyici ile arayüz, renk, yazı tipi, arka plan gibi öğeler kişiselleştirilebilir.  
• Topluluk ve Paylaşım: Kullanıcılar, oluşturdukları temaları paylaşabilir, beğenebilir ve diğerlerinden ilham alabilir.

# Teknoloji Yığını

Ygg, hem web hem de mobil platformlarda kesintisiz bir kullanıcı deneyimi sunmak için modern teknolojiler kullanılarak geliştirilmiştir.

## Landing Page (Web):

• Next.js (React Çatısı):  
 - Seçilme Nedeni: Sunucu tarafı render (SSR), kolay sayfa yönlendirmesi ve SEO dostu yapısı.  
 - Artıları: Hızlı performans, SEO uyumluluğu, karmaşık arayüzler oluşturma.  
 - Eksileri: Basit projeler için ekstra karmaşıklık, Node.js gereksinimi.

• Tailwind CSS:  
 - Seçilme Nedeni: Hızlı ve tutarlı tasarım oluşturmayı sağlar.  
 - Artıları: Hızlı iterasyon, tutarlı stil, küçük paket boyutu.  
 - Eksileri: HTML içerisinde yardımcı sınıfların yoğunluğu, öğrenme eğrisi.

• Vercel:  
 - Seçilme Nedeni: Next.js ile uyumlu, otomatik dağıtım ve global CDN desteği.  
 - Artıları: Kolay sürekli entegrasyon, sunucusuz fonksiyon desteği.  
 - Eksileri: Yüksek trafik durumunda maliyet artışı, ekosisteme bağımlılık.

## Mobil:

• Flutter:  
 - Seçilme Nedeni: Tek kod tabanı ile iOS ve Android’de yerel uygulama geliştirme imkanı.  
 - Artıları: Tek kod tabanı, "hot reload" ile hızlı geliştirme, çekici arayüzler.  
 - Eksileri: Uygulama boyutu, Dart dilinin öğrenilmesi, platforma özgü entegrasyon zorlukları.

## Backend & Servisler:

• Python (Backend API):  
 - Seçilme Nedeni: Hızlı geliştirme, zengin kütüphane desteği, deneyim faktörü.  
 - Artıları: Hızlı prototipleme, geniş ekosistem.  
 - Eksileri: Yorumlanmış dil olması nedeniyle bazı performans kısıtlamaları.

• Firebase:  
 - Seçilme Nedeni: Gerçek zamanlı veritabanı ve hızlı kimlik doğrulama desteği sağlar.  
 - Artıları: Anında veri senkronizasyonu, güvenli kimlik doğrulama, düşük maliyet.  
 - Eksileri: NoSQL yapının bazı kısıtlamaları.

• Müzik Servisleri API’ları:  
 - Seçilme Nedeni: Kullanıcıların mevcut müzik kütüphaneleriyle entegre olarak zengin müzik deneyimi sunmak.  
 - Artıları: Kişisel çalma listeleri, geniş müzik erişimi.  
 - Eksileri: Üçüncü taraf API’lara bağımlılık, farklı yetkilendirme akışları ve API kısıtlamaları.

# 14 Haftalık Geliştirme Planı

Geliştirme planı, web, mobil ve backend çalışmalarının paralel ilerleyeceği şekilde düzenlenmiş olup, Spotify entegrasyonu ilk aşamada öncelikli olarak ele alınacak; diğer müzik servisleri (Apple Music, YouTube Music) daha sonraki haftalarda entegre edilecektir.  
  
Hafta 1: Environment Setups  
• Gerekli uygulamaların ortamlarının kurulması ve kullanılacak teknoloji yığınının (tech stack) belirlenmesi.  
 - Web: Python-Flask ortamının kurulması.  
 - Mobil: Flutter ortamının kurulması.  
 - Backend: Python ortamının oluşturulması, Firebase yapılandırması ve temel ayarların yapılması.  
  
Hafta 2: Development Designs  
• Geliştirme süreci için tasarım dilinin (design language) belirlenmesi; giriş (login) ve ana sayfaların (main pages) tasarımının yapılması.  
 - Web: Wireframe tasarımlarının hazırlanması ve temel düzenin oluşturulması.  
 - Mobil: Ana ekran tasarımları ve placeholder sayfaların hazırlanması.  
  
Hafta 3: Firebase  
• Saklanacak veri türlerinin belirlenmesi ve Firebase ortamının kurulması.  
 - Backend: Veri modelleri, API uç noktalarının planlanması ve Firebase entegrasyonunun yapılması.  
  
Hafta 4: Spotify Account Link  
• Kullanıcıların Spotify hesaplarını uygulamaya bağlaması için Spotify OAuth akışının uygulanması; Spotify implementasyonu üzerine araştırma yapılması.  
 - Web: Spotify OAuth akışının ve token işlemlerinin gerçekleştirilmesi.  
 - Mobil: Spotify entegrasyonu için uygun plugin’lerin kullanılması.  
 - Backend: Spotify API çağrılarını yönetecek uç noktanın oluşturulması ve token yenileme mekanizmasının entegre edilmesi.  
  
Hafta 5: Connecting and Playing Spotify Playlists  
• Kullanıcıların istedikleri Spotify çalma listelerini seçip çalabilmelerini sağlayacak Spotify entegrasyonunun uygulanması.  
 - Web, Mobil ve Backend: Spotify üzerinden çalma listesi seçimi, oynatma kontrolü ve hata yönetimi mekanizmalarının kurulması.  
  
Hafta 6: Pomodoro Timer Implementation  
• Pomodoro zamanlayıcı mantığının uygulanması; kullanıcıların çalışma süresi boyunca seçtikleri çalma listelerini kullanabilmeleri sağlanacaktır.  
 - Web: Başlat, duraklat, sıfırla işlevlerinin uygulanması.  
 - Mobil: Flutter’da benzer zamanlayıcı mantığının uygulanması.  
 - Backend: Seans tamamlandığında veri kaydı için API geliştirilmesi.  
  
Hafta 7: Youtube Music Implementation  
• YouTube Music entegrasyonu üzerine araştırma yapılması ve Spotify ile farklarının incelenmesi.  
 - Araştırma: YouTube Music API dokümantasyonu ve OAuth akışları hakkında bilgi toplanması.  
  
Hafta 8: Youtube Music Account and Playlist Operations  
• YouTube Music entegrasyonunun uygulanarak, kullanıcıların istedikleri çalma listelerini seçip oynatabilmelerinin sağlanması.  
 - Web, Mobil ve Backend: YouTube Music için gerekli API anahtarlarının alınması, OAuth entegrasyonu ve oynatma kontrollerinin uygulanması.  
  
Hafta 9: Apple Music Implementation  
• Apple Music entegrasyonu üzerine araştırma yapılması ve Spotify ile farklarının belirlenmesi.  
 - Araştırma: Apple Music API (MusicKit) dokümantasyonunun incelenmesi ve implementasyon stratejilerinin belirlenmesi.  
  
Hafta 10: Apple Music Account and Playlist Operations  
• Apple Music entegrasyonunun uygulanması; kullanıcıların Apple Music hesapları üzerinden çalma listelerini seçip oynatabilmeleri sağlanacaktır.  
 - Web, Mobil ve Backend: Apple Music için gerekli entegrasyonların ve oynatma kontrollerinin uygulanması.  
  
Hafta 11: Front End Designs Figma and AI Improvement Discussion  
• Nihai ürün tasarımlarının Figma üzerinden belirlenmesi; AI ve diğer iyileştirmeler üzerine tartışmalar yapılması.  
 - Tasarım: UI/UX düzenlemeleri için Figma üzerinden son tasarım kararı alınması.  
 - Tartışma: AI uygulama yöntemleri ve geliştirme iyileştirmeleri üzerine tartışmalar.  
  
Hafta 12: Front End Designs Implementation  
• Figma üzerinden belirlenen tasarımların uygulamaya entegre edilmesi; kullanıcı arayüzünün son haline getirilmesi.  
 - Web: Tasarımın HTML/CSS/JavaScript ile uygulanması.  
 - Mobil: Flutter’da tasarımın uygulanması.  
  
Hafta 13: N/A or TBA  
• AI uygulaması ve diğer geliştirmeler için planlama; detaylar kararlaştırıldıktan sonra uygulanacaktır.  
  
Hafta 14: N/A or TBA  
• AI uygulaması ve diğer geliştirmelerin tamamlanması, son testler, dağıtım ve sunum hazırlıkları; detaylar belirlendikten sonra eklenecektir.

# Benzer Proje Karşılaştırması

1. Pomtify – Pomodoro + Spotify Web Uygulaması:  
 - Özellikler: Spotify entegrasyonlu temel zamanlayıcı.  
 - Artıları: Basit kullanım, doğrudan müzik kontrolü.  
 - Eksileri: Fonksiyonellik sınırlı, yalnızca Spotify destekli.  
 - Ygg’nin Farkı: Birden fazla müzik servisini destekleyerek; tema, verimlilik analitiği ve topluluk özellikleriyle kapsamlı bir deneyim sunar.  
  
2. Astrostation – Lofi Temalı Verimlilik Zamanlayıcısı:  
 - Özellikler: Estetik, lofi temalı arayüz ve Spotify entegrasyonu.  
 - Artıları: Görsel çekicilik, rahatlatıcı deneyim.  
 - Eksileri: Yalnızca Spotify, mobil desteği yok, gelişmiş özellikler bulunmuyor.  
 - Ygg’nin Farkı: Web ve mobil platformlarda çalışır, çoklu müzik servisi entegrasyonu ve gelişmiş analiz/özelleştirme seçenekleri sunar.  
  
3. Focus@Will – Odaklanmayı Artıran Müzik Servisi:  
 - Özellikler: Bilimsel temelli optimize edilmiş müzik servisi ve zamanlayıcı.  
 - Artıları: Odaklanmayı artıran müzik içerikleri.  
 - Eksileri: Abonelik tabanlı, kullanıcının kendi müziklerini dinleme esnekliği yok.  
 - Ygg’nin Farkı: Kullanıcının kendi müzik kaynaklarını kullanmasına olanak tanır; ücretsiz ve çok yönlü özellikler sunar.  
  
4. Windows 11 “Focus Sessions”:  
 - Özellikler: İşletim sistemi entegrasyonlu Pomodoro zamanlayıcısı, Spotify ve Microsoft To-Do entegrasyonu.  
 - Artıları: OS ile entegre, ek yazılım gerektirmez.  
 - Eksileri: Sadece Windows platformuna özgü, sınırlı özelleştirme.  
 - Ygg’nin Farkı: Platform bağımsızlığı (web ve mobil desteği) ve geniş müzik servisi entegrasyonu ile daha esnek bir deneyim sunar.  
  
5. Diğer Projeler:  
 - Birçok Pomodoro uygulaması yalnızca yerleşik sesler veya basit arayüzler sunarken, Ygg; müzik entegrasyonu, tema düzenleyici, önerileri ve topluluk özellikleriyle fark yaratır.

# Ek Bölüm: Web Uygulaması (Python-Flask)

Mevcut Next.js tabanlı açılış sayfası yalnızca projenin tanıtımını yaparken, kullanıcıların Pomodoro uygulamasını tüm özellikleriyle kullanabilecekleri yeni bir web uygulaması Python-Flask çatısı ile geliştirilecektir.

Amaç:  
• Tam fonksiyonlu Pomodoro deneyimi sunmak; zamanlayıcı, müzik entegrasyonu, tema düzenleyici, verimlilik analitiği ve topluluk özellikleri gibi tüm modüller web ortamında erişilebilir olacaktır.

Mimari ve Teknoloji Kararları:  
• Backend: Python-Flask tercih edilecektir. Hızlı geliştirme, RESTful API’ler ve esnek yapı sunması nedeniyle tercih edilir. Entegrasyonlar, mevcut Python backend modülleri yeniden kullanılacak; özellikle öncelikli olarak Spotify entegrasyonu yapılacak, ardından diğer müzik servisleri entegre edilecektir.  
• Frontend: Flask tabanlı sunucu tarafı render (SSR) ile dinamik HTML/CSS sayfaları oluşturulacaktır. Kullanıcı dostu arayüz için Bootstrap veya Tailwind CSS tercih edilebilir.  
• Veritabanı ve Kimlik Doğrulama: Firebase veya alternatif veritabanı (ör. PostgreSQL, MySQL) kullanılarak kullanıcı verileri, seans kayıtları ve tema verileri yönetilecektir. JWT ya da benzeri token bazlı kimlik doğrulama yöntemi ile güvenlik sağlanacaktır.

Özellikler:  
• Pomodoro Zamanlayıcısı: Başlat/duraklat/sıfırla işlevlerinin yanı sıra, görsel/sesli bildirimler ve seans ayarlarının özelleştirilmesi.  
• Müzik Entegrasyonu: İlk olarak Spotify entegrasyonu gerçekleştirilecek; sonrasında Apple Music ve YouTube Music eklenerek, OAuth akışları ve otomatik çalma kontrolü sağlanacaktır.  
• Tema Düzenleyici: Kullanıcıların arayüzü özelleştirebileceği, canlı ön izlemeli tema düzenleyici.  
• Verimlilik Analitiği: Seans verilerinin grafik ve istatistiklerle sunulması.  
• Topluluk Özellikleri: Tema paylaşımı, beğenme ve sosyal etkileşimlerin sağlanması.

Geliştirme Süreci:  
1. Planlama ve Tasarım: Proje kurulumu, veri modelleri ve API uç noktalarının belirlenmesi.  
2. Temel Özellikler: Pomodoro zamanlayıcısı, öncelikli olarak Spotify entegrasyonu ve kullanıcı doğrulamasının uygulanması.  
3. İleri Özellikler: Tema düzenleyici, verimlilik analitiği ve topluluk modülünün entegrasyonu; diğer müzik servislerinin eklenmesi.  
4. Test ve Optimizasyon: Kapsamlı testler, performans iyileştirmeleri ve entegrasyon kontrolleri.  
5. Dağıtım: Uygulamanın uygun bir sunucuda (örn. Heroku, AWS, DigitalOcean) canlıya alınması; Next.js açılış sayfasıyla entegrasyonunun sağlanması.

Sonuç:  
Yeni Python-Flask tabanlı web uygulaması, Ygg’nin mobil uygulamasıyla uyumlu, tüm fonksiyonların web üzerinden de kullanılabildiği tam entegre bir Pomodoro deneyimi sunacaktır. Böylece, kullanıcılar hem web hem de mobil platformlarda aynı kapsamlı özelliklere erişebilecektir.

# Ek Bölüm: Risk Yönetimi – Entegrasyonlar ve Önlemler

Amaç:  
Üçüncü taraf servisler (özellikle Spotify, Apple Music, YouTube Music) ile entegrasyon sırasında ortaya çıkabilecek riskleri belirleyip, bu risklere karşı önleyici stratejiler geliştirmektir.

Risk Faktörleri ve Değerlendirme:  
• API Erişim Kısıtlamaları ve Rate Limiting:  
 - Risk: Üçüncü taraf API’ların belirli çağrı limitleri, yoğun kullanım dönemlerinde erişim kısıtlamalarına neden olabilir.  
 - Önlemler:  
 • API çağrılarının önbelleğe alınması, gereksiz tekrarların önlenmesi.  
 • Rate limit’lere uygun hata yönetimi ve loglama mekanizmaları.  
 • Back-off stratejileri ve otomatik yeniden deneme mekanizmaları.  
  
• Yetkilendirme ve Token Yönetimi:  
 - Risk: OAuth akışlarında token’ların süresi dolması, hatalı yenileme işlemleri veya yetkilendirme aksaklıkları entegrasyonun kesilmesine neden olabilir.  
 - Önlemler:  
 • Güvenilir token yenileme (refresh) mekanizmalarının uygulanması.  
 • Hata ve istisna yönetimi, başarısız çağrılar için otomatik yeniden deneme.  
 • Token’ların güvenli bir şekilde saklanması ve şifrelenmesi.  
  
• API Değişiklikleri ve Uyum Sorunları:  
 - Risk: Üçüncü taraf API’larında yapılacak güncellemeler, endpoint değişiklikleri veya kullanım şartlarının revize edilmesi entegrasyonlarda aksamalara neden olabilir.  
 - Önlemler:  
 • API dokümantasyonlarının düzenli takibi ve entegrasyon modüllerinin güncel tutulması.  
 • Modüler ve esnek kod yapısı ile hızlı adaptasyon.  
 • Alternatif API sağlayıcıları veya yedek planların hazırlanması.  
  
• Veri Güvenliği ve Gizlilik:  
 - Risk: Üçüncü taraf servislerle entegrasyon sırasında kullanıcı verilerinin güvenliği risk altında olabilir.  
 - Önlemler:  
 • HTTPS gibi güvenli iletişim protokollerinin kullanılması.  
 • Veri aktarımı ve saklamada end-to-end şifreleme.  
 • Kullanıcı onayının alınması ve gizlilik politikalarının netleştirilmesi.  
  
• Bağlantı ve Altyapı Kesintileri:  
 - Risk: Üçüncü taraf servislerin sunucu veya ağ sorunları, entegrasyonların kesintiye uğramasına neden olabilir.  
 - Önlemler:  
 • Yedekleme ve failover mekanizmalarının planlanması.  
 • Servis kesintilerinde kullanıcıya bilgilendirici mesajlar ve alternatif çözümler sunulması.  
 • Sürekli izleme (monitoring) sistemlerinin kurulması ve anlık müdahale planlarının oluşturulması.  
  
Genel Strateji:  
Entegrasyon modülleri, esnek, modüler ve ölçeklenebilir olacak şekilde geliştirilecek; düzenli testler, kod incelemeleri ve bakım çalışmaları ile risklerin minimize edilmesi hedeflenecektir.

# Bağlantılar, Makale Linkleri ve GitHub Linkleri

Ygg Proje Sitesi: https://pomodoro.yggbranch.dev/  
  
GitHub Link: https://github.com/cagatay-softgineer/ygg-pomodoro  
  
Pomodoro Tekniği Hakkında: https://francescocirillo.com/pages/pomodoro-technique  
  
Pomtify (Benzer Proje): https://pomtify.com/  
  
Astrostation (Benzer Proje): https://astrostation.me/  
  
Focus@Will (Benzer Proje): https://www.focusatwill.com/  
  
Windows 11 Focus Sessions (Benzer Proje): https://thenextweb.com/news/windows-11-focus-timer-with-spotify  
  
Spotify API Dokümantasyonu: https://developer.spotify.com/documentation/web-api/  
  
Apple Music API (MusicKit): https://developer.apple.com/documentation/musickitjs  
  
YouTube Data API Dokümanları: https://developers.google.com/youtube/v3  
  
Flutter Dokümantasyonu: https://flutter.dev/docs  
  
Firebase Dokümantasyonu – Web & Flutter: https://firebase.google.com/docs