



**softITo 4. DÖNEM 1. GRUP
FRONTEND GELİŞTİRME BİTİRME PROJESİ RAPORU**

**Deutschlicht — Yapay Zekâ Destekli ve Kişiselleştirilmiş
Almanca Öğrenme Platformu**

Ayşenur Akgün
(Bireysel Proje)

Temmuz, 2026
softITo Yazılım Bilişim Akademisi, İstanbul

İÇİNDEKİLER

1. GİRİŞ	3
1.1. Projenin Amacı	3
1.2. Proje Kısıtları	3
2. MATERYAL VE YÖNTEM	4
2.1. React 19 ve Vite 8	4
2.2. Redux Toolkit ile Durum Yönetimi	4
2.3. Bootstrap 5 ve Sass ile Arayüz Tasarımı	4
2.4. Form Yönetimi ve Bildirim Altyapısı	5
2.5. Recharts, Framer Motion ve Lucide React	5
2.6. AI Entegrasyonu: Qwen AI & OpenRouter / Local Web-LLM	5
2.7. Proje Mimarisi ve Dosya Yapısı	5
2.8. Geliştirme Ortamı ve Çalıştırma	6
3. UYGULAMA	6
3.1. Giriş, Kayıt Ekranları ve Onboarding Akışı	6
3.2. Kullanıcı Kontrol Paneli (Dashboard) ve İlerleme Takibi	6
3.3. Eğitim Modülleri	7
3.3.1. Dersler ve Konu Anlatımı Arayüzü	7
3.3.2. Kelime Öğrenme ve Flashcard (SRS) Sistemi	7
3.3.3. Okuma ve Dinleme Portalı (Reading & Listening)	7
3.3.4. Quiz ve İnteraktif Sınav Modülü	7
3.4. AI Destekli Pratik ve Yazma Değerlendirici (AI Writing Evaluator & Chat)	8
4. SONUÇ	8

1. GİRİŞ

Kişiselleştirilmiş eğitim ve yapay zekânın dil öğrenme süreçlerine entegrasyonu, günümüz eğitim teknolojilerinin (EdTech) en dinamik alanlarından biridir. Geleneksel dil öğrenme yöntemleri, her öğrencinin aynı hızda ve aynı materyalle öğrenmesini varsayarken; bireysel eksiklikleri, öğrenme stillerini ve motivasyon unsurlarını göz ardı etmektedir. Özellikle Almanca gibi kuralları ve kelime artikelleri (der/die/das) karmaşık olan bir dili öğrenirken, kullanıcıların sürekli pratik yapması, hatalarından anında dönüt alması ve kendi hedeflerine uygun kişisel bir çalışma programı takip etmesi büyük önem taşımaktadır.

Bu gereksinimler doğrultusunda geliştirilen **Deutschlicht**, yapay zekâ destekli, modern, kullanıcı dostu ve kişiselleştirilmiş bir Almanca öğrenme platformudur. Uygulama, kullanıcıların Avrupa Ortak Dil Referans Çerçevesi (CEFR) standartlarında, A1 seviyesinden C2 seviyesine kadar interaktif bir eğitim süreci geçirmelerini amaçlar.

1.1. Projenin Amacı

Projenin temel amacı; dil öğrenenlerin kelime haznesini geliştirebildiği, dil bilgisi kurallarını interaktif derslerle pekiştirebildiği, okuma ve dinleme yetkinliklerini test edebildiği ve en önemlisi kendi tarayıcısında çalışan veya harici API'ler aracılığıyla desteklenen bir yapay zekâ öğretmeninden anında yazım ve dil bilgisi düzeltmeleri alabildiği uçtan uca bir frontend uygulaması ortaya koymaktır.

Bu doğrultuda projenin özel hedefleri şunlardır:

Kişiselleştirilmiş Onboarding: Kullanıcının seviyesini, öğrenme amacını ve günlük çalışma hedeflerini analiz ederek yapay zekâ destekli bir başlangıç profili oluşturmak.

Oyunlaştırma (Gamification): Streak (öğrenme serisi), XP puanları, seviye atlama ve başarı rozetleri (Achievements) gibi mekaniklerle kullanıcının motivasyonunu sürekli yüksek tutmak.

Aralıklı Tekrar Sistemi (SRS): Öğrenilen veya favorilere eklenen kelimelerin, unutma eğrisine uygun zaman aralıklarında flashcardlar aracılığıyla kullanıcıya tekrar ettirilmesi.

AI Asistanı Entegrasyonu: Kullanıcının yazdığı Almanca metinleri analiz edip hataları belirten, Türkçe dil bilgisi açıklamaları sunan, okuma parçalarındaki bilinmeyen kavramları açıklayan bir yapay zekâ motoru geliştirmek.

Dinamik Raporlama: Kullanıcının haftalık çalışma süresini ve kazandığı XP puanlarını grafikler eşliğinde görselleştirmek.

1.2. Proje Kısıtları

Uygulama, frontend geliştirme eğitim projesi kapsamında tasarlanmış olup şu kısıtlamalara sahiptir:

Veri Saklama: Gerçek bir uzak veritabanı (PostgreSQL, MongoDB vb.) yerine, uygulamanın durum yönetimi Redux Toolkit ve tarayıcının yerel belleğini kullanan redux-persist kütüphanesi ile simüle edilmektedir. Böylece kullanıcı verileri, ders ilerlemeleri ve istatistikler tarayıcı hafızasında saklanmaktadır.

Kimlik Doğrulama: Üretim seviyesi bir Auth sağlayıcısı (OAuth2, Firebase Auth vb.) yerine, Redux store üzerinde kurgulanan demo/simüle kayıt ve giriş akışları tercih edilmiştir.

AI Sunucu Bağlantısı: Yapay zekâ özellikleri, OpenRouter API üzerinden ücretsiz Qwen AI modellerine istek atarak çalışmaktadır. İnternet bağlantısı kesildiğinde veya API kota sınırlarına ulaşıldığında, sistemin kararlılığını korumak adına aiService içinde tanımlanan mock veritabanı motoru devreye girerek çevrimdışı/simüle yanıtlar sağlamaktadır.

2. MATERYAL VE YÖNTEM

Projenin teknik altyapısı, modern web standartlarına, yüksek performanslı bileşen yapılarına ve akıcı kullanıcı deneyimi ilkelerine göre seçilmiş kütüphanelerden oluşmaktadır.

2.1. React 19 ve Vite 8

Uygulamanın arayüzü, bileşen (component) tabanlı yapısı ve verimli DOM güncelleme yetenekleri nedeniyle React 19 ile geliştirilmiştir. React 19'un getirdiği performans iyileştirmeleri ve güncel yaşam döngüsü kuralları kod kalitesini artırmıştır. Geliştirme ortamı ve derleme aracı olarak Vite 8 tercih edilmiştir. Vite 8; hızlı ön bellekleme (caching), anında HMR (Hot Module Replacement) desteği ve üretim aşamasında (build) son derece optimize edilmiş çıktı sunması sebebiyle projede geliştirme hızını maksimuma çıkarmıştır.

2.2. Redux Toolkit ile Durum Yönetimi

Deutschlicht'in en kritik süreçlerinden biri olan durum yönetimi (state management), Redux Toolkit ile merkezi bir yapıda kurulmuştur. Uygulamada yer alan global durum dilimleri (slices) şu şekildedir:

authSlice: Kullanıcı profili, seviyesi, streak durumu, günlük hedefleri ve giriş durumu yönetilir.

lessonSlice: Seviyelere göre ders listeleri, derslerin tamamlanma durumları ve favoriler saklanır.

vocabularySlice: Kullanıcının kelime kütüphanesi, favori kelimeleri ve SRS algoritması verileri tutulur.

quizSlice: Quiz sonuçları, başarı oranları ve geçmiş test verileri izlenir.

aiSlice: Yapay zekâ zayıf nokta analizleri ve kişisel öneri planları yönetilir.

statisticsSlice: Günlük kazanılan XP değerleri ve haftalık aktivite günlüğü takip edilir.

redux-persist entegrasyonu sayesinde kullanıcı sayfayı yenilese dahi öğrenme geçmişi ve oturum bilgileri kaybolmamaktadır.

2.3. Bootstrap 5 ve Sass ile Arayüz Tasarımı

Arayüz tasarımında, responsive grid yapısı ve esnek CSS sınıfları sağlaması amacıyla Bootstrap 5 kullanılmıştır. Ancak projenin özgün görsel kimliğini yansıtmak adına Bootstrap stilleri, Sass bileşenleri ve özel yazılmış modern CSS değişkenleri ile tamamen özelleştirilmiştir. Uygulamada modern bir cam morfolojisi (Glassmorphism) stili benimsenmiştir. backdrop-filter: blur(), yarı şeffaf

renk geçişleri (gradients) ve ince border tasarımları kullanılarak karanlık mod (dark theme) odaklı, gözü yormayan premium bir arayüz dili oluşturulmuştur.

2.4. Form Yönetimi ve Bildirim Altyapısı

Kullanıcı giriş, kayıt ve onboarding süreçlerindeki form verileri React kontrollü bileşen durumları (useState) ile yönetilmektedir. Form alanlarının doğruluğu, boş geçilemezlik durumları ve şifre kuralları istemci tarafında anlık olarak doğrulanmakta; kullanıcılara geri bildirimler react-hot-toast kütüphanesiyle hazırlanan şık, yarı saydam bildirim balonları ile sunulmaktadır.

2.5. Recharts, Framer Motion ve Lucide React

Recharts: Dashboard ekranında yer alan haftalık ilerleme grafiğinde kullanılmıştır. Esnek, responsive ve SVG tabanlı alan grafiği (AreaChart) yapısıyla kullanıcının günlük XP kazanımları görselleştirilmiştir.

Framer Motion: Sayfa geçişleri, kelime kartlarının flip (ters yüz olma) animasyonları ve AI chat mesajlarının ekrana yumuşakça yansması için kullanılmıştır. Mikro animasyonlar sayesinde premium uygulama hissi desteklenmiştir.

Lucide React: Uygulamanın tamamında tutarlı, minimalist ve modern çizgi ikon seti olarak tercih edilmiştir.

2.6. AI Entegrasyonu: Qwen AI & OpenRouter / Local Web-LLM

Deutschlicht'in yapay zekâ yetenekleri iki katmanlı olarak planlanmıştır:

OpenRouter API Bağlantısı: Çevrimiçi modda, OpenRouter üzerinden ücretsiz sunulan Qwen AI API'sine istekler gönderilir. Gelişmiş prompt mühendisliği (prompt engineering) teknikleriyle modelden dönen yanıtların her zaman geçerli bir JSON formatında olması sağlanır. Bu JSON verileri frontend tarafından parse edilerek yapılandırılmış arayüz elemanlarına dönüştürülür.

Yerel Mock Engine: API anahtarı girilmediğinde veya kota aşıldığında, aiService.js içerisinde bulunan ve kullanıcının girdiği yaygın gramer hatalarını (örneğin "ich gehen" veya "mit den auto") algılayarak anında düzeltmeler yapan özel bir kural tabanlı simülasyon motoru çalışır.

2.7. Proje Mimarisi ve Dosya Yapısı

Proje, sorumlulukların net biçimde ayrıştırıldığı katmanlı (layered) bir klasör mimarisine sahiptir:

```

src/
├─ app/                # Redux Store yapılandırması ve kalıcılık ayarları
├─ assets/             # Resimler, logolar ve statik medya dosyaları
├─ components/         # Yeniden kullanılabilir arayüz bileşenleri (örnek: GermanTutor.jsx)
├─ data/               # Dersler, quizler ve kelimeler için mock veri setleri
├─ features/           # Redux slice dosyaları (authSlice.js, aiSlice.js vb.)
├─ layouts/            # Genel sayfa düzenleri (MainLayout.jsx - Sidebar ve Header)
├─ pages/              # Sayfa seviyesindeki bileşenler (Dashboard, Vocabulary, Lessons vb.)
├─ routes/             # Yönlendirme şeması ve yetki kontrolleri (AppRoutes.jsx)
├─ services/           # Yapay zekâ ve veri servis katmanı (aiService.js)
├─ styles/             # Sass dosyaları, global CSS değişkenleri (variables.css)
└─ utils/              # Yardımcı fonksiyonlar (konfeti patlatma efektleri vb.)

```

2.8. Geliştirme Ortamı ve Çalıştırma

Projenin çalıştırılması için makinede Node.js (v18+) kurulu olması gerekmektedir. Geliştirme ortamını ayağa kaldırmak için aşağıdaki komutlar sırasıyla çalıştırılır:

Bağımlılıkların yüklenmesi:

```
npm install
```

Uygulamanın geliştirici modunda başlatılması:

```
npm run dev
```

Uygulama varsayılan olarak **http://localhost:5173** adresi üzerinden yerel sunucuda hizmet vermektedir.

3. UYGULAMA

Bu bölümde uygulamanın çekirdek ekranları ve bu ekranların teknik işleyişleri detaylandırılmıştır.

3.1. Giriş, Kayıt Ekranları ve Onboarding Akışı

Giriş ve Kayıt (Login / Register): Kullanıcı, e-posta adresi ve şifresiyle güvenli bir şekilde giriş yapabilir. Başarılı girişte authSlice üzerinden kullanıcı durumu aktif edilir ve tarayıcı yönlendirilir.

Profil Yapılandırma (Onboarding): İlk defa kayıt olan kullanıcılar doğrudan Onboarding ekranına alınır. Bu ekranda kullanıcıdan Almanca seviyesi (A1, A2, B1, B2 vb.), günlük çalışma hedefi (dakika bazlı) ve öğrenme motivasyonu (iş, eğitim, seyahat vb.) bilgileri istenir. Bu veriler Redux'ta kaydedilerek kullanıcının kişisel hedefleri ve dashboard parametreleri hesaplanır.

3.2. Kullanıcı Kontrol Paneli (Dashboard) ve İlerleme Takibi

Dashboard, kullanıcının dil öğrenme yolculuğunu takip ettiği merkez üssüdür:

Günlük Hedef Göstergesi: Kullanıcının günlük çalışma hedefine ne kadar yaklaştığını gösteren dinamik bir SVG dairesel ilerleme çubuğu (gauge) içerir.

Streak & XP Takibi: Kullanıcının kaç gündür aralıksız çalıştığını gösteren alev ikonu (Streak) ve güncel seviyesi gösterilir.

AI Zayıf Nokta Analizi: Kullanıcının çözdüğü son quizlerdeki yanlışlarına göre yapay zekâ tarafından hazırlanan kişisel çalışma planı listelenir. Örneğin, ismin hallerinde hata yapan bir kullanıcıya yapay zekâ doğrudan “Akkusativ ve Dativ Gramer Dersini Tekrar Et” önerisinde bulunur ve yönlendirme linki sunar.

Haftalık İlerleme Grafiği: Recharts kullanılarak çizilen grafik, son 7 günde kazanılan XP puanlarının değişimini gösterir.

3.3. Eğitim Modülleri

3.3.1. Dersler ve Konu Anlatımı Arayüzü

Lessons.jsx sayfasında tüm dersler seviyelerine göre filtrelenerek kartlar halinde listelenir. Bir ders seçildiğinde açılan LessonDetail.jsx şu içerikleri barındırır:

Konu Anlatımı (Theory): Konunun detaylı anlatımı ve kuralları.

Örnekler (Examples): Günlük hayattan Almanca cümle örnekleri ve Türkçe karşılıkları.

Ders Sonu Alıştırmaları: Konuyu pekiştirici boşluk doldurma ve çoktan seçmeli sorular.

Kutlama Etkisi: Ders başarıyla tamamlandığında confetti.js yardımcı aracı çağrılarak ekranda konfeti patlatılır ve kullanıcıya XP ödülü verilir.

3.3.2. Kelime Öğrenme ve Flashcard (SRS) Sistemi

Almanca kelimelerin artikelleri ile birlikte öğrenilmesini kolaylaştırmak amacıyla geliştirilen kelime modülü, Aralıklı Tekrar Algoritması (SRS) ile çalışır:

Kart İçeriği: Kelimeler Almanca yazılışı, artikeli, Türkçe anlamı ve örnek cümle bilgisiyle kartlar halinde listelenir.

Flip Animasyonu: Flashcard arayüzünde kullanıcı karta tıkladığında kart 3D animasyonla döner (flip) ve Türkçe anlamı ile örnek cümlesi görüntülenir.

Tekrar Mekanizması: Kullanıcı kelimeyi “Biliyorum” veya “Tekrar Et” olarak işaretler. “Tekrar Et” olarak işaretlenen kelimeler, algoritma tarafından daha sık karşısına çıkarılmak üzere saklanır.

3.3.3. Okuma ve Dinleme Portalı (Reading & Listening)

Kullanıcıların anlama becerilerini artırmak için hikayeler, makaleler ve podcast'ler sunulur:

Okuma Parçaları (Reading): Kullanıcı metni okurken anlamadığı bir kelimeye tıkladığında, yapay zekâ o kelimeyi bağlama göre Türkçe olarak açıklar.

Dinleme Parçaları (Listening): Ses dosyası/podcast çalınırken aynı zamanda konuşma dökümü (transcript) ekrandan takip edilebilir. Tarayıcı ses sentezi (Speech Synthesis) entegrasyonu ile kelimelerin Almanca telaffuzları dinlenebilir. Ses hızı ayarı (örneğin 0.55x yavaş çekim) dil öğrenenler için kolaylık sağlar.

3.3.4. Quiz ve İnteraktif Sınav Modülü

Ders sonlarında veya bağımsız olarak çözülebilen quizler; gramer, kelime bilgisi, okuma ve dinleme türlerindedir.

Sonuç Hesaplama: Quiz tamamlandığında doğru ve yanlış sayıları, kazanılan XP hesaplanır.

AI Soru Açıklaması: Kullanıcı yanlış yaptığı soruların üzerine tıkladığında, yapay zekâ sorunun doğru cevabını ve neden o seçeneğin doğru olduğunu açıklayan Türkçe analiz sunar.

3.4. AI Destekli Pratik ve Yazma Değerlendirici (AI Writing Evaluator & Chat)

Platformun en güçlü modüllerinden biri yapay zekâ tabanlı pratik ekranlarıdır:

AI Chat (Qwen AI Almanca Öğretmeni): Kullanıcı yapay zekâ ile Almanca olarak yazışır. Kullanıcı bir mesaj gönderdiğinde, yapay zekâ hem Almanca cevap verir hem de kullanıcının yazdığı mesajda dil bilgisi hatası varsa (örneğin hatalı fiil çekimi veya yanlış artikkel kullanımı) bunu tespit ederek doğrusunu Türkçe dil bilgisi açıklamasıyla birlikte ekranda gösterir.

AI Yazma Değerlendiricisi (Writing Evaluator): Kullanıcıya belirli bir konuda (örneğin kendini tanıtmaya veya şikayet e-postası yazma) kompozisyon ödevi verilir. Kullanıcı metni yazıp gönderdiğinde yapay zekâ; yazıyı 100 üzerinden puanlar, hataları liste halinde orijinali/doğrusu şeklinde çıkarır ve konuya uygun kusursuz bir Almanca model cevap (model answer) üretir.

4. SONUÇ

Deutschlicht projesi; modern frontend kütüphanelerinin (React 19, Vite 8, Redux Toolkit, Bootstrap 5, Recharts, Framer Motion) bir araya getirilmesiyle ortaya çıkmış, son derece dinamik ve premium tasarıma sahip bir uygulamadır.

Proje kapsamında geliştirilen arayüz, cam morfolojisi teması ve mikro etkileşimleri ile kullanıcıların ilgisini çekecek estetiğe kavuşturulmuştur. Arka planda kurgulanan yapay zekâ entegrasyonları (OpenRouter Qwen API ve yerel simülasyon motoru), bir frontend projesinin sınırlarını aşarak gerçek dünyada katma değer üretebilecek bir eğitim platformu altyapısı sunmaktadır. Oyunlaştırma mekanikleri, aralıklı tekrar sistemi ve konuşma sentezi özellikleri ile dil öğrenimini eğlenceli ve verimli hale getiren Deutschlicht, softlTo Frontend Developer eğitimi boyunca edinilen teorik ve pratik bilgilerin eksiksiz bir yansımasıdır.