**Proje Adı:** World Explorer: Artırılmış Gerçeklik Temelli Eğitici Mobil Uygulama

**Proje Türü:** Mobil Uygulama (React Native + AR)

**Hedef Platform:** Android & iOS

**Geliştirme Ortamı:** React Native + Expo + ViroReact (eject sonrası AR)

**Teknolojiler:**

* React Native
* Expo
* TypeScript
* ViroReact (AR desteği)
* Zustand (state management)
* AsyncStorage / Firebase (veri depolama)
* Lottie (animasyonlar)

**🚀 Projenin Amacı**

Bu projenin temel amacı, çocuklara dünyayı gezdirerek öğretmek ve bu süreci oyunla birleştirmektir. Artırılmış Gerçeklik (AR) teknolojisinin sunduğu sürükleyici deneyim sayesinde çocuklar, hem sanal hem gerçek dünyayı bir arada deneyimlerken öğrenirler. Proje, özellikle 6–12 yaş grubuna hitap etmektedir.

**📖 Proje Kapsamı**

World Explorer uygulaması, oyun tabanlı eğitim modeli ile coğrafya, tarih, dil ve genel kültüre dair bilgiler sunar. Aşağıdaki bölümleri kapsamaktadır:

* Avatar ve karakter yaratma
* AR destekli dünya haritası
* Görev bazlı keşif durakları (her ülke bir durak)
* Mini oyunlar ve bilgi kartları
* Gelişim takibi ve dijital pasaport
* Ebeveyn kontrol paneli

**📄 Hedef Kitle**

* 6-12 yaş arası çocuklar
* Eğitici dijital içerik arayan ebeveynler
* Okul öncesi ve ilköğretim öğretmenleri

**🔎 Benzer Uygulamalardan Farklılıkları**

* Mobil AR teknolojisi ile fiziksel etkileşimi birleştirir.
* Avatar, pasaport ve ödül sistemleri ile çocukları motive eder.
* Güvenli içerik sunumu ve ebeveyn kontrolü sağlar.
* Çoklu ülke, kültür ve dil bilgisi ile çok yönlü öğrenme sağlar.

## 📅 1. Zaman Çizelgesi (Geliştirme Aşamaları)

| **Aşama** | **Süre** | **Tarih Aralığı** |
| --- | --- | --- |
| Proje Planlama ve Araştırma | 1 hafta | 1-7 Temmuz |
| Gereksinim Analizi | 1 hafta | 8-14 Temmuz |
| UI/UX Tasarım | 1 hafta | 15-21 Temmuz |
| Geliştirme - MVP Sürüm | 3 hafta | 22 Temmuz - 11 Ağustos |
| AR Modül Entegrasyonu (Eject + ViroReact) | 2 hafta | 12-25 Ağustos |
| Test Süreci ve Düzeltmeler | 1 hafta | 26 Ağustos - 1 Eylül |
| Yayınlama ve Dokümantasyon | 1 hafta | 2-8 Eylül |

Toplam süre: Yaklaşık **8 hafta**

## 👥 2. Kaynak Planlaması

### 💼 Ekip Rolleri

* **Proje Yöneticisi**: Planlama, süreç takibi
* **Mobil Geliştirici**: React Native + Expo + AR
* **AR Uzmanı**: ViroReact üzerinden 3D modelleme ve AR senaryoları
* **UI/UX Tasarımcı**: Eğitici ve çocuk dostu arayüz
* **Test Uzmanı**: Fonksiyonel ve AR testleri
* **İçerik Yöneticisi**: ülke, bilgi kartları, mini oyun verileri

### 🛠 Gerekli Yazılım ve Donanım

* React Native çalışan IDE (VS Code, Android Studio)
* AR geliştirme test cihazları (Android/iOS, ARCore/ARKit destekli)
* 3D modelleme dosyaları (obj, gltf)
* Firebase ya da benzeri backend (isteğe bağlı)

## ⚠️ 3. Risk Analizi

| **Risk Faktörü** | **Çözüm Önerisi** |
| --- | --- |
| ARKit/ARCore cihaz uyumsuzluğu | WebAR yedekleme, cihaz testi listesi |
| Expo'dan Eject sonrası hata riski | Eject öncesi kod temizlik ve ayrı test projesi |
| Çocuklar için karmaşık arayüz tasarımı riski | UX testleri, sade ekran yapıları |
| AR içerik performans problemleri | Model optimizasyonu, FPS testi |
| Ebeveyn kontrolü gibi hassas içeriklerde hata | Manuel test, JSON veri kontrol sistemleri |

## ⭐ 4. Başarı Kriterleri

* Uygulama minimum AR senaryolarla yayınlanmış olacak
* Kullanıcı 3 ülkeyi tamamlayıp bilgi kartlarına ulaşabilecek
* Avatar sistemi, mini oyunlar ve pasaport sistemi aktif olacak
* AR görevlerindeki objeler %90 doğrulukla görüntülenecek
* Ebeveyn kontrol menüsü çalışır durumda olacak

## 🔢 1. Fonksiyonel Gereksinimler

### 1.1 Kullanıcı Girişi ve Avatar Oluşturma

* Kullanıcı oyuna giriş yaptığında "Kaşif Akademisi" ekranı görünmelidir.
* Kullanıcı, saç stili, göz rengi, kıyafet ve aksesuar gibi seçeneklerle avatar oluşturabilmelidir.
* Seçilen avatar ve tercihler yerel bellekte (AsyncStorage) saklanmalıdır.

### 1.2 Uçuş Rotaları ve Duraklar

* Kullanıcılar, harita üzerinde duraklar seçebilmeli (her durak bir ülke veya şehir).
* Rotalar üzerinden görevler tetiklenmelidir.
* Her durakta AR görevler, mini oyunlar ve bilgi kartları olmalıdır.

### 1.3 AR Görevleri ve Etkileşim

* Kamera üzerinde 3D yapılar (Eyfel Kulesi gibi) görüntülenmelidir.
* Kullanıcı, bu yapılara dokunarak bilgi kartlarını açabilmelidir.
* Nesne tanıma veya yüzey algılama ile yerleştirme yapılmalıdır (ViroReact).

### 1.4 Mini Oyunlar ve Öğrenme Aktiviteleri

* "Harita Dedektifi", "Bulut Boyama", "Hava Durumu Tahmincisi" gibi mini oyunlar oynanabilmelidir.
* Her mini oyun, belirli bir kazanımla ilişkili olmalı (rozet, puan vs.).
* Oyun içi ilerleme durumu kaydedilmelidir.

### 1.5 Gelişim Takibi ve Ödüllendirme

* Her ülke geçildiğinde pasaporta dijital damga eklenmelidir.
* Kullanıcının ilerlemesi "Süper Kaşif Seviyesi" ile takip edilmelidir.
* Rozetler ve avatar öğeleri kazanılmalıdır.

### 1.6 Ebeveyn Kontrol Paneli

* Ebeveynler, çocuğun ilerlemesini görebilmelidir.
* Ebeveynler oyun sürelerini sınırlayabilmelidir.
* Filtreleme ve içerik izleme opsiyonları bulunmalıdır.

## 🔨 2. Fonksiyonel Olmayan Gereksinimler

### 2.1 Performans

* AR içerikler düşük gecikmeyle çalışmalıdır (FPS ≥30).
* Uygulama maksimum 60 FPS hedeflemelidir.

### 2.2 Platform Desteği

* Android 10+ ve iOS 13+ desteklenmelidir.
* ARKit/ARCore destekli cihazlar tespit edilip AR açılmalıdır.

### 2.3 Erişilebilirlik

* Çocuklar için sade arayüz ve renkli tema uygulanmalı.
* Sesli anlatımlar ve ikon temelli navigasyon olmalıdır.

### 2.4 Güvenlik

* Veriler sadece yerel olarak ya da Firebase üzerinden şifreli tutulmalıdır.
* Çocuk gizliliği için COPPA uyumlu içerik geliştirilmelidir.

### 2.5 Bakım ve Güncellenebilirlik

* Yeni ülke, görev ve oyun içerikleri JSON veya API ile dinamik olarak eklenebilir olmalıdır.

## 🛠️ 1. Genel Mimari Yapı

Proje, katmanlı ve modüler yapıya sahip olacak şekilde tasarlanmıştır:

### Katmanlar:

* **Sunum Katmanı (UI)**: Kullanıcı arayüzleri ve ekranlar
* **Uygulama Mantığı (Logic)**: Bileşen içi etkileşimler, mini oyunlar, veri filtreleme
* **Durum Yönetimi (State)**: Zustand üzerinden kullanıcı durumu, avatar bilgisi, ilerleme
* **Veri Katmanı (Data)**: AsyncStorage veya Firebase üzerinden veri saklama ve alma
* **Servis Katmanı (Services)**: İleride eklenecek API veya dinamik içerik yapıları için hazır

## 🔄 2. Navigasyon ve Akış Yapısı

Expo Router kullanılacak ve aşağıdaki sekmeler bulunacak:

* **Akademi**: Avatar yaratma, sertifika ekranları
* **Uçuş**: Harita, rota seçimi, 3D dünya görüntüleme
* **Görevler**: Seçilen ülke için AR görev ekranları
* **Mini Oyunlar**: Matematik, kelime, hava durumu oyunları
* **Profil**: Pasaport, kazanılan rozetler, ebeveyn paneli

Her sekme için stack yapısı içinde alt ekranlar bulunur (detay, bilgi kartı, oyun içi gibi).

## 🪡 3. Teknik Tasarım Detayları

### 3.1 AR Modülü

* **ViroReact** kullanılacak (Expo'dan eject edildikten sonra)
* Kamera yüzey tespiti (plane detection) üzerine 3D model yerleşirme
* ViroARScene, Viro3DObject, ViroText gibi bileşenlerle AR sahneler
* 3D modeller .obj veya .gltf olarak /assets/models içinde barındırılacak

### 3.2 Durum Yönetimi

* Zustand store yapısı:

interface UserStore {

avatar: AvatarData;

passport: PassportEntry[];

progress: Record<string, boolean>;

setAvatar: (data: AvatarData) => void;

completeTask: (taskId: string) => void;}

### 3.3 Veri Yapıları

* Pasaport: countryId, dateVisited, badge
* Avatar: hair, eyes, outfit, equipment
* Ülke Bilgisi: id, name, landmark3D, languagePack, miniGames[]

### 3.4 AsyncStorage

* Kullanıcı avatarı, oyun ilerlemesi ve seçilen ülkeler AsyncStorage üzerinden tutulur.
* Opsiyonel olarak Firebase senkronizasyonu planlanabilir.

## 📚 4. Bileşen Yapıları

### 4.1 /components

* AvatarEditor.tsx
* FlightMap.tsx
* ARLandmarkViewer.tsx
* GameCard.tsx
* ParentalPanel.tsx

### 4.2 /app

* (tabs)/academy/index.tsx
* (tabs)/flight/index.tsx
* (tabs)/missions/[country].tsx
* (tabs)/games/index.tsx
* (tabs)/profile/index.tsx

## 🎓 1. Geliştirme Ortamı Kurulumu

### 1.1 Expo Projesi Oluşturma

npx create-expo-app world-explorer --template tabs (veya blank)

cd world-explorer

npm install

### 1.2 Kullanılan Önemli Kütüphaneler

npm install zustand @react-navigation/native @react-navigation/stack

npm install lottie-react-native

npm install expo-router

### 1.3 AR Özellikleri için Expo'dan Çıkılması (Eject)

npx expo prebuild

npx expo eject

npm install --save react-viro

## 📂 2. Dosya Yapısı Özeti

/app

(tabs)/academy

(tabs)/flight

(tabs)/missions

(tabs)/games

(tabs)/profile

/components

/constants

/hooks

/services

/types

/utils

/assets (resimler, sesler, 3D modeller)

## 🔄 3. Temel Özelliklerin Kodlanması

### 3.1 Avatar Oluşturma

// components/AvatarEditor.tsx

export default function AvatarEditor() {

const [avatar, setAvatar] = useState({ hair: 'short', outfit: 'explorer' });

return (

<View>

<Text>Avatar Seçimi</Text>

<Button title="Saç Değiştir" onPress={() => setAvatar({ ...avatar, hair: 'long' })} />

</View>

);

}

### 3.2 Uçuş Haritası

// components/FlightMap.tsx

import MapView from 'react-native-maps';

export default function FlightMap() {

return (

<MapView style={{ flex: 1 }} initialRegion={{

latitude: 48.85,

longitude: 2.35,

latitudeDelta: 50,

longitudeDelta: 50,

}} />

);

}

### 3.3 AR Nesne Gösterimi

// components/ARLandmarkViewer.tsx

import { ViroARScene, Viro3DObject } from 'react-viro';

export default function ARLandmarkViewer() {

return (

<ViroARScene>

<Viro3DObject

source={require('../assets/models/eiffel.obj')}

position={[0, 0, -2]}

scale={[0.1, 0.1, 0.1]}

type="OBJ"

/>

</ViroARScene>

);

}

### 3.4 Oyun Durumu Yönetimi (Zustand)

// store/useUserStore.ts

import create from 'zustand';

export const useUserStore = create(set => ({

avatar: null,

setAvatar: avatar => set({ avatar }),

passport: [],

addStamp: country => set(state => ({ passport: [...state.passport, country] }))

}));

## 📚 4. Geliştirme Aşamalarında Notlar

* Her bölüm bileşen bazlı geliştirildi.
* Expo Router ile ekranlar dosya bazlı organize edildi.
* AR içerikler minimum test modellerle başlandı (OBJ, GLB).
* AsyncStorage ile ilerleme ve avatar bilgisi yerel olarak tutuldu.
* Firebase ileride eklenecek opsiyonel modül olarak tanımlandı.

## 🔧 1. Test Stratejisi

World Explorer projesinde hem teknik hem de kullanıcı odaklı testler uygulanacaktır. AR teknolojisi kullanıldığı için cihaz testi ayrı bir önem taşır.

### 1.1 Test Türleri:

* **Birim Testleri (Unit Tests)**: Mini oyunlar, veri dönüşümleri, store mantığı
* **Bütünleşik (Integration) Testleri**: Avatar seçimi + kaydetme + görüntüleme
* **Fonksiyonel Testler**: Görev tamamlama, rozet alma, pasaport sistemi
* **AR Testleri**: Nesne yerleşme, boyut, etkileşim
* **Kullanıcı Kabul Testleri**: Gerçek kullanıcılar (5-10 yaş aralığında) ile test

## 🔹 2. Test Senaryoları

### 2.1 Avatar Oluşturma

* Kullanıcı saç, göz, aksesuar seçebiliyor mu?
* Avatar bilgisi kaydedilip tekrar açıldığında yükleniyor mu?

### 2.2 Harita ve Uçuş Ekranı

* Harita görüntüleniyor mu?
* Durak seçimi yapılabiliyor mu?
* Göreve geçiş yapılıyor mu?

### 2.3 AR Nesne Yerleştirme

* 3D model gerçek zemin üzerine doğru yerleşiyor mu?
* Objeye tıklandığında bilgi kartı açılıyor mu?
* AR görev FPS'leri 30'un altına düşüyor mu?

### 2.4 Mini Oyunlar

* Gökyüzü Matematik oyunu doğru hesaplama yapıyor mu?
* Bulut boyama oyununda kayıt alınıyor mu?

### 2.5 Ebeveyn Paneli

* Ebeveyn için ilerleme raporları görüntüleniyor mu?
* Günlük süre limiti uygulanıyor mu?

## 📊 3. AR Özel Test Yaklaşımları

* **Cihaz Uyumluluğu**: Android ve iOS cihazlarda ARCore/ARKit desteği test edilir.
* **Zemin Algılama**: Model yerleşimi farklı zemin tiplerinde test edilir (halı, masa, dış mekan).
* **İşık Koşulları**: Loş ortamlarda model algısı test edilir.
* **Performans**: AR sahne açıldığında uygulama donuyor mu, frame drop yaşanıyor mu?

## 🛋️ 4. Kullanıcı Test Planı

* **Amaç**: Çocukların oyunu ne kadar anladığını, görevleri başarıp başaramadığını anlamak
* **Yaş grubu**: 6-12 yaş
* **Test süreci**: Gözetimli 15-20 dakikalık oturumlar
* **Kriterler**:
  + Oyunda kaybolmadan ilerleyebiliyor mu?
  + AR nesneleriyle etkileşim kurabiliyor mu?
  + Bilgileri anlayıp uygulayabiliyor mu?

## 📆 1. Sürekli Gelişim Stratejisi

World Explorer uygulaması, yayınlandıktan sonra da içerik ve teknik altyapı olarak güncellenmeye devam edecektir. Bu sürekli gelişim aşağıdaki alanlara odaklanacaktır:

### 1.1 Yeni Ülke ve Görev Paketleri

* Ayda en az 2 yeni ülke AR paketi eklenmesi hedeflenmektedir.
* Her paket, 1 AR nesnesi, 2 mini oyun ve 3 bilgi kartı içermelidir.
* Veriler JSON dosyalarıyla dinamik olarak uygulamaya entegre edilecektir.

### 1.2 Avatar ve Eşya Genişlemesi

* Yeni saç modelleri, göz renkleri, giyimler ve ekipmanlar
* Sezonluk ödüllerle gelen "sürpriz aksesuarlar"

### 1.3 Mini Oyun Kütüp hanesi

* Yeni oyun tipleri: hafıza oyunu, harita bulmaca, sesli eşleştirme
* Skor tablosu ve liderlik sisteminin eklenmesi

## 🚧 2. Hata Takibi ve Düzeltme

### 2.1 Hata Bildirim Sistemi

* Kullanıcı hatalarını raporlayabileceği bir form veya Firebase Crashlytics entegrasyonu
* AR hataları için cihaz tipi ve işletim sistemi bilgisi toplanacak

### 2.2 İzleme ve Loglama

* Konsol logları, mobil performans izleyicileri (Sentry, Bugsnag)
* İlgili güncelleme için changelog girişi zorunluluğu

## 🚀 3. Gelecek Gelişme Planları

### 3.1 Çok Oyunculu Mod

* Gerçek zamanlı mini görevler
* Çocukların avatarları ile ortak keşif
* Ebeveyn izniyle arkadaş ekleme sistemi

### 3.2 Gelişmiş Ebeveyn Kontrol Paneli

* Günlük ilerleme grafikleri
* Kazanılan bilgiler için raporlama
* Oyun sürelerindeki trend takibi

### 3.3 AR İçin Yapay Zeka Desteği

* Kamera üzerinden şekil veya nesne tanıma
* Gerçek zamanlı nesne algısına göre bilgi kartı çıkarma

## 🔄 4. Bakım Takvimi

| **Periyot** | **Faaliyet** |
| --- | --- |
| Haftalık | Hata raporu gözden geçirme, kritik düzeltme |
| Aylık | Yeni ülke/görev ekleme, mini oyun testi |
| 3 Aylık | Versiyon artışı, yeni büyük özellik yayını |