**roje: Online Bilgi Kapışması Oyunu ve Mobil Uygulama**

**Vizyon:** Kullanıcıların kayıt olabileceği, online ödeme yapabileceği, pul kazanıp/harcayabileceği, diğer oyuncularla düello yapabileceği ve mobil cihazlarda oynanabilen bir Bilgi Kapışması platformu oluşturmak.

**AŞAMA 1: Temel Backend (Sunucu Tarafı) ve Veritabanı Kurulumu (Kullanıcı Yönetimi Odaklı)**

Bu aşama, projenin kalbi olacak. Oyunun tüm online özelliklerinin arkasındaki altyapı burada kurulacak.

**Tahmini Süre:** 3-6 Hafta (Deneyime ve öğrenme hızına bağlı)

**Gereken Teknolojiler/Uzmanlıklar:**

* **Backend Programlama Dili:** Node.js (Express.js ile), Python (Django/Flask), PHP (Laravel), Ruby (Rails) gibi popüler bir seçenek. **Node.js** ve **Express.js** JavaScript tabanlı olduğu için mevcut HTML/CSS/JS bilginle uyumlu olabilir.
* **Veritabanı:** PostgreSQL, MySQL (ilişkisel) veya MongoDB (NoSQL) gibi. **MongoDB** hızlı başlangıç için iyi bir seçenek olabilir.
* **API Geliştirme:** RESTful API veya GraphQL.
* **Kimlik Doğrulama (Authentication):** JWT (JSON Web Tokens) veya OAuth.
* **Sunucu:** AWS, Google Cloud, Azure veya DigitalOcean gibi bulut sağlayıcıları.

**Adımlar:**

1. **Backend Dili ve Framework Seçimi:** Yukarıdaki seçeneklerden birine karar ver. Node.js başlangıç için iyi bir tercih olabilir.
2. **Veritabanı Seçimi ve Modellemeyi Planlama:**
   * Kullanıcılar tablosu (ID, kullanıcı adı, e-posta, şifre (hashed), pul miktarı, doğru/yanlış cevap sayısı vb.)
   * Sorular tablosu (ID, soru metni, seçenekler, doğru cevap, maliyet, kategori vb.) - Mevcut soruları buraya taşıyacaksın.
   * Oyunlar tablosu (ID, Oyuncu1 ID, Oyuncu2 ID, kazanan ID, başlangıç zamanı, bitiş zamanı, pul değişimi vb.)
   * Ödemeler tablosu (işlem ID, kullanıcı ID, miktar, tarih, durum vb.)
3. **Temel Backend Projesini Başlatma:** Seçtiğin dil ve framework ile boş bir proje oluştur.
4. **Veritabanı Bağlantısını Kurma:** Backend uygulamanı seçtiğin veritabanına bağla.
5. **Kullanıcı Kayıt (Register) API Uç Noktası:**
   * Kullanıcı adı, e-posta ve şifre alacak.
   * Şifreyi güvenli bir şekilde hash'leyip veritabanına kaydedecek.
6. **Kullanıcı Giriş (Login) API Uç Noktası:**
   * Kullanıcı adı/e-posta ve şifre alacak.
   * Şifreyi doğrulayacak ve başarılı ise bir JWT token döndürecek.
7. **Kullanıcı Profili API Uç Noktası (Yetkilendirme ile):**
   * Gelen JWT token'ı doğrulayarak kullanıcının pul miktarını, doğru/yanlış sayısını ve diğer profil bilgilerini döndürecek.
8. **Mevcut Frontend'i Backend ile Entegre Etme:** Pul miktarını ve soruları artık backend'den çekmek için JavaScript kodunu güncelleyeceksin.

**AŞAMA 2: Online Ödeme ve Pul Yönetimi Sistemi**

Bu aşamada, kullanıcıların pul satın almasını ve pul geri ödeme işlemlerini yöneteceğiz.

**Tahmini Süre:** 2-4 Hafta

**Gereken Teknolojiler/Uzmanlıklar:**

* **Ödeme Ağ Geçidi Entegrasyonu:** Stripe, Iyzico, PayTR gibi bir ödeme sağlayıcısı API'si. Türkiye için Iyzico veya PayTR daha uygun olabilir.
* **Güvenlik:** SSL/TLS, PCI DSS uyumluluğu (genellikle ödeme sağlayıcısı halleder).

**Adımlar:**

1. **Ödeme Sağlayıcısı Araştırması ve Seçimi:** İş modeline ve bütçene uygun bir ödeme ağ geçidi seç. Geliştirici dokümantasyonlarını incele.
2. **Ödeme Sağlayıcısı API Entegrasyonu:**
   * Pul satın alma için ödeme isteği gönderme.
   * Ödeme onaylarını (webhooks) işleme ve kullanıcının pul miktarını veritabanında güncelleme.
3. **Pul Geri Ödeme (Cash-Out) Sistemi:**
   * Kullanıcının pul satma talebi için API uç noktası.
   * Kullanıcının pulunu düşürme ve bir geri ödeme işlemi kaydı oluşturma.
   * (Gerçek dünyada, bu manuel bir onay süreci veya otomatik bir banka transferi entegrasyonu gerektirebilir ki bu oldukça karmaşıktır.)
4. **Frontend Ödeme Arayüzü:** Mevcut HTML ve JavaScript'i, ödeme sağlayıcının entegrasyonuna uygun hale getir.

**AŞAMA 3: Online Düello ve Gerçek Zamanlı Oyun Mekaniği**

Bu, projenin en heyecan verici ve teknik olarak zorlu kısımlarından biri.

**Tahmini Süre:** 4-8 Hafta

**Gereken Teknolojiler/Uzmanlıklar:**

* **WebSockets:** Gerçek zamanlı iletişim için (Socket.IO gibi bir kütüphane).
* **Oyun Durumu Yönetimi:** Sunucu tarafında oyunun ve oyuncuların durumunu yönetmek için mantık.
* **Eşleştirme Algoritması:** Oyuncuları eşleştirmek için basit veya karmaşık bir sistem.

**Adımlar:**

1. **WebSocket Sunucusunu Kurma:** Backend uygulamanın üzerine bir WebSocket katmanı ekle (örneğin Node.js için Socket.IO).
2. **Oyun Odası ve Eşleştirme Sistemi:**
   * Oyuncuların bir düello araması başlatması için API.
   * Uygun bir oyuncu bulunduğunda (veya AI bot ile eşleştiğinde) bir oyun odası oluşturma.
   * Oyuncuları bu odaya sokma.
3. **Gerçek Zamanlı Soru Akışı:**
   * Soru yükleme, cevap gönderme, süre bitimi gibi olayları WebSocket üzerinden iki oyuncu arasında senkronize etme.
   * Sunucu, hangi oyuncunun sırası olduğunu, süreyi ve cevabın doğruluğunu yönetecek.
4. **Pul ve İstatistik Güncellemesi:** Oyun sonunda kazananın pulunu artırma, kaybedenin pulunu düşürme ve doğru/yanlış cevap sayılarını güncelleme.
5. **Frontend Düello Arayüzü:** Düello başlatma, bekleme, oyun içi arayüz ve sonuç ekranını güncelleyen JavaScript.

**AŞAMA 4: Mobil Uygulama Geliştirme**

Web uygulaman tamamlandığında, bunu mobil platformlara taşıyabiliriz.

**Tahmini Süre:** 6-12 Hafta (Platform sayısına ve deneyime bağlı)

**Gereken Teknolojiler/Uzmanlıklar:**

* **Cross-Platform (Önerilen):** React Native, Flutter (tek kod tabanıyla hem iOS hem Android).
* **Native (Alternatif):** Swift/Kotlin (iOS/Android için ayrı ayrı). Bu daha fazla zaman ve uzmanlık gerektirir.
* **API Tüketimi:** Mobil uygulamanın geliştirdiğin backend API'leri ile iletişim kurması.

**Adımlar:**

1. **Mobil Framework Seçimi:** React Native veya Flutter ile başlama kararı al. Bu iki framework, mevcut JavaScript bilginle React Native'i öğrenmeyi daha kolay kılabilir.
2. **Geliştirme Ortamı Kurulumu:** Seçtiğin framework için gerekli SDK'ları ve araçları kur.
3. **Mobil Arayüz Tasarımı:** Web arayüzüne benzer ama mobil deneyime uygun bir UI/UX taslağı oluştur.
4. **API Entegrasyonu:** Mobil uygulamanın kullanıcı kaydı, giriş, pul işlemleri, soru çekme ve düello gibi tüm işlemler için backend API'lerini kullanmasını sağla.
5. **Offline Destek (İsteğe Bağlı):** İnternet bağlantısı olmasa bile belirli özelliklerin çalışmasını sağlamak.
6. **Test ve Hata Ayıklama:** Cihazlarda ve emülatörlerde kapsamlı testler yap.

**AŞAMA 5: Dağıtım (Deployment) ve Pazarlama**

Uygulamanın dünyaya açıldığı son aşama.

**Tahmini Süre:** 2-4 Hafta

**Gereken Teknolojiler/Uzmanlıklar:**

* **App Store Connect (Apple) ve Google Play Console (Android) Hesapları:** Geliştirici hesapları.
* **Pazarlama Materyalleri:** Uygulama ekran görüntüleri, tanıtım metinleri, uygulama simgesi.
* **ASO (App Store Optimization):** Uygulamanın keşfedilmesini sağlama.

**Adımlar:**

1. **Backend Sunucusunu Canlıya Alma (Deployment):** Backend uygulamanı bir bulut sağlayıcısında (AWS EC2/Lambda, Google Cloud Run, Heroku vb.) canlıya al.
2. **Domain ve SSL Sertifikası:** Uygulaman için bir alan adı al ve HTTPS için SSL sertifikası kur.
3. **Uygulama Mağazası Hesaplarını Oluşturma:** Apple Developer Program ve Google Play Geliştirici hesabı için kaydol (yıllık ücretleri var).
4. **Uygulama Mağazalarına Yükleme:**
   * Uygulamanın son sürümünü derle.
   * Mağaza girişlerini hazırla (açıklama, ekran görüntüleri, kategori vb.).
   * Uygulamayı incelemeye gönder.
5. **Pazarlama ve Tanıtım:** Sosyal medya, web sitesi veya diğer kanallar aracılığıyla uygulamanı duyur.

**Genel Notlar ve Önemli Uyarılar:**

* **Öğrenme Eğrisi:** Bu projenin her aşaması yeni teknolojiler öğrenmeni gerektirecek. Bu yüzden sabırlı ve araştırmacı olmalısın.
* **Güvenlik:** Özellikle ödeme ve kullanıcı bilgileri konusunda güvenlik her zaman en yüksek önceliğin olmalı. Şifreleri hash'leme, SSL kullanma, API anahtarlarını güvenli tutma gibi konulara dikkat et.
* **Maliyet:** Sunucu, veritabanı, domain, SSL sertifikası ve geliştirici hesapları için belli bir maliyetin olacağını unutma.
* **Yardım Almak:** Her adımı tek başına yapmak zorunda değilsin. Belirli konularda (örneğin ödeme entegrasyonu veya mobil uygulama geliştirme) uzmanlardan yardım almayı düşünebilirsin.
* **Minimum Viable Product (MVP):** Tüm özellikleri bir anda yapmaya çalışma. Önce kullanıcı kaydı ve basit bir online soru-cevap mekaniği ile başlayıp (MVP), başarılı oldukça diğer özellikleri eklemek daha gerçekçi bir yaklaşımdır.

Bu yol haritası oldukça kapsamlı. Hangi adımla başlamak istediğini veya daha fazla bilgi almak istediğin bir alan olursa, bana söyleyebilirsin. Başlangıç olarak genellikle **Aşama 1: Temel Backend ve Veritabanı Kurulumu** ile başlanır.