**Movie Recommendation App – Geliştirme ve Kullanım Dokümanı**

**1. Giriş**

Movie Recommendation App, kullanıcılara film arama, film detaylarını görüntüleme ve yapay zekâ destekli film önerileri sunmayı amaçlayan bir Android uygulamasıdır.  
Proje, film tutkunlarının yeni filmler keşfetmesini kolaylaştırmak, kullanıcı deneyimini kişiselleştirmek ve modern mobil teknolojilerin kullanımını göstermek amacıyla geliştirilmiştir.

Hedef kitle: Film izleyicileri, yeni filmler keşfetmek isteyen kullanıcılar ve kişisel film arşivine yenilik katmak isteyen bireyler.

**2. Geliştirme Süreci**

1. **Proje oluşturma ve yapılandırma**
   * Android Studio üzerinde Kotlin dilinde proje başlatıldı.
   * MVVM mimarisi temel alınarak models, repository, viewmodels, ui, database, network paketleri oluşturuldu.
2. **Kullanıcı Yönetimi (Login & Register)**
   * Kullanıcı bilgileri Room Database kullanılarak saklandı.
   * Kayıt sırasında kullanıcı adı ve şifre SHA-256 ile hashlenerek veritabanına kaydedildi.
   * Giriş ekranında kullanıcı doğrulaması yapıldı.
3. **API Entegrasyonları**
   * **TMDB API**: Film arama ve detay bilgilerini sağladı.
   * **OpenRouter API**: Yapay zekâ üzerinden benzer film önerileri üretildi.
4. **UI Geliştirme**
   * Tüm ekranlar **Jetpack Compose** ile Material3 bileşenleri kullanılarak tasarlandı.
   * Kullanıcı deneyimini iyileştirmek için sade, Netflix tarzı tema uygulandı.
5. **Öneri Mekanizması**
   * Kullanıcı bir film arattığında, OpenRouter’dan yapay zekâya “Benzer 5 film öner” isteği gönderildi.
   * Gelen yanıt TMDB üzerinden doğrulandı ve kullanıcıya liste halinde sunuldu.
6. **Test ve Hata Ayıklama**
   * API bağlantılarında hata yönetimi (try-catch, error handling) eklendi.
   * Yanlış giriş/kayıt durumlarında kullanıcıya hata mesajı gösterildi.
   * UI testleri emulator üzerinde yapıldı.

**3. Kullanılan Teknolojiler ve Kütüphaneler**

* **Kotlin** – Ana programlama dili.
* **Jetpack Compose** – Modern UI geliştirme framework’ü.
* **Material3** – Arayüz tasarım bileşenleri.
* **Room Database** – Kullanıcı kayıt/giriş işlemleri.
* **Retrofit + OkHttp** – API çağrıları.
* **Gson** – JSON verilerini parse etmek için.
* **Coil** – Film posterlerini yüklemek için.
* **Coroutines & Flow** – Asenkron veri işlemleri ve state yönetimi.
* **SHA-256** – Şifre güvenliği için hashleme.

Her kütüphane projede doğrudan belirli bir işlevi yerine getirmiştir:

* Retrofit → API iletişimi
* Room → Kalıcı kullanıcı verisi
* Coil → Poster görselleri
* Flow → UI güncellemeleri

**4. Mimarî Yapı**

Proje **MVVM (Model-View-ViewModel)** mimarisi ile tasarlandı.

* **Model:** Film, User
* **Database:** AppDatabase, UserDao
* **Network:** TmdbApiService, OpenRouterApiService
* **Repository:** FilmRepository, AIRecommendationRepository
* **ViewModel:** SearchViewModel, DetailViewModel, RecommendationsViewModel, LoginViewModel, RegisterViewModel
* **UI (View):** Compose ekranları (LoginScreen, RegisterScreen, SearchScreen, DetailScreen, RecommendationsScreen)

**Veri Akışı:**  
Kullanıcı → UI → ViewModel → Repository → API/DB → ViewModel → UI

**5. Uygulama Özellikleri**

* **Kayıt Olma** → Yeni kullanıcı oluşturulur.
* **Giriş Yapma** → Mevcut kullanıcı şifre doğrulaması ile giriş yapar.
* **Film Arama** → TMDB API üzerinden film araması yapılır.
* **Yapay Zekâ Önerileri** → OpenRouter API kullanılarak benzer filmler önerilir.
* **Film Detayları** → Seçilen film hakkında poster, yıl, puan ve özet bilgileri gösterilir.

**6. Kullanım Senaryosu**

1. Kullanıcı uygulamayı açar.
2. Kayıt olur veya giriş yapar.
3. Film adı girerek arama yapar.
4. Yapay zekâ önerileri alarak benzer filmleri keşfeder.
5. Sonuçlardan bir filmi seçerek detaylarını görüntüler.

**9. Sonuç**

Proje, temel film arama ve öneri işlevlerini yerine getirmektedir. Kullanıcı dostu arayüz ve yapay zekâ destekli öneri sistemi sayesinde kullanıcıların film keşif süreci kolaylaştırılmıştır.