### ****Akademik AI Asistanı Proje Planı****

#### ****1. Proje Planlaması (Gereksinimler & Hedefler)****

* **Proje Kapsamını Belirleme**: Uygulamanın fonksiyonel ve fonksiyonel olmayan gereksinimlerini belirleyin. Parafraz (yeniden ifade etme), dilbilgisi kontrolü, alıntı formatlama, kaynak önerisi ve yerel AI model kullanım gibi istenen özellikleri belirleyin.
* **Aşamalar ve Teslim Tarihleri Belirleme**: Proje genel zaman çizelgesini oluşturun, her aşamayı yönetilebilir görevlere bölün.
* **Temel Teknolojileri Belirleme**: LLaMA/Mistral AI modelleri, FastAPI Backend, React Frontend ve SQLite yerel veri depolama gibi kullanılacak ana teknolojileri seçin.

#### ****2. Model Seçimi & Yerel Kurulum****

* **AI Modellerini Değerlendirme**: Parafraz, özetleme, dilbilgisi kontrolü ve kaynak önerisi görevleri için LLM modellerini inceleyin. Bu modellerin LangChain çerçevesiyle uyumluluğunu kontrol edin.
* **Yerel Ortam Kurulumu**: Yerel AI modellerini (LLama olabilir) GGUF formatında kurarak internet bağımsız çalışacak şekilde yapılandırın.
* **Model Performansını Doğrulama**: Başlangıçta testler yaparak modellerin kaliteli sonuçlar sunduğundan emin olun ve gerekirse optimizasyon yapın.

#### ****3. Temel AI Modülleri Geliştirme****

* **Parafraz Ajanı**:
  + LangChain kullanarak parafraz işlevini geliştirin, doğru ve çeşitli çıktılar elde etmek için promptları ayarlayın.
  + Parafraz doğruluğunu ve tutarlılığını test edin.
* **Özetleme Ajanı**:
  + LangChain zincirlerini kullanarak içeriği özetleyen bir modül oluşturun.
  + Özetlerin önemli bilgileri ve okunabilirliği koruyacak şekilde düzenlendiğinden emin olun.
* **Dilbilgisi & Yazım Kontrolü**:
  + Dilbilgisi düzeltmeleri, cümle yapılandırmaları ve yazım hatası tespiti için spaCy ve TextBlob kullanın.
  + Cümle yapıları ve dilbilgisel seçimler hakkında geliştirmeler öneren bir geri bildirim döngüsü oluşturun.
* **Alıntı Formatlama**:
  + Farklı alıntı stillerini (APA, MLA, Chicago vb.) yönetmek için Pybtex veya CSL JSON ayrıştırıcısını kullanarak bir alıntı formatlama modülü oluşturun.
  + Kaynak verilerinden seçilen alıntı formatına doğru bir şekilde haritalama yapılmasını sağlayın.
* **Kaynak Ajanı**:
  + Kullanıcı sorguları veya metin girişlerine dayalı olarak ilgili akademik kaynakları öneren bir AI kaynak öneri ajansı geliştirin.
  + Kaynak ajansını, kaynakların doğru kategorilere yerleştirilmiş ve alakalı olduğundan emin olacak şekilde entegre edin.
  + Kaynak önerilerinin doğru ve geçerli olduğundan emin olun, yerel veri kümesinden öneriler sunun.

#### ****4. Veri ve Kaynak Önerisi****

* **Veri Kümesi Tasarımı**:
  + Veri kümesinin yapısını ve gerekli alanları belirleyin. Bu alanlar, kaynakların meta verileri, etiketleme ve kategoriler içerecek şekilde tasarlanacaktır.
  + Kaynakları akademik disiplinlere göre etiketleme mekanizması oluşturun.
* **Veri Kümesi Toplama**:
  + Açık kaynaklı akademik veritabanlarını toplayın, AYBU'dan (Ankara Yıldırım Beyazıt Üniversitesi) ve diğer serbest erişilebilir kaynaklardan veri kazıyın.
  + Akademik kaynaklardan kapsamlı ve çeşitli bir set oluşturun.
* **Veri Kümesi Entegrasyonu & Yerel İndeksleme**:
  + Toplanan verileri bir SQLite veritabanına organize edin.
  + Kaynaklara hızlı erişim sağlamak için indeksleme ve arama özelliklerini uygulayın.

#### ****5. Backend Entegrasyonu****

* **API Tasarımı**:
  + Parafraz, özetleme, alıntı formatlama ve kaynak önerisi gibi görevleri işlemek için FastAPI arka uç API'sini oluşturun.
  + Her görev için RESTful API uç noktalarını tanımlayın ve doğru belgeler sağlayın.
* **Ajan - API Bağlantısı**:
  + LangChain ajanlarını FastAPI Backendi ile entegre edin ve veri akışının sorunsuz olmasını sağlayın.
  + Gerekli özellikleri (kimlik doğrulama ve veri doğrulama gibi) entegre edin.
* **Yardımcı Araç Entegrasyonu**:
  + Metin temizleme, formatlama gibi yardımcı işlevleri arka uca entegre edin.
  + Metinlerin temiz, iyi yapılandırılmış ve tutarlı olmasını sağlayın.

#### ****6. Frontend Geliştirme****

* **UI Prototip Tasarımı**:
  + Web veya masaüstü uygulaması için basit prototipler oluşturun. Ana özellikler: metin giriş alanları, AI sonuçlarının gösterimi ve formatlama seçenekleri.
  + Kullanıcı dostu ve sezgisel bir arayüz tasarımı yapın.
* **Frontend Geliştirme**:
  + React kullanarak uygulamanın ana sayfalarını geliştirin.
  + Parafraz, özetleme, dilbilgisi kontrolü, alıntı oluşturma, kaynak önerileri modülleri için işlevsellik ekleyin.
* **Backend API Bağlantısı**:
  + Axios veya fetch kullanarak ön uç ile FastAPI arka ucu arasında bağlantı kurun.
  + Ön uç ve arka uç arasındaki asenkron veri alışverişini sağlayın.

#### ****7. Test ve İyileştirme****

* **Fonksiyonel Testler**:
  + Parafraz, özetleme, dilbilgisi kontrolü, kaynak önerisi gibi tüm temel AI işlevlerini test edin.
  + Backend ve Frontend entegrasyonunu test ederek sistemin sorunsuz çalıştığından emin olun.
* **Kullanılabilirlik Testleri**:
  + Potansiyel kullanıcılarla kullanılabilirlik testleri yapın ve kullanıcı arayüzü ile genel deneyim hakkında geri bildirim toplayın.
  + Geri bildirimlere göre UI'yi iyileştirerek kullanım kolaylığını ve erişilebilirliği artırın.

#### ****8. Belgeleme & Bitirme****

* **Kullanıcı Kılavuzu**:
  + Uygulamanın nasıl kullanılacağını, metin girişini nasıl yapılacağını ve sonuçların nasıl yorumlanacağını açıklayan bir kılavuz oluşturulabilir.
* **Geliştirici Belgeleri**:
  + Kod tabanını, kurulum talimatlarını, API uç noktalarını ve bağımlılıkları içeren geliştirici belgeleri hazırlayın.
* **Son Rapor & Sunum**:
  + Projenin genel yapısını, mimarisini, geliştirme sürecini ve test sonuçlarını içeren bir final raporu hazırlayın.