Öğrenci Bilgi Sistemi Kodlama Adımı

**1. Kodlama Planının Yapılması**

* **Planlama**:
  + ÖBS'nin işlevleri ve gereksinimleri doğrultusunda hangi modüllerin ve bileşenlerin kodlanacağı belirlenir.
  + Örneğin: Öğrenci kaydı, ders seçimi, not görüntüleme modülleri.
* **Zaman Çizelgesi**:
  + Kodlama süreci için bir takvim oluşturulur.
  + Örneğin: Kullanıcı arayüzü için bir hafta, veritabanı bağlantıları için iki hafta gibi.

**2. Modül ve Sınıf Yapısının Oluşturulması**

* **Modüler Yaklaşım**:
  + ÖBS farklı modüllere bölünerek her bir modül için ayrı sınıflar oluşturulur.
  + Örneğin:
    - Ogrenci sınıfı: Öğrenci bilgilerini tutar.
    - Ders sınıfı: Ders detaylarını içerir.
    - Notlar sınıfı: Not kayıtlarını yönetir.
* **Soyutlama ve Kalıtım**:
  + Kodda tekrarları önlemek için ortak özellikler soyut sınıflar veya arayüzlerle tasarlanır.

**3. Fonksiyonların Geliştirilmesi**

* **İşlevlerin Kodlanması**:
  + Her modülde belirli görevleri yerine getiren fonksiyonlar yazılır.
  + Örneğin:
    - ogrenciEkle(): Yeni bir öğrenci ekler.
    - dersKaydet(): Öğrencinin ders kaydını yapar.
    - notGoruntule(): Öğrenci notlarını listeler.
* **Güvenlik Kontrolleri**:
  + Kullanıcı doğrulaması ve giriş kontrolleri eklenir.
  + Örneğin: Yetkisiz bir kişinin başka bir öğrencinin notlarına erişmesi engellenir.

**4. Veritabanı ve Diğer Harici Bağlantıların Kodlanması**

* **Veritabanı İşlemleri**:
  + Öğrenci, öğretmen, ders ve not bilgileri gibi verilerin saklanacağı ilişkisel bir veritabanı tasarlanır ve bağlanır.
  + Örneğin: SQL ile ogrenciler ve dersler tablolarının oluşturulması.
* **API ve Dış Bağlantılar**:
  + Harici servislerle entegrasyon sağlanır.
  + Örneğin: Öğrencilerin ders programlarını dış bir sistemden çekmek için REST API kullanılması.

**5. Kodun Test Edilmesi ve Hata Ayıklama (Debugging)**

* **Unit Testleri**:
  + Her fonksiyonun bağımsız olarak çalıştığından emin olmak için testler yazılır.
  + Örneğin: ogrenciEkle() fonksiyonu, yalnızca doğru verilerle çalışmalı ve hata vermemeli.
* **Hata Ayıklama**:
  + Tespit edilen hatalar düzeltilir.
  + Örneğin: Veri girişindeki hataların programı çökertmesini engellemek.

**6. Optimizasyon**

* **Performans İyileştirmeleri**:
  + Veritabanı sorguları ve kod, hızlı çalışacak şekilde optimize edilir.
  + Örneğin: Tablolar arasında gereksiz sorguların azaltılması.
* **Kaynak Kullanımı**:
  + Yazılımın minimum sistem kaynaklarıyla çalışması sağlanır.

**7. Kodun Gözden Geçirilmesi (Code Review)**

* **Takım İncelemesi**:
  + Yazılan kod, ekip üyeleri tarafından kontrol edilir.
  + Örneğin: Modüllerin birbiriyle uyumlu çalıştığının kontrolü.
* **Kod Kalitesi**:
  + Kodun okunabilirliği ve sürdürülebilirliği değerlendirilir.

**8. Dökümantasyon**

* **Teknik Dokümantasyon**:
  + Kodun nasıl çalıştığını ve hangi mantıkla yazıldığını açıklayan belgeler hazırlanır.
  + Örneğin: ogrenciEkle() fonksiyonunun parametreleri ve dönüş değerleri açıklanır.
* **Kullanıcı Kılavuzları**:
  + Öğrenciler ve personel için yazılımın kullanımına dair rehberler oluşturulur.