Aşağıdaki talimatları **aynen** uygulayarak Flutter ile (iOS + Android) çalışan bir "Ehliyet Sınavı Hazırlık" uygulamasının **tam çalışan temelini** üret. Sen kıdemli bir Flutter mühendisisin. Kod **Flutter 3.x** ve **Dart** ile yazılacak, **Material 3** kullanılacak. Proje; ekranlar, durum yönetimi, veri modeli, local JSON'dan veri okuma, deneme sınavı akışı, sonuç istatistikleri, yer imleri ve ayarlar ile birlikte çalışır halde gelmeli.

---

1) Proje ve Teknoloji Seçimleri

- **Framework:** Flutter 3.x (Dart)
- **UI Kiti:** Material 3
- **Durum Yönetimi:** `provider`
- **Kalıcı Depolama (küçük ayarlar/yer imleri):** `shared_preferences`
- **Yerel veri okuma:** `rootBundle` üzerinden `assets/*.json`
- **Paket Önerileri:**
- `provider`
- `shared_preferences`
- `google_fonts` (font için, opsiyonel)
- **Hedef Platformlar:** iOS ve Android (tek kod tabanı)

Not: Kodun derlenebilir olması için `pubspec.yaml` içine gerekli bağımlılıkları ve `assets` tanımlarını ekle.

---

2) Tasarım & Tema (Arka Plan Rengi Dahil)

- **Tema:** Aydınlık ve karanlık tema desteği.
- **Aydınlık Tema:**
- Arka plan: `#F8FAFC`
- Primary: `#2563EB` (mavi)
- Secondary/Accent: `#22C55E` (doğru), `#EF4444` (yanlış)
- Metin: `#0F172A`
- **Karanlık Tema:**
- Arka plan: `#0B1320`
- Primary: `#60A5FA`
- Secondary/Accent: `#34D399` (doğru), `#F87171` (yanlış)
- Metin: `#E5E7EB`
- **Buton Stili:** Büyük dokunma alanı, köşeler `12-16px` radius.
- **Yazı Tipi:** `Inter` (google_fonts ile), bulunamazsa varsayılan Material.
- **İkonlar:** Material Icons. Doğru/yanlış geri bildirimleri için net renk ve ikon kullan.

---

3) Veri Modeli ve JSON Şeması

- Uygulama, şimdilik **yerel JSON** dosyalarından soruları okur. Daha sonra dış kaynaktan import edilecektir.
- JSON şeması (örnek):

```
{
  "id": 1,
  "kategori": "Trafik ve Çevre",
  "soru": "Emniyet kemerinin temel amacı nedir?",
  "secenekler": { "A": "...", "B": "...", "C": "...", "D": "..." },
  "dogru": "C",
  "aciklama": "Kısa rasyonel açıklama (opsiyonel)"
}
```

- Kategoriler: **Trafik ve Çevre**, **Motor ve Araç Tekniği**, **İlk Yardım**, **Trafik Adabı**.
- Dosya yerleri:
- `assets/data/trafik_50.json`
- `assets/data/motor_50.json`
- `assets/data/ilkyardim_50.json`
- `assets/data/tefrik_adabi_50.json` (dosya adı: `trafik_adabi_50.json`)
- Başlangıçta bu dosyalar boş gelebilir; ancak **kod**, bu yapıya uygun JSON'u okuyup uygulamada gösterir halde olmalı.

---

4) Uygulama Mimarisi ve Dizin Yapısı

Örnek dizin:

```
lib/
  main.dart
  core/
    theme/app_theme.dart
    utils/result_calculator.dart
  data/
    models/question.dart
    repositories/question_repository.dart
    sources/local_json_loader.dart
  features/
    home/home_screen.dart
    category/category_screen.dart
    quiz/quiz_screen.dart
    quiz/widgets/
      option_tile.dart
      progress_bar.dart
    result/result_screen.dart
    review/review_screen.dart
    bookmarks/bookmarks_screen.dart
    settings/settings_screen.dart
  providers/
    quiz_provider.dart
    settings_provider.dart
    bookmarks_provider.dart
```

Bileşen Açıklamaları

- `question.dart`: Model (id, kategori, soru, secenekler Map\<String,String>, dogru, aciklama?)
- `local_json_loader.dart`: `assets` altından JSON dosyalarını okuyup `List<Question>` döndürür.
- `question_repository.dart`: Kategoriye göre soruları sağlayan katman. Gelecekte API/Sheets için genişletilebilir.
- `quiz_provider.dart`: Quiz akışının state'i (aktif soru index, işaretlenen şık, doğru sayısı, sayaç, durum vb.)
- `settings_provider.dart`: Ayarlar (zamanlayıcı açık/kapalı, şıkları karıştır, soru sırasını karıştır vb.)
- `bookmarks_provider.dart`: Yer imleri (soruları kaydet, listele) – `shared_preferences` ile id listesi saklanır.

---

5) Ekranlar ve Akış

5.1 Home Screen

• Başlık: "Ehliyet Sınavı Hazırlık"
• Dört kategori kartı (ikon + başlık + toplam soru sayısı)
• Hızlı erişim: "Deneme Sınavı", "Son Çalışma", "Yer İmleri", "Ayarlar"

5.2 Category Screen

• Seçilen kategoriye ait sorularla **hızlı test** veya **deneme** başlatma.
• Filtreler: "Sıra karıştır", "Şıklar karışsın", "Zamanlayıcı açık/kapalı".

5.3 Quiz Screen

• Üstte ilerleme: "Soru 5/50" + progress bar.
• (Opsiyonel) Zamanlayıcı göstergesi (ör. 45:00) – Deneme modunda aktif.
• Soru metni, altında 4 şık (A-D). Dokununca seçilir, tekrar dokunulursa değiştirilebilir.
• **Sonraya bırak / Bayrak ikonu** ile işaretleme.
• Alt butonlar: "Önceki", "Sonraki", "Bitir".
• Ayarlara göre şıkların ve soruların sırası karışık olabilir.
• (Opsiyonel) Anında doğru/yanlış geri bildirimi **kapalı**; sonuç ekranında gösterilecek.

5.4 Result Screen

• Doğru / Yanlış / Boş sayısı, başarı yüzdesi.
• Pasta/çubuk benzeri basit özet (yalın UI, ekstra paket gerekmeden basit çubuk kullanılabilir).
• "Yanıtları Gözden Geçir" butonu (Review Screen'e gider).
• "Yer İmlerine Ekle" (yanlış/emin olunmayan soruları topluca ekleme).

5.5 Review Screen

• Soru + Kullanıcı yanıtı + Doğru yanıt + (varsa) kısa açıklama.
• "Yer imine ekle/çıkar" butonu.

5.6 Bookmarks Screen

- Yer imi eklenmiş soruların listesi.
- Tek tek çözme veya mini test başlatma.

5.7 Settings Screen

- Zamanlayıcı: Aç/Kapalı (Varsayılan: Açık, deneme 50 soru → 45 dk)
- Soru sırasını karıştır: Aç/Kapalı (Varsayılan: Açık)
- Şıkları karıştır: Aç/Kapalı (Varsayılan: Açık)
- Tema: Aydınlık / Karanlık / Sistem

---

6) İş Kuralları ve Davranışlar (Türkiye Ehliyet Sınavına Uygun)

- **Deneme sınavı**: 50 soru, 45 dakika. Sayaç bittiğinde sınav otomatik biter ve sonuç ekranına gider.
- **Puanlama**: Doğru=1, Yanlış=0, Negatif puanlama yok. Boşlar 0.
- **Kategori testleri**: Kategoriye özel 10/20/30/50 soruluk hızlı test opsiyonları olabilir.
- **Erişilebilirlik**: Butonlar yeterli boyutta; ekran okuyucu etiketleri; renk körlüğü dostu geri bildirim (renk + ikon).

---

7) Kod Ayrıntıları (İstenenler)

- ``: Gerekli bağımlılıkları ve `assets/data/` klasörünü tanımla.
- **Model & Mapper**: `Question.fromJson(Map)` ve `toJson()`.
- **QuestionRepository**: `Future<List<Question>> getByCategory(String kategori)`; local JSON'dan okur.
- **QuizProvider (ChangeNotifier)**: alanlar → `questions`, `currentIndex`, `selectedOptions` (Map\<int,String>), `correctCount`, `isFinished`, `flagged` (Set), `timerSeconds` vb. Metodlar → `selectOption()`, `next()`, `prev()`, `finish()`, `toggleFlag()`.
- **SettingsProvider (ChangeNotifier)**: `shuffleQuestions`, `shuffleOptions`, `timerEnabled`, `themeMode`.
- **BookmarksProvider (ChangeNotifier)**: `savedIds` (Set), `toggle(id)`, `isSaved(id)`. `shared_preferences` ile persist.
- **LocalJsonLoader**: `load(categoryAssetPath)`: `rootBundle.loadString()` → `jsonDecode` → `List<Question>`.
- **Tema**: `AppTheme.lightTheme` ve `AppTheme.darkTheme` sınıfları.
- **Routing**: Named routes veya `go_router` kullanmadan basit `Navigator.push` yeterli.

---

8) Örnek Veri (Assets)

- `assets/data/trafik_50.json`, `motor_50.json`, `ilkyardim_50.json`, `trafik_adabi_50.json` – şu an boş olabilir.
- Kod; dosya yoksa kullanıcıya uyarı veren basit bir boş-durum (empty state) göstersin.

---

9) Geleceğe Hazırlık (Import İçin Kancalar)

- `QuestionRepository` içine `` gibi bir iskelet ekle.
- Yarın; Google Sheets API, Firebase Firestore ya da bir REST API'den gelen veriyi bu metotla ekleyebilelim.

---

10) Teslim Şartları

- Derlenebilir **tam Flutter projesi** (main.dart dahil tüm ekranlar, provider'lar ve theme).
- En azından **dummy** 3-5 soru ile akışın uçtan uca çalıştığını göster.
- README.md: Kurulum (Flutter versiyon), `flutter pub get`, `flutter run`, tema bilgisi, JSON şeması, gelecekte import notu.

---

11) Kabul Kriterleri (Acceptance)

- Uygulama açılır; Home → Category → Quiz → Result → Review akışı sorunsuz.
- Ayarlar sayfasında yapılan değişiklikler (ör. şık karıştırma) quiz'e yansır.
- Deneme modunda sayaç işler; süre bitince otomatik sonuç ekranına gider.
- Yer imleri kalıcıdır (uygulama kapanıp açılınca durur).
- Aydınlık/Karanlık tema düzgün çalışır; arka plan renkleri talep edildiği gibidir.

---

12) Ek Notlar

- Kod düzenli, yorumlu ve modüler olsun.
- UI yalın, temiz; görsel hiyerarşi ve boşluk kullanımı iyi olsun.
- Test cihazı yoğunluğu düşünülerek dokunma hedefleri en az 44x44 dp.

Şimdi bu gereksinimlerle uyumlu, derlenebilir bir Flutter projesi üret. Kod içinde gerekli yerlerde `TODO:` yorumlarıyla dış kaynaktan soru importuna hazırlık noktalarını işaretle.