* flutter/material.dart: Flutter’ın Material Design bileşenleri için
* flutter/widgets.dart: Düşük seviyede widget’lar oluşturmak için
* const arkaplanRenkim = Color.fromARGB(0, 31, 45, 71); Sabit bir renk tanımlama yapıyoruz burada
* import 'package:flutter/material.dart'; Ana kütüphane
* import 'package:go\_router/go\_router.dart'; Sayfalar arası geçiş yapmak için
* import 'package:xenolauncher/screens/history\_screen.dart'; HistoryScreen adında bir ekran tanımlayıp bunu screen klasöründe yer aldığını gösterir
* **import 'package:xenolauncher/screens/search\_screen.dart'; Projemizde yer alan SearchScreen(arama ekranı) adında bir ekranımızın bulunduğunu gösteriyor**
* import 'package:xenolauncher/screens/voice\_screen.dart'; VoiceScreen(ses ekranı) için tanımlanmış ekranımızı projemizde bulunduğunu gösteriyor.
* import '../screens/loading\_screen.dart'; LoadingScreen (yükleme ekranı)screens klasöründe bulunur ve kullanıcıya göstermek için bir ekran sağlar
* import '../screens/home\_screen.dart'; projenin ana ekranını oluşturur
* import '../screens/profile\_screen.dart'; kullanıcının profil bilgilerini göstermek ve düzenlemek için oluşturduğumuz bir ekrandır
* final router = GoRouter(): GoRouter sınıfından nesne oluşturarak uygulama içinde sayfalar arasında geçiş yapmamızı sağlar (final kullanımı router nesnesinin değiştirilemez sabit kalmasını sağlar)
* initialLocation: '/' Başlangıç rotasıdır
* GoRoute path: URL yoludur hangi sayfaya gitmek istiyorsak bu bileşenden yararlanırız
* GoRoute builder: Belirtilen rotada hangi widget’ın gösterileceğini belirler
* class HistoryScreen extends StatefulWidget: HistoryScreen bir statefulWidget olarak tanımlanıp dinamik olarak değişen bir arayüz sağlar. StatefulWidget ise widget tanımı ve durum sınıfı olarak iki parçadan oluşur
* const HistoryScreen({super.key}); HistoryScreen için constructor yani bir yapıcı tanımladık

super.key => widget’ların benzersiz anahtarlarla yönetilmesini sağlar.

* State<HistoryScreen> createState() => \_HistoryScreenState();\_HistoryScreenState sınıfı kullanılarak StatefulWidget’ın durumunu yönetecek olan bir sınıf döndürür
* class \_HistoryScreenState extends State<HistoryScreen> : Bu sınıf HistoryScreen için durum tanımlar ve yönetir
* @override Widget build(BuildContext context):

Build: widget ağacının bir parçasını döndürür

Context: widget’ın konumunu ve meta bilgilerini içerir

* return Scaffold: Material design kurallarına uygun bşr ekran iskeleti sağlamakta
* body: Center Ekranın ana içeriğini sağlar
* bottomNavigationBar: BottomMenu()

bottomNavigationBar Ekranın alt kısmında bir gezinme çubuğunu tanımlamakta

**BottomMenu()** uygulama içindeki alt gezinme çubuğu işlevini sağlar

* @override Widget build(BuildContext context): build metodu ile kullanıcı arayüzü oluşturur. Context, widget’ın widget ağacındaki konumunu ve ilişki meta bilgilerini içerir
* Dotlottie\_loader: .lottie formatındaki animasyon dosyalarını yüklemek için
* Body: SizedBox.expand widget’ın hem genişlik (width) hem de yükseklik (height) değerini ekran boyutuna göre ayarlarlar
* Column: widget’ları dikey olarak hizalamak için
* Expanded Logo widget’ına kalan alanın tamamını kaplamak için kullanırız
* Image.asset:

**'assets/images/logo.webp':** Logo olarak kullanılacak görsel dosyası

**BoxFit.contain:** Görseli orijinal boyutlarını koruyarak widget içine sığdırır

* InkWell: Tıklama (ontap) olayını algılayan widget.
* DotLottieLoader.fromAsset:

assets/motions/loading.lottie Yükleme animasyon dosyası

frameBuilder Animasyon dosyasını yüklediğimizde nasıl bir yapıya dönüştüğünü bize belirler

Lottie.memory Yüklü olan animasyonu bellekte çalıştırır

* SizedBox(height: 1) Animasyon ile ekranın altı arasında bir boşluk bırakmak için kullanılır
* **AppBar:** Uygulama çubuğu
* **Drawer:** Yan menüyü oluşturur
* **Container:** Üst kısımda bir kullanıcı simgesi (icon) barındırır
* **ListTile:** Menü öğeler
* **Column:** İçeriği dikey olarak düzenler
* DotLottieLoader.fromAsset: q2.lottie dosyasından bir animasyon yükler eğer yükleme başarılı olursa lottie.memory ile animasyonu oynatır başarısız olursa boş bir container döndürür
* **Row:** İkonları yatay olarak hizalar