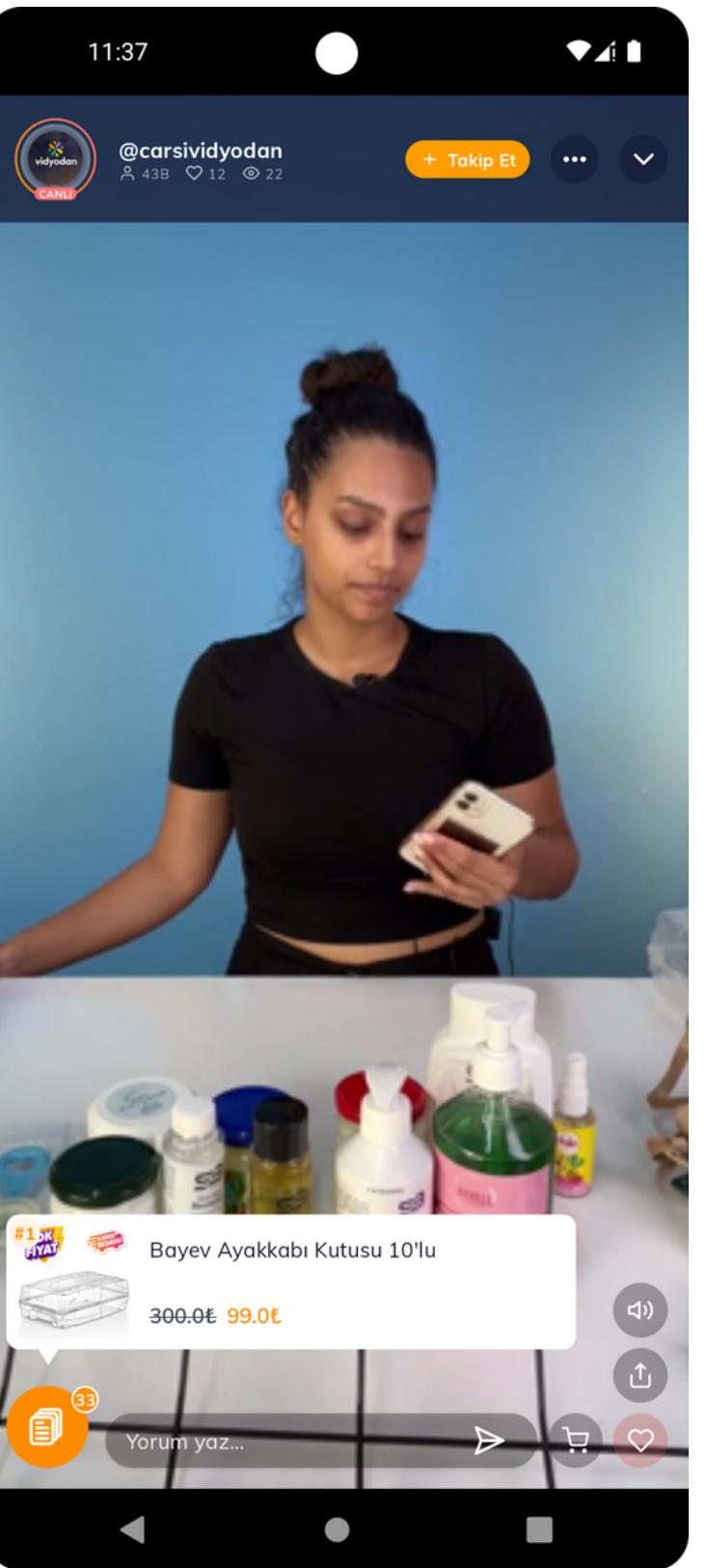
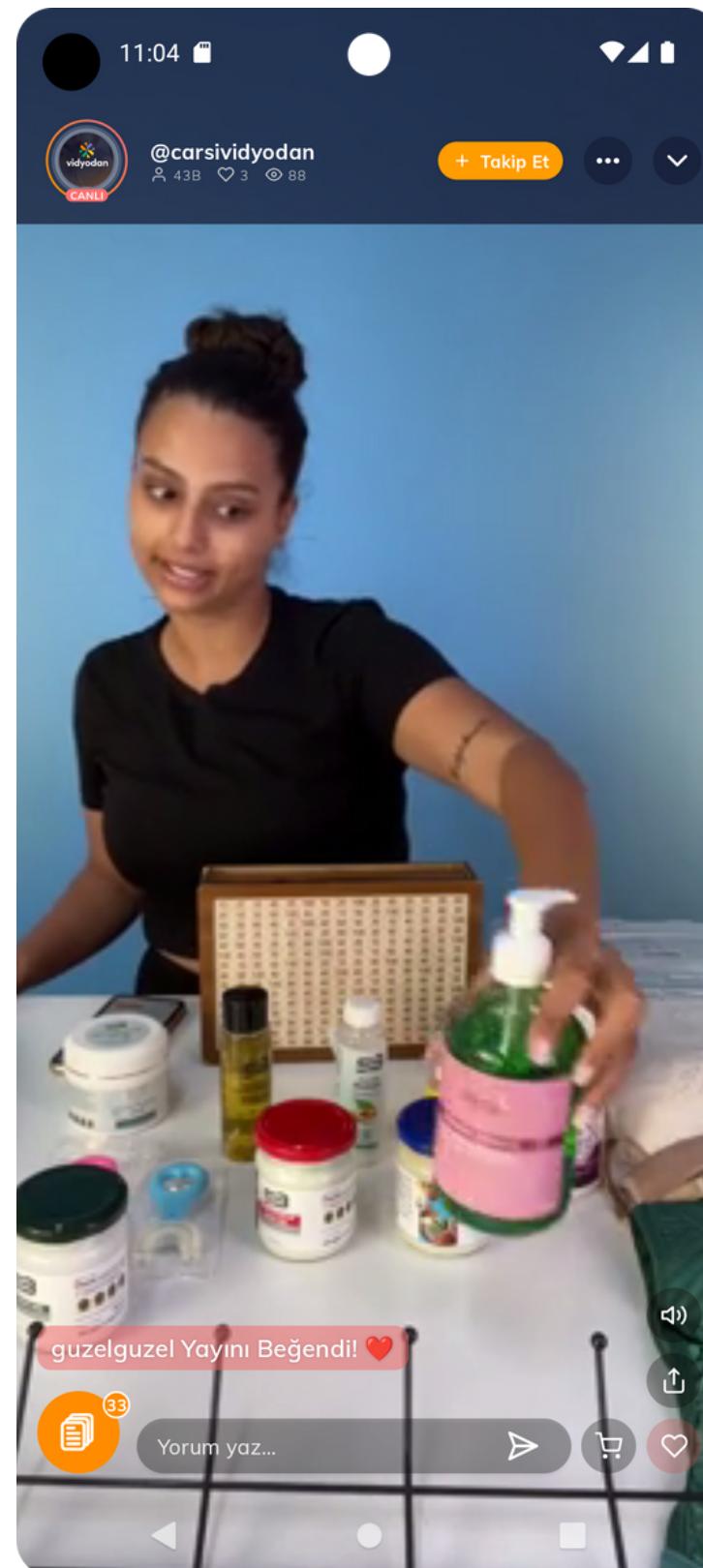


Regular



Edge to Edge



Adım 1: Tema flaglerinin setlenmesi

View

```
WindowCompat.setDecorFitsSystemWindows(window, decorFitsSystemWindows: false)
```

Compose

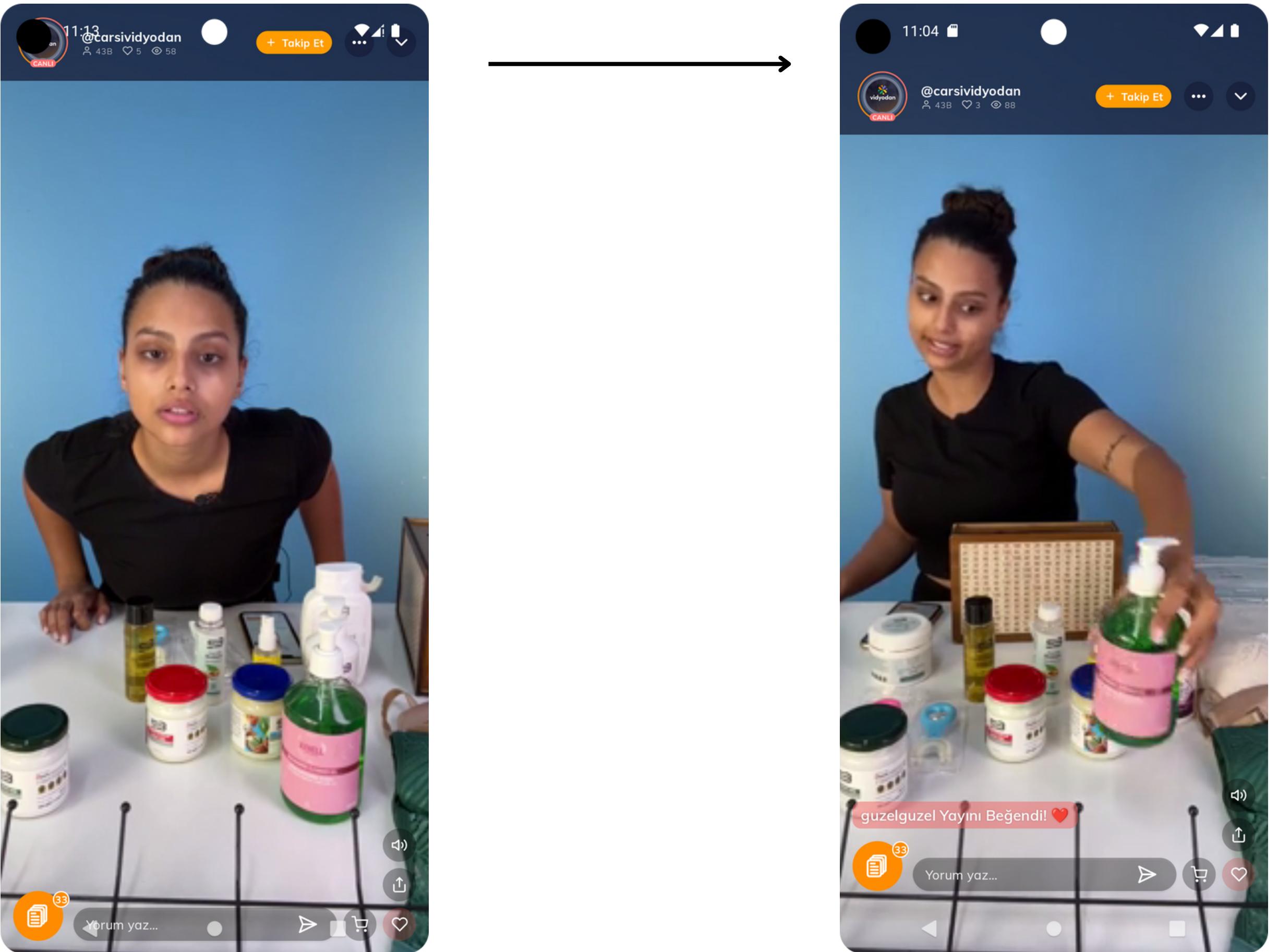
```
WindowCompat.setDecorFitsSystemWindows(window, decorFitsSystemWindows: false)  
//veya  
enableEdgeToEdge()
```

Adım 2: System Bar renk değişimi

```
<item name="android:statusBarColor">@android:color/transparent</item>
<item name="android:navigationBarColor">@android:color/transparent</item>
```

```
val view = LocalView.current
if (!view.isInEditMode) {
    SideEffect {
        val window = (view.context as Activity).window
        window.statusBarColor = Color.TRANSPARENT
        window.navigationBarColor = Color.TRANSPARENT
    }
}
```

Adım 3: Handling Window Insets



Handling Window Insets

View

```
ViewCompat.setOnApplyWindowInsetsListener(view) { view, windowInsets ->
    val insets = windowInsets.getInsets(WindowInsetsCompat.Type.systemBars())
    // Apply the insets as a margin to the view. This solution sets
    // only the bottom, left, and right dimensions, but you can apply whichever
    // insets are appropriate to your layout. You can also update the view padding
    // if that's more appropriate.
    view.updateLayoutParams<MarginLayoutParams>(
        leftMargin = insets.left,
        bottomMargin = insets.bottom,
        rightMargin = insets.right,
    )

    // Return CONSUMED if you don't want want the window insets to keep passing
    // down to descendant views.
    WindowInsetsCompat.CONSUMED
}
```

[View Doc](#)

Compose

```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier.statusBarsPadding()
    )
}
```

[Compose Doc](#)

Handling Window Insets - Some of the views / composables are coded to dodge system bars as default behaviour.

BottomNavigationView.java kaynak kodu

```
private void applyWindowInsets() {
    ViewUtils.doOnApplyWindowInsets(
        this,
        new ViewUtils.OnApplyWindowInsetsListener() {
            @NonNull
            @Override
            public WindowInsetsCompat onApplyWindowInsets(
                View view,
                @NonNull WindowInsetsCompat insets,
                @NonNull RelativePadding initialPadding) {
                // Apply the bottom, start, and end padding for a BottomNavigationView
                // to dodge the system navigation bar
                initialPadding.bottom += insets.getSystemWindowInsetBottom();

                boolean isRtl = ViewCompat.getLayoutDirection(view) == ViewCompat.LAYOUT_DIRECTION_RTL;
                int systemWindowInsetLeft = insets.getSystemWindowInsetLeft();
                int systemWindowInsetRight = insets.getSystemWindowInsetRight();
                initialPadding.start += isRtl ? systemWindowInsetRight : systemWindowInsetLeft;
                initialPadding.end += isRtl ? systemWindowInsetLeft : systemWindowInsetRight;
                initialPadding.applyToView(view);
            }
            return insets;
        }
    );
}
```

Compose Material3'de insetleri otomatik handle eden componentler

Inset handling composables

Below is a list of the [Material Components](#) that automatically handle insets.

App bars

- [TopAppBar](#) / [SmallTopAppBar](#) / [CenterAlignedTopAppBar](#) / [MediumTopAppBar](#) / [LargeTopAppBar](#): Applies the *top* and *horizontal* sides of the system bars as padding since it is used at the top of the window.
- [BottomAppBar](#): Applies the *bottom* and *horizontal* sides of the system bars as padding.

Content containers

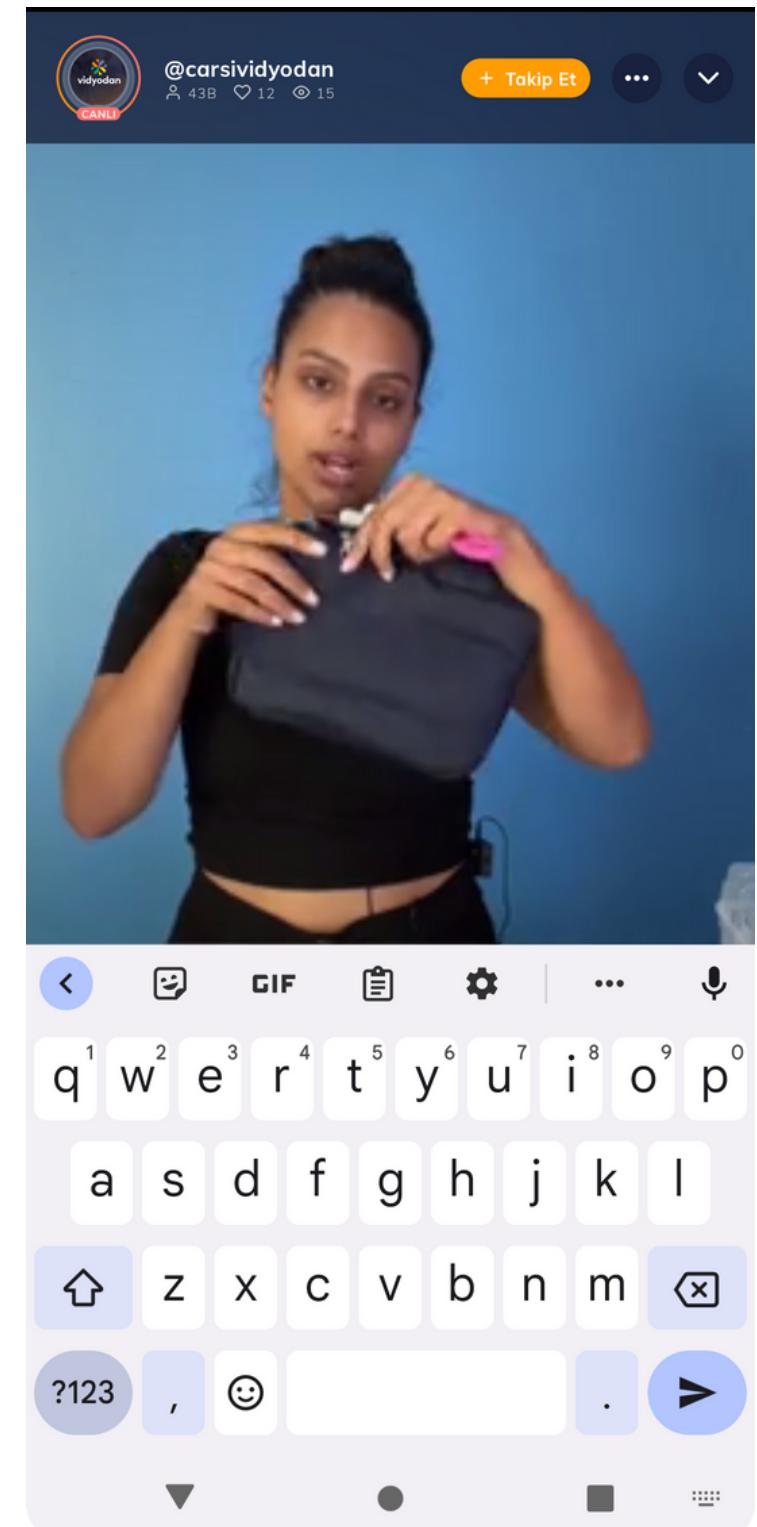
- [ModalDrawerSheet](#) / [DismissibleDrawerSheet](#) / [PermanentDrawerSheet](#) (content inside a modal navigation drawer): Applies *vertical* and *start* insets to content.
- [ModalBottomSheet](#): Applies the *bottom* insets.
- [NavigationBar](#): Applies the *bottom* and *horizontal* insets.
- [NavigationRail](#): Applies the *vertical* and *start* insets.

Scaffold

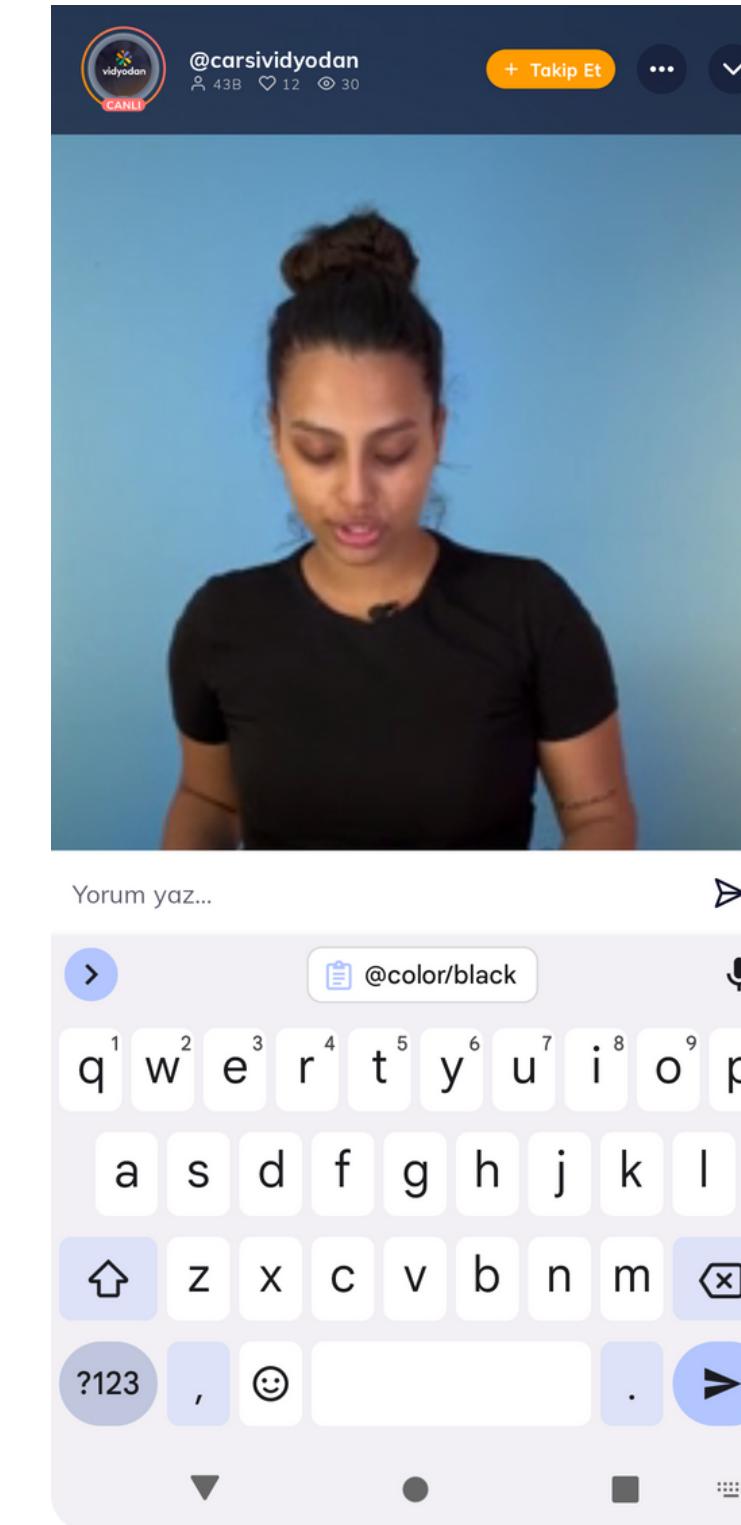
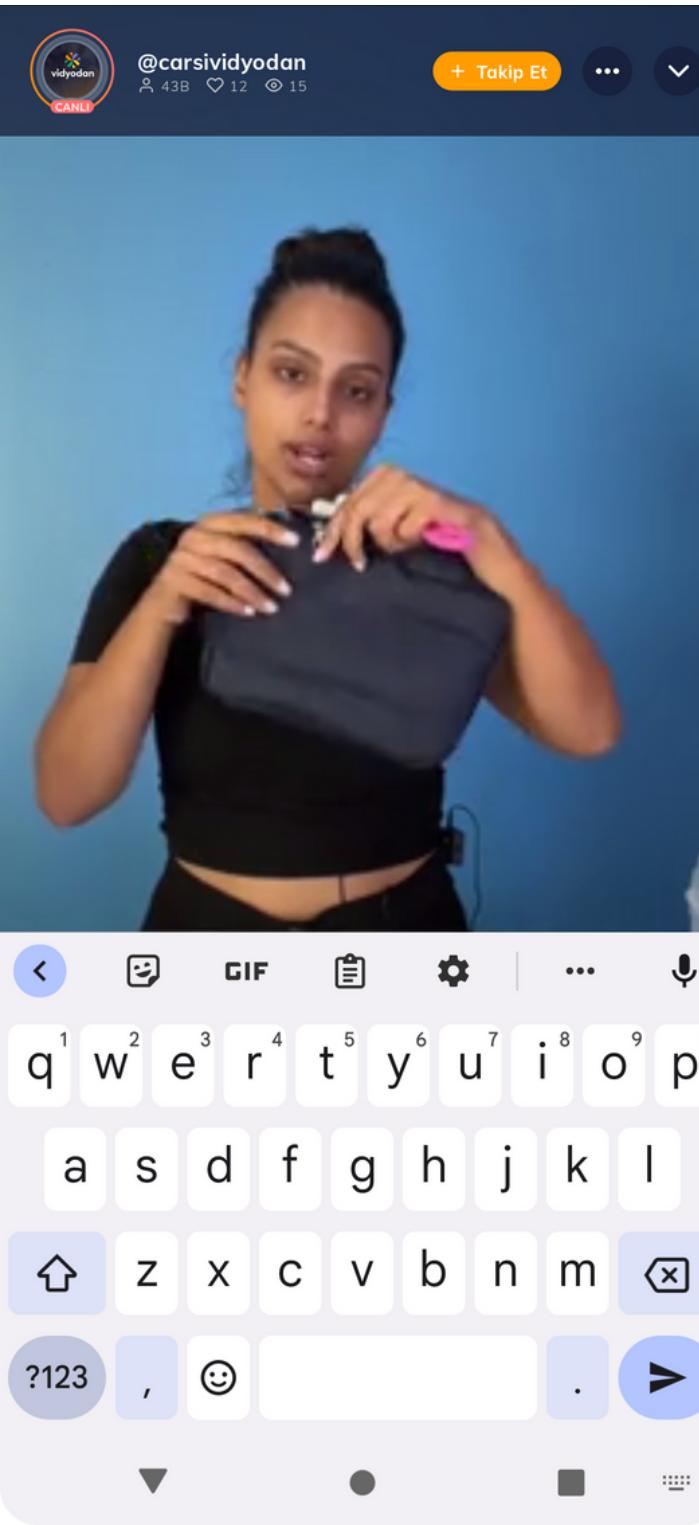
By default, [Scaffold](#) provides insets as parameter `paddingValues` for you to consume and use. [Scaffold](#) does not apply the insets to content; this responsibility is yours. For example, to consume these insets with a [LazyColumn](#) inside a [Scaffold](#):

Handling Window Insets - Klavye

Edge to Edge ‘temasında’ iken
adjustResize input mode'u
çalışmıyor. Elementler klavye
arkasında kalıyor.



Handling Window Insets - Klavye



Handling Window Insets - Klavye

View

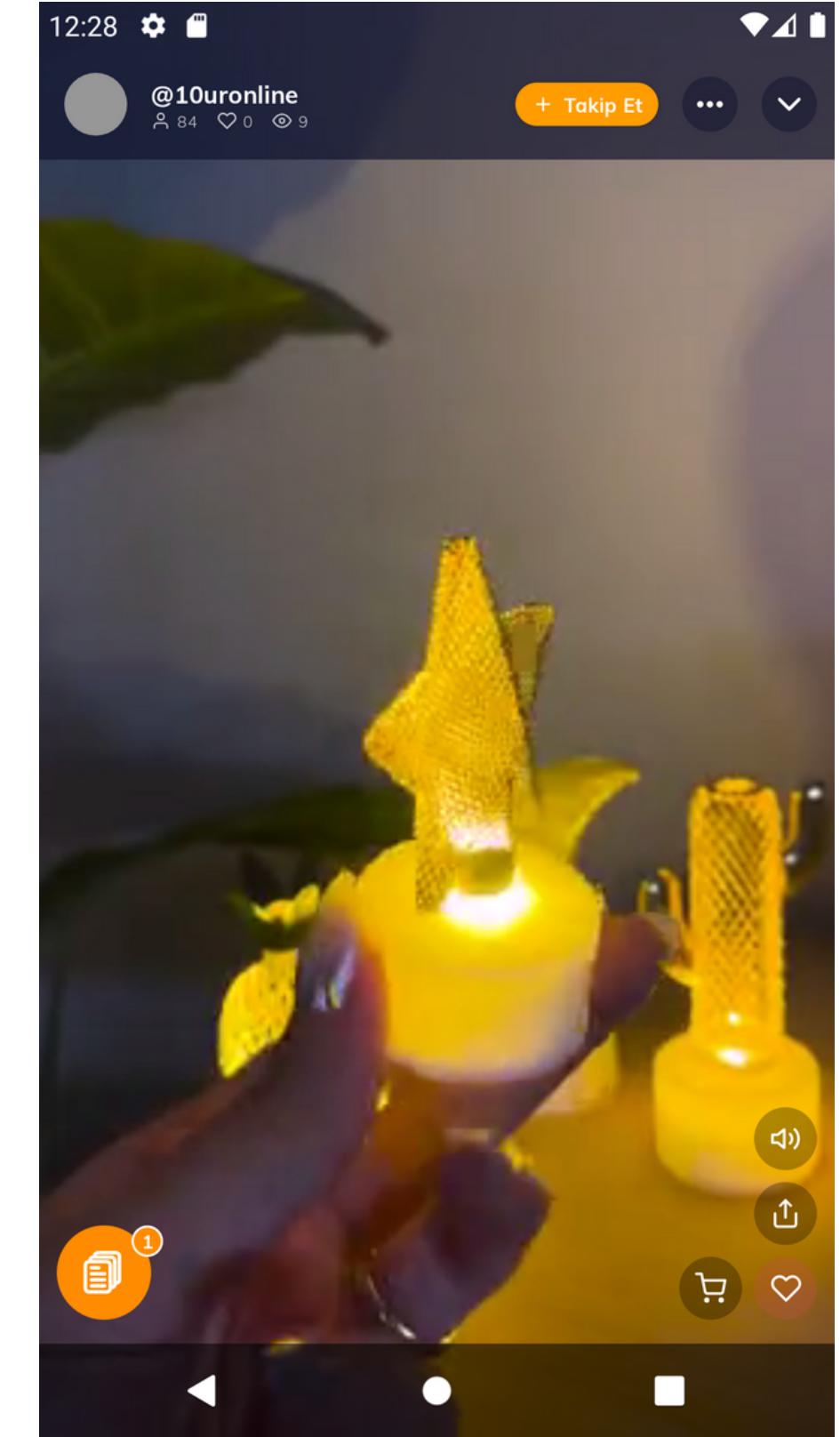
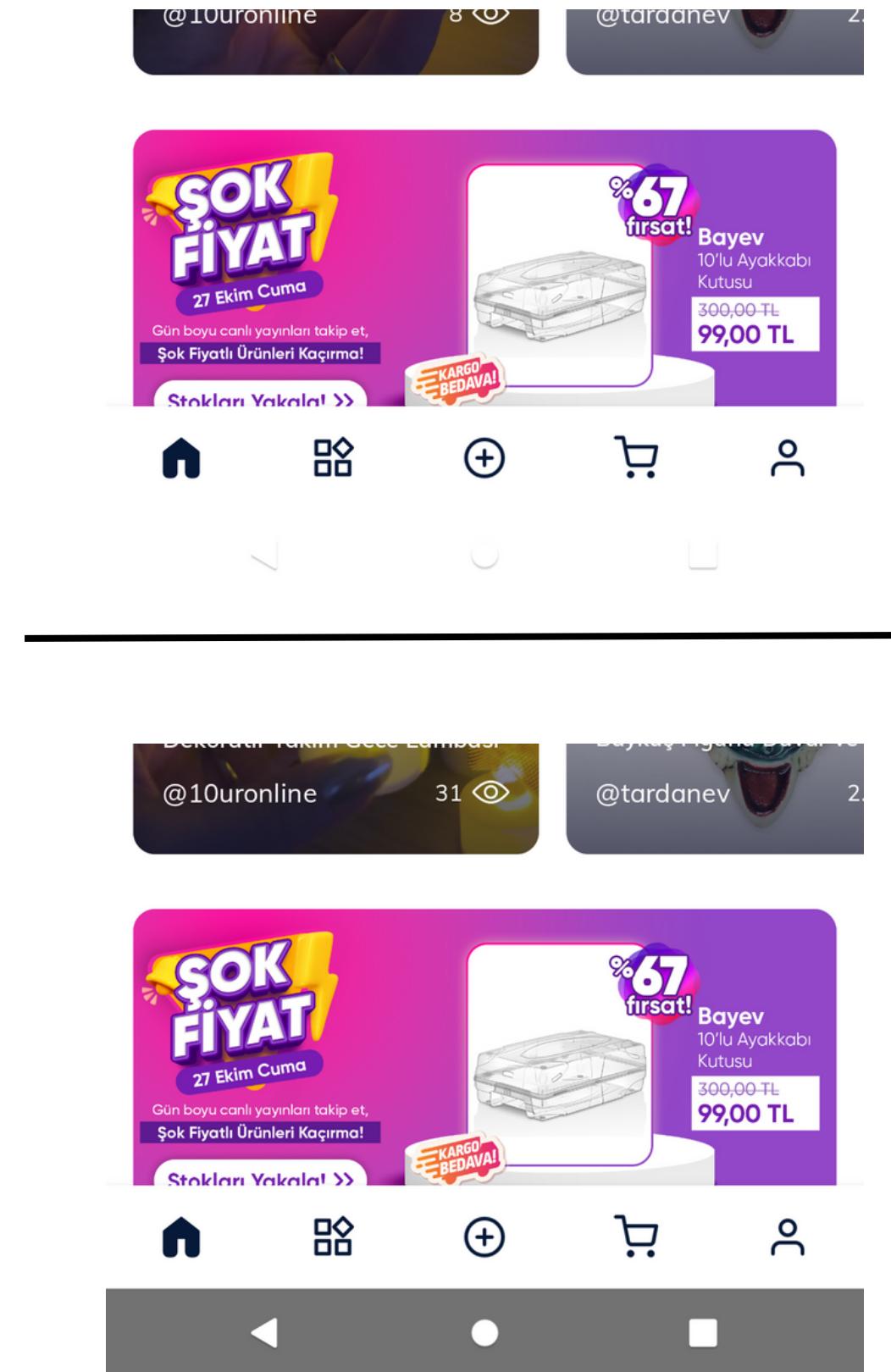
```
ViewCompat.setOnApplyWindowInsetsListener(view) { _, insets ->
    val imeVisible = insets.isVisible(WindowInsetsCompat.Type.ime())
    val imeHeight = insets.getInsets(WindowInsetsCompat.Type.ime()).bottom
    insets
}
```

Compose

```
Box { this: BoxScope
    TextField(
        modifier = Modifier.align(Alignment.BottomCenter)
            .imePadding()
            .navigationBarsPadding(),
        value = "",
        onValueChange = {})
}
```

System Bar Tint

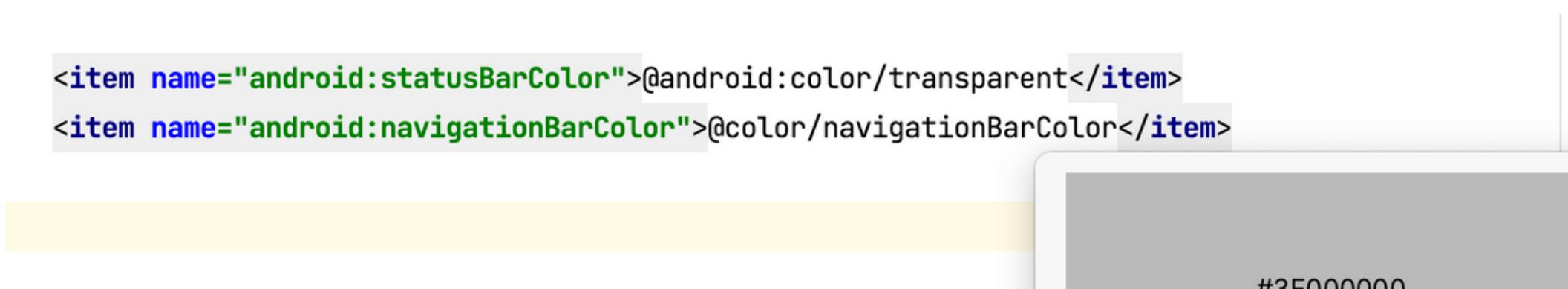
Api versiyon 27 öncesinde navigation bar content rengini değiştiremiyoruz, hardcoded beyaz. Bu durum, transparent navigation bar ile birleştiğinde istenmeyen görüntüye neden olabiliyor. En common çözüm ise bu versiyon özelinde navigasyon barı yarı transparan yapmak.



System Bar Tint - View

/styles/

```
<item name="android:statusBarColor">@android:color/transparent</item>
<item name="android:navigationBarColor">@color/navigationBarColor</item>
```



#3E000000

A screenshot of an Android application interface. The top part shows a yellow status bar. Below it is a grey navigation bar. To the right of the navigation bar, there is some text: "#3E000000".

/styles/v27

```
<item name="android:windowLightNavigationBar">true</item>
<item name="android:navigationBarColor">@color/transparent</item>
```

System Bar Tint - Compose, YTD

```
enableEdgeToEdge(statusBarStyle = SystemBarStyle.auto(  
    lightScrim = android.graphics.Color.DKGRAY,  
    darkScrim = android.graphics.Color.LTGRAY  
)  
)
```

Compose Activity'e özel enableEdgeToEdge fonksiyonu hem edge-to-edge temasını ayarlıyor, hem de aldığı opsiyonel scrim renk parametreleriyle bahsettiğimiz versiyon farkından oluşan görünümü handle ediyor.