

```
1 package com.example.MyYoutubeApp_20BCE7010
2
3 import android.app.Activity
4 import android.content.Context
5 import android.content.ContextWrapper
6 import android.content.pm.ActivityInfo
7 import android.os.Bundle
8 import androidx.activity.ComponentActivity
9 import androidx.activity.compose.setContent
10 import androidx.compose.foundation.Image
11 import androidx.compose.foundation.isSystemInDarkTheme
12 import androidx.compose.foundation.layout.fillMaxSize
13 import androidx.compose.foundation.layout.padding
14 import androidx.compose.foundation.layout.size
15 import androidx.compose.foundation.shape.RoundedCornerShape
16 import androidx.compose.foundation.shape.RoundedCornerShape
17 import androidx.compose.material3.Button
18 import androidx.compose.material3.ExperimentalMaterial3Api
19 import androidx.compose.material3.Icon
20 import androidx.compose.material3.MaterialTheme
21 import androidx.compose.material3.OutlinedTextField
22 import androidx.compose.material3.Surface
23 import androidx.compose.material3.Text
24 import androidx.compose.runtime.Composable
25 import androidx.compose.runtime.DisposableEffect
26 import androidx.compose.runtime.getValue
27 import androidx.compose.runtime.mutableStateOf
28 import androidx.compose.runtime.remember
29 import androidx.compose.runtime.setValue
30 import androidx.compose.ui.Modifier
31 import androidx.compose.ui.graphics.Color
32 import androidx.compose.ui.platform.LocalContext
```

```

File - C:\Users\Akhilesh\Downloads\My Youtube App 20BCE7010\My Youtube App 20BCE7010\app\src\main\java\com\example\MyYoutubeApp_20BCE7010\MainActivity.kt

33 import androidx.compose.ui.res.painterResource
34 import androidx.compose.ui.unit.dp
35 import androidx.compose.ui.viewinterop.AndroidView
36 import androidx.constraintlayout.compose.ConstraintLayout
37 import androidx.constraintlayout.compose.ConstraintLayout
38 import androidx.navigation.NavHostController
39 import androidx.navigation.compose.NavHost
40 import androidx.navigation.compose.composable
41 import androidx.navigation.compose.rememberNavController
42 import com.example.compose.AppTheme
43 import com.example.youchube.R
44 import com.google.accompanist.systemuicontroller.rememberSystemUiController
45 import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.YouTubePlayer
46 import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.listeners.AbstractYouTubePlayerListener
47 import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.views.YouTubePlayerView
48 import java.util.regex.Pattern
49
50 class MainActivity : ComponentActivity() {
51     override fun onCreate(savedInstanceState: Bundle?) {
52         super.onCreate(savedInstanceState)
53         setContent {
54             AppTheme {
55                 // A surface container using the 'background' color from the theme
56                 Surface(
57                     modifier = Modifier.fillMaxSize(),
58                     color = MaterialTheme.colorScheme.background
59                 ) {
60                     Navi()
61                     Hide()
62                 }
63             }
64         }
65     }
66 }

```

```

File - C:\Users\Akhilesh\Downloads\My Youtube App 20BCE7010\My Youtube App 20BCE7010\app\src\main\java\com\example\MyYoutubeApp_20BCE7010\MainActivity.kt

65    }
66    }
67    @Composable
68    private fun Hide() {
69        val systemUiController = rememberSystemUiController()
70        if (isSystemInDarkTheme()) {
71            systemUiController.setSystemBarsColor(
72                color = MaterialTheme.colorScheme.primaryContainer
73            )
74        } else {
75            systemUiController.setSystemBarsColor(
76                color = Color.White
77            )
78        }
79    }
80    }
81    @Composable
82    fun Navi() {
83        val navigator = rememberNavController()
84        NavHost(navController = navigator, startDestination = "Body") {
85            composable(route = "Body") {
86                Body(Navigator = navigator)
87            }
88        }
89        composable(route = "Video/{VId}") {
90            val video = it.arguments?.getString("VId")
91            if (video != null) {
92                YouTubeScreen(videoId = video)
93            }
94        }
95    }
96}

```

```

File - C:\Users\Akhilesh\Downloads\My Youtube App 20BCE7010\My Youtube App 20BCE7010\app\src\main\java\com\example\MyYoutubApp_20BCE7010\MainActivity.kt

97
98 @OptIn(ExperimentalMaterial3Api::class)
99 @Composable
100 fun Body(Navigator: NavHostController) {
101     Card(
102         modifier = Modifier
103             .fillMaxSize()
104             .padding(20.dp)
105     ) {
106         ConstraintLayout(modifier = Modifier.fillMaxSize()) {
107             val (logo, url, button) = createRefs()
108             val vChain = createVerticalChain(logo, url, button, chainStyle = ChainStyle.Packed)
109             var URL by remember {
110                 mutableStateOf(" ")
111             }
112             var id: String? by remember {
113                 mutableStateOf(" ")
114             }
115
116             if (isSystemInDarkTheme()) {
117                 Image(painter = painterResource(id = R.drawable.yt_logo_rgb_dark),
118                     contentDescription = " ",
119                     modifier = Modifier
120                         .size(200.dp)
121                         .constrainAs(logo) {
122                             start.linkTo(parent.start)
123                             end.linkTo(parent.end)
124                         })
125             } else {
126                 Image(painter = painterResource(id = R.drawable.yt_logo_rgb_light),
127                     contentDescription = " ",
128                     modifier = Modifier

```

```

File - C:\Users\Akhilesh\Downloads\My Youtube App 20BCE7010\My Youtube App 20BCE7010\app\src\main\java\com\example\MyYoutubApp_20BCE7010\MainActivity.kt

129
130     .size(200.dp)
131     .constrainAs(logo) {
132         start.linkTo(parent.start)
133         end.linkTo(parent.end)
134     }
135
136     OutlinedTextField(
137         value = URL,
138         onValueChange = { URL = it },
139         label = { Text(text = "Paste the URL") },
140         modifier = Modifier
141             .padding(top = 100.dp)
142             .constrainAs(url) {
143                 start.linkTo(parent.start)
144                 end.linkTo(parent.end)
145             }
146
147     Button(onClick = {
148         id = urlParser(URL)
149         Navigator.navigate(route = "Video/$id")
150     },
151         shape = RoundedCornerShape(5.dp),
152         modifier = Modifier
153             .padding(30.dp)
154             .constrainAs(button) {
155                 start.linkTo(parent.start)
156                 end.linkTo(parent.end)
157                 bottom.linkTo(parent.bottom, margin = 600.dp)
158             }
159             Text(text = "Lets play")
160             Icon(

```

```

File - C:\Users\Akhilesh\Downloads\My Youtub App 20BCE7010\My Youtub App 20BCE7010\app\src\main\java\com\example\MyYoutubApp_20BCE7010\MainActivity.kt

161     painter = painterResource(id = R.drawable.baseline_play_arrow_24),
162     contentDescription = ""
163     )
164     }
165     }
166     }
167     }
168     )
169     @Composable
170     fun YoutubeScreen(
171         videoId: String
172     ) {
173     AndroidView(factory = {
174         val view = YouTubePlayerView(it)
175         val fragment = view.addYouTubePlayerListener(
176             object : AbstractYouTubePlayerListener() {
177                 override fun onReady(youTubePlayer: YouTubePlayer) {
178                     super.onReady(youTubePlayer)
179                     youTubePlayer.loadVideo(videoId, 0f)
180                 }
181             }
182         )
183         view
184         }, modifier = Modifier.padding(75.dp))
185     ScreenOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE)
186     }
187
188     @Composable
189     fun ScreenOrientation(orientation: Int) {
190         val context = LocalContext.current
191         DisposableEffect(orientation) {
192             val activity = context.findActivity() ?: return@DisposableEffect onDispose {}

```

```

193     val originalOrientation = activity.requestedOrientation
194     activity.requestedOrientation = orientation
195     onDispose {
196         // restore original orientation when view disappears
197         activity.requestedOrientation = originalOrientation
198     }
199 }
200 }

201 fun urlParser(url: String): String? {
202     var vId: String? = null
203     val pattern = Pattern.compile(
204         "^https://.*(?:youtu.be/v/u/\w|embed/watch?v=)([^#&?]*).*$",
205         Pattern.CASE_INSENSITIVE
206     )
207     val matcher = pattern.matcher(url)
208     if (matcher.matches()) {
209         vId = matcher.group(1)
210     }
211     return vId
212 }
213 }

214 fun Context.getActivity(): Activity? = when (this) {
215     is Activity -> this
216     is ContextWrapper -> baseContext.getActivity()
217     else -> null
218 }
219 }
```

```

File - C:\Users\Akhilesh\Downloads\My Youtube App 20BCE7010\My Youtube App 20BCE7010\app\src\main\java\com\example\MyYoutubeApp_20BCE7010\ui\theme\Typee.kt

1 package com.example.MyYoutubeApp_20BCE7010.ui.theme
2
3 import androidx.compose.material3.Typography
4 import androidx.compose.ui.text.TextStyle
5 import androidx.compose.ui.text.font.FontFamily
6 import androidx.compose.ui.text.font.FontWeight
7 import androidx.compose.ui.unit.sp
8
9 // Set of Material typography styles to start with
10 val Typography = Typography(
11     bodyLarge = TextStyle(
12         fontFamily = FontFamily.Default,
13         fontWeight = FontWeight.Normal,
14         fontSize = 16.sp,
15         lineHeight = 24.sp,
16         letterSpacing = 0.5.sp
17     )
18     /* Other default text styles to override
19     titleLarge = TextStyle(
20         fontFamily = FontFamily.Default,
21         fontWeight = FontWeight.Normal,
22         fontSize = 22.sp,
23         lineHeight = 28.sp,
24         letterSpacing = 0.sp
25     ),
26     labelSmall = TextStyle(
27         fontFamily = FontFamily.Default,
28         fontWeight = FontWeight.Medium,
29         fontSize = 11.sp,
30         lineHeight = 16.sp,
31         letterSpacing = 0.5.sp
32     )
)

```

```
33 */  
34 )
```

```
1 package com.example.compose
2 import androidx.compose.ui.graphics.Color
3
4 val md_theme_light_primary = Color(0xFFFFB12B22)
5 val md_theme_light_onPrimary = Color(0xFFFFFFFF)
6 val md_theme_light_primaryContainer = Color(0xFFFDAD5)
7 val md_theme_light_onPrimaryContainer = Color(0xFF410001)
8 val md_theme_light_secondary = Color(0xFF9A4058)
9 val md_theme_light_onSecondary = Color(0xFFFFFFFF)
10 val md_theme_light_secondaryContainer = Color(0xFFFFD9DF)
11 val md_theme_light_onSecondaryContainer = Color(0xFF3F0017)
12 val md_theme_light_tertiary = Color(0xFF9C413F)
13 val md_theme_light_onTertiary = Color(0xFFFFFFFF)
14 val md_theme_light_tertiaryContainer = Color(0xFFFDAD7)
15 val md_theme_light_onTertiaryContainer = Color(0xFF410005)
16 val md_theme_light_error = Color(0xFFBA1A1A)
17 val md_theme_light_errorContainer = Color(0xFFFFFDAD6)
18 val md_theme_light_onError = Color(0xFFFFFFFF)
19 val md_theme_light_onErrorContainer = Color(0xFF410002)
20 val md_theme_light_background = Color(0xFFFBFFF)
21 val md_theme_light_onBackground = Color(0xFF201A19)
22 val md_theme_light_surface = Color(0xFFFFFBFF)
23 val md_theme_light_onSurface = Color(0xFF201A19)
24 val md_theme_light_surfaceVariant = Color(0xFFFF5DDA)
25 val md_theme_light_onSurfaceVariant = Color(0xFF534341)
26 val md_theme_light_outline = Color(0xFF857370)
27 val md_theme_light_inverseOnSurface = Color(0xFFFFBEEC)
28 val md_theme_light_inverseSurface = Color(0xFF362F2E)
29 val md_theme_light_inversePrimary = Color(0xFFFFB4A9)
30 val md_theme_light_shadow = Color(0xFF000000)
31 val md_theme_light_surfaceTint = Color(0xFFB12B22)
32 val md_theme_light_outlineVariant = Color(0xFFD8C2BE)
```

```

33 val md_theme_light_scrim = Color(0xFF000000)
34
35 val md_theme_dark_primary = Color(0xFFFFFB4A9)
36 val md_theme_dark_onPrimary = Color(0xFF690002)
37 val md_theme_dark_primaryContainer = Color(0xFF8F100D)
38 val md_theme_dark_onPrimaryContainer = Color(0xFFFFFDAD5)
39 val md_theme_dark_secondary = Color(0xFFFFB1C1)
40 val md_theme_dark_onSecondary = Color(0xFF5F112B)
41 val md_theme_dark_secondaryContainer = Color(0xFF7C2941)
42 val md_theme_dark_onSecondaryContainer = Color(0xFFFFFD9DF)
43 val md_theme_dark_tertiary = Color(0xFFFFFB3AF)
44 val md_theme_dark_onTertiary = Color(0xFF5F1416)
45 val md_theme_dark_tertiaryContainer = Color(0xFF7E2A2A)
46 val md_theme_dark_onTertiaryContainer = Color(0xFFFFDAD7)
47 val md_theme_dark_error = Color(0xFFFFFB4AB)
48 val md_theme_dark_errorContainer = Color(0xFF93000A)
49 val md_theme_dark_onError = Color(0xFF690005)
50 val md_theme_dark_onErrorContainer = Color(0xFFFFFDAD6)
51 val md_theme_dark_background = Color(0xFF201A19)
52 val md_theme_dark_onBackground = Color(0xFFEDE0DE)
53 val md_theme_dark_surface = Color(0xFF201A19)
54 val md_theme_dark_onSurface = Color(0xFFEDE0DE)
55 val md_theme_dark_surfaceVariant = Color(0xFF534341)
56 val md_theme_dark_onSurfaceVariant = Color(0xFFFFD8C2BE)
57 val md_theme_dark_outline = Color(0xFFA08C89)
58 val md_theme_dark_inverseOnSurface = Color(0xFF201A19)
59 val md_theme_dark_inverseSurface = Color(0xFFEDE0DE)
60 val md_theme_dark_inversePrimary = Color(0xFFFFB12B22)
61 val md_theme_dark_shadow = Color(0xFF000000)
62 val md_theme_dark_surfaceTint = Color(0xFFFFFB4A9)
63 val md_theme_dark_outlineVariant = Color(0xFF534341)
64 val md_theme_dark_scrim = Color(0xFF000000)

```

```
65  
66  
67 val seed = Color(0xFF880808)  
68
```

```
1 package com.example.compose
2
3 import androidx.compose.foundation.isSystemInDarkTheme
4 import androidx.compose.material3.MaterialTheme
5 import androidx.compose.material3.lightColorScheme
6 import androidx.compose.material3.darkColorScheme
7 import androidx.compose.runtime.Composable
8
9
10 private val LightColors = lightColorScheme(
11     primary = md_theme_light_primary,
12     onPrimary = md_theme_light_onPrimary,
13     primaryContainer = md_theme_light_primaryContainer,
14     onPrimaryContainer = md_theme_light_onPrimaryContainer,
15     secondary = md_theme_light_secondary,
16     onSecondary = md_theme_light_onSecondary,
17     secondaryContainer = md_theme_light_secondaryContainer,
18     onSecondaryContainer = md_theme_light_onSecondaryContainer,
19     tertiary = md_theme_light_tertiary,
20     onTertiary = md_theme_light_onTertiary,
21     tertiaryContainer = md_theme_light_tertiaryContainer,
22     onTertiaryContainer = md_theme_light_onTertiaryContainer,
23     error = md_theme_light_error,
24     errorContainer = md_theme_light_errorContainer,
25     onError = md_theme_light_onError,
26     onErrorContainer = md_theme_light_onErrorContainer,
27     background = md_theme_light_background,
28     onBackground = md_theme_light_onBackground,
29     surface = md_theme_light_surface,
30     onSurface = md_theme_light_onSurface,
31     surfaceVariant = md_theme_light_surfaceVariant,
32     onSurfaceVariant = md_theme_light_onSurfaceVariant,
```

```

File - C:\Users\Akhilesh\Downloads\My Youtube App 20BCE7010\My Youtube App 20BCE7010\app\src\main\java\com\example\MyYoutubeApp_20BCE7010\ui\theme\TTheme.kt

33 outline = md_theme_light_outline,
34 inverseOnSurface = md_theme_light_inverseOnSurface,
35 inverseSurface = md_theme_light_inverseSurface,
36 inversePrimary = md_theme_light_inversePrimary,
37 surfaceTint = md_theme_light_surfaceTint,
38 outlineVariant = md_theme_light_outlineVariant,
39 scrim = md_theme_light_scrim,
40 )
41
42 private val DarkColors = darkColorScheme(
43     primary = md_theme_dark_primary,
44     onPrimary = md_theme_dark_onPrimary,
45     primaryContainer = md_theme_dark_primaryContainer,
46     onPrimaryContainer = md_theme_dark_onPrimaryContainer,
47     secondary = md_theme_dark_secondary,
48     onSecondary = md_theme_dark_onSecondary,
49     secondaryContainer = md_theme_dark_secondaryContainer,
50     onSecondaryContainer = md_theme_dark_onSecondaryContainer,
51     tertiary = md_theme_dark_tertiary,
52     onTertiary = md_theme_dark_onTertiary,
53     tertiaryContainer = md_theme_dark_tertiaryContainer,
54     onTertiaryContainer = md_theme_dark_onTertiaryContainer,
55     error = md_theme_dark_error,
56     errorContainer = md_theme_dark_errorContainer,
57     onError = md_theme_dark_onError,
58     onErrorContainer = md_theme_dark_onErrorContainer,
59     background = md_theme_dark_background,
60     onBackground = md_theme_dark_onBackground,
61     surface = md_theme_dark_surface,
62     onSurface = md_theme_dark_onSurface,
63     surfaceVariant = md_theme_dark_surfaceVariant,
64

```

```
65     onSurfaceVariant = md_theme_dark_onSurfaceVariant,  
66     outline = md_theme_dark_outline,  
67     inverseOnSurface = md_theme_dark_inverseOnSurface,  
68     inverseSurface = md_theme_dark_inverseSurface,  
69     inversePrimary = md_theme_dark_inversePrimary,  
70     surfaceTint = md_theme_dark_surfaceTint,  
71     outlineVariant = md_theme_dark_outlineVariant,  
72     scrim = md_theme_dark_scrim,  
73 )  
74  
75 @Composable  
76 fun AppTheme(  
77     useDarkTheme: Boolean = isSystemInDarkTheme(),  
78     content: @Composable() () -> Unit  
79 ) {  
80     val colors = if (!useDarkTheme) {  
81         LightColors  
82     } else {  
83         DarkColors  
84     }  
85     MaterialTheme(  
86         colorScheme = colors,  
87         content = content  
88     )  
89 }  
90 }
```