```kotlin
package com.example.youchube

import android.app.Activity
import android.content.Context
import android.content.ContextWrapper
import android.content.pm.ActivityInfo
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.isSystemInDarkTheme
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.Button
import androidx.compose.material3.Card
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Icon
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.OutlinedTextField
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.DisposableEffect
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import androidx.compose.ui.viewinterop.AndroidView
import androidx.constraintlayout.compose.ChainStyle
import androidx.constraintlayout.compose.ConstraintLayout
import androidx.navigation.NavHostController
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import com.example.compose.AppTheme
import com.google.accompanist.systemuicontroller.rememberSystemUiController
import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.YouTubePlayer
import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.listeners.AbstractYouTubePlayerListener
import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.views.YouTubePlayerView
import java.util.regex.Pattern

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            AppTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    Navi()
                    Hide()
                }
            }
        }
    }
}

@Composable
private fun Hide() {
    val systemUiController = rememberSystemUiController()
    if (isSystemInDarkTheme()) {
        systemUiController.setSystemBarsColor(
            color = MaterialTheme.colorScheme.primaryContainer
        )
    } else {
        systemUiController.setSystemBarsColor(
            color = Color.White
        )
    }
}
```

```kotlin

@Composable
fun Navi() {
    val navigator = rememberNavController()
    NavHost(navController = navigator, startDestination = "Body") {
        composable(route = "Body") {
            Body(Navigator = navigator)
        }
        composable(route = "Video/{VId}") {
            val video = it.arguments?.getString("VId")
            if (video != null) {
                YoutubeScreen(videoId = video)
            }
        }
    }
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun Body(Navigator: NavHostController) {
    Card(
        modifier = Modifier
            .fillMaxSize()
            .padding(20.dp)
    ) {
        ConstraintLayout(modifier = Modifier.fillMaxSize()) {
            val (logo, url, button) = createRefs()
            val vChain = createVerticalChain(logo, url, button, chainStyle = ChainStyle.Packed)
            var URL by remember {
                mutableStateOf("")
            }
            var id: String? by remember {
                mutableStateOf("")
            }

            if (isSystemInDarkTheme()) {
                Image(painter = painterResource(id = R.drawable.yt_logo_rgb_dark),
                    contentDescription = "",
                    modifier = Modifier
                        .size(200.dp)
                        .constrainAs(logo) {
                            start.linkTo(parent.start)
                            end.linkTo(parent.end)
                        })
            } else {
                Image(painter = painterResource(id = R.drawable.yt_logo_rgb_light),
                    contentDescription = "",
                    modifier = Modifier
                        .size(200.dp)
                        .constrainAs(logo) {
                            start.linkTo(parent.start)
                            end.linkTo(parent.end)
                        })
            }

            OutlinedTextField(
                value = URL,
                onValueChange = { URL = it },
                label = { Text(text = "Video URL") },
                modifier = Modifier
                    .padding(top = 100.dp)
                    .constrainAs(url) {
                        start.linkTo(parent.start)
                        end.linkTo(parent.end)
                    })

            Button(onClick = {
                id = urlParser(URL)
                Navigator.navigate(route = "Video/$id")
            },
                shape = RoundedCornerShape(5.dp),
                modifier = Modifier
                    .padding(30.dp)
                    .constrainAs(button) {
                        start.linkTo(parent.start)
                        end.linkTo(parent.end)
                        bottom.linkTo(parent.bottom, margin = 600.dp)
                    }) {
                Text(text = "Play")
                Icon(
```

```kotlin
                    painter = painterResource(id = R.drawable.
baseline_play_arrow_24),
                    contentDescription = ""
                )
            }
        }
    }
}

@Composable
fun YoutubeScreen(
    videoId: String
) {
    AndroidView(factory = {
        val view = YouTubePlayerView(it)
        val fragment = view.addYouTubePlayerListener(
            object : AbstractYouTubePlayerListener() {
                override fun onReady(youTubePlayer: YouTubePlayer) {
                    super.onReady(youTubePlayer)
                    youTubePlayer.loadVideo(videoId, 0f)
                }
            }
        )
        view
    }, modifier = Modifier.padding(75.dp))
    ScreenOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE)
}
```

```kotlin

@Composable
fun ScreenOrientation(orientation: Int) {
    val context = LocalContext.current
    DisposableEffect(orientation) {
        val activity = context.findActivity() ?: return@DisposableEffect onDispose
 {}
        val originalOrientation = activity.requestedOrientation
        activity.requestedOrientation = orientation
        onDispose {
            // restore original orientation when view disappears
            activity.requestedOrientation = originalOrientation
        }
    }
}

fun urlParser(url: String): String? {
    var vId: String? = null
    val pattern = Pattern.compile(
        "^https?://.*(?:youtu.be/|v/|u/\\w/|embed/|watch?v=)([^#&?]*).*$",
        Pattern.CASE_INSENSITIVE
    )
    val matcher = pattern.matcher(url)
    if (matcher.matches()) {
        vId = matcher.group(1)
    }
    return vId
}

fun Context.findActivity(): Activity? = when (this) {
    is Activity -> this
    is ContextWrapper -> baseContext.findActivity()
    else -> null
}
```