

Main Code:-

```
package com.zach.vit_20BCE2381_youtube

import android.os.Bundle

import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.layout.size
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.*
import androidx.compose.ui.Modifier
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.unit.dp
import androidx.compose.ui.viewinterop.AndroidView
import coil.annotation.ExperimentalCoilApi
import coil.compose.rememberImagePainter
import com.google.api.client.googleapis.json.GoogleJsonResponseException
import com.google.api.client.http.javanet.NetHttpTransport
import com.google.api.client.json.gson.GsonFactory
import com.google.api.services.youtube.YouTube
import com.google.api.services.youtube.model.SearchListResponse
import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.YouTubePlayer
import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.listeners.AbstractYouTubePlayerListener
```

```
import com.pierfrancescosoffritti.androidyoutubeplayer.core.player.views.YouTubePlayerView

import com.zach.vit_20bce1690_youtube.ui.theme.VIT_20BCE1690_YOUTUBETheme

import kotlinx.coroutines.Dispatchers

import kotlinx.coroutines.withContext
```

```
class MainActivity : ComponentActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContent {
            VIT_20BCE1690_YOUTUBETheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colorScheme.background
                ) {
                    VideoList()
                }
            }
        }
    }
}

@Composable
fun YoutubeScreen(
    videoid: String,
    modifier: Modifier
) {
    val ctx = LocalContext.current

    AndroidView(factory = {
        var view = YouTubePlayerView(it)

        val fragment = view.addYouTubePlayerListener(
            object : AbstractYouTubePlayerListener() {
                override fun onReady(youTubePlayer: YouTubePlayer) {
```

```

        super.onReady(youTubePlayer)

        youTubePlayer.loadVideo(videoid, 0f)
    }
}
)
view
})
}

```

```

data class VideoItem(
    val videoid: String,
    val title: String,
    val description: String,
    val thumbnailUrl: String
)

```

@Composable

```

fun VideoList() {
    var videos by remember { mutableStateOf(emptyList<VideoItem>()) }

    LaunchedEffect(Unit) {
        fetchVideos { fetchedVideos ->
            videos = fetchedVideos
        }
    }

    Column(modifier.fillMaxSize()) {
        LazyColumn {
            items(videos) { video ->
                YoutubeScreen(video.videoid, Modifier.fillMaxWidth().height(300.dp))
            }
        }
    }
}

```

```

    }
}
}

```

```

private suspend fun fetchVideos(onVideosFetched: (List<VideoItem>) -> Unit) {
    withContext(Dispatchers.IO) {
        try {
            val youtube = YouTube.Builder(
                NetHttpTransport(),
                GsonFactory.getDefaultInstance(),
                null
            ).setApplicationName("Your Application Name").build()

            val searchListResponse: SearchListResponse = youtube.search().list("snippet").apply {
                key = ""//Your Api key
                type = "video"
                maxResults = 5
                channelId = "UCRcgy6GzDeccl7dkbbBna3Q"
                order = "date"
            }.execute()

            val videos = searchListResponse.items.map { video ->
                val videoId = video.id.videoId
                val title = video.snippet.title
                val description = video.snippet.description
                val thumbnailUrl = video.snippet.thumbnails.default.url
                VideoItem(videoId, title, description, thumbnailUrl)
            }

            // Invoke the callback with the fetched videos
            onVideosFetched(videos)
        } catch (e: GoogleJsonResponseException) {

```

```
        // Handle API request errors
        e.printStackTrace()
        onVideosFetched(emptyList())
    }
}
}
```

Gradle code:-

```
implementation 'com.google.api-client:google-api-client-android:1.32.1'
implementation 'com.google.apis:google-api-services-youtube:v3-rev206-1.25.0'
implementation 'com.google.api-client:google-api-client-gson:1.32.1'
implementation "io.coil-kt:coil-compose:1.3.2"
implementation "com.pierfrancescosoffritti.androidyoutubeplayer:core:11.1.0"
```