Assignment – 3

Name: MD SHABREZ

Register Number: 20BCE1690

YouTube App

MainActivity.kt

```
package com.zach.vit_20bce1690_youtube


import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.ui.Modifier
import com.zach.vit_20bce1690_youtube.ui.theme.VIT_20BCE1690_YOUTUBETheme


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
                ) {
                    ActivityNavigation()
                }
            }
        }
```

```
    }
}
```

ActivityNavigation.kt

```
package com.zach.vit_20bce1690_youtube

import androidx.compose.foundation.Image
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Arrangement
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.ExperimentalLayoutApi
import androidx.compose.foundation.layout.PaddingValues
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.consumedWindowInsets
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.Divider
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Text
import androidx.compose.material3.TopAppBar
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.runtime.getValue
import androidx.compose.runtime.mutableStateOf
import androidx.compose.runtime.remember
import androidx.compose.runtime.setValue
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.drawBehind
import androidx.compose.ui.geometry.Offset
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
```

```
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.compose.ui.viewinterop.AndroidView
import androidx.navigation.NavController
import androidx.navigation.NavType
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.navArgument
import androidx.navigation.compose.rememberNavController
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.Abst
ractYouTubePlayerListener
import
com.pierfrancescosoffritti.androidyoutubeplayer.core.player.views.YouTubeP
layerView
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.withContext

@Preview
@Composable
fun ActivityNavigation() {
    val navController = rememberNavController()
    NavHost(navController = navController, startDestination =
Activities.Home.route) {

        composable(Activities.Home.route) {
            Home(navController = navController)
        }
```

```
        composable(
            route = Activities.VideoScreen.route +
                    "/{title}/{views}/{videoId}/{desc}/{likes}",
            arguments = listOf(
                navArgument(name = "title") {
                    type = NavType.StringType
                },
                navArgument(name = "views") {
                    type = NavType.StringType
                },
                navArgument(name = "videoId") {
                    type = NavType.StringType
                },
                navArgument(name = "desc") {
                    type = NavType.StringType
                },
                navArgument(name = "likes") {
                    type = NavType.StringType
                },
            )
        ) { entry ->
            Column(
                Modifier.fillMaxSize()
            ) {
                val title = entry.arguments?.getString("title").toString()
                val videoId =
entry.arguments?.getString("videoId").toString()
                val views = entry.arguments?.getString("views").toString()
                val likes = entry.arguments?.getString("likes").toString()
                val desc =
entry.arguments?.getString("desc").toString().replace("·", "/")

                VideoScreen(title, videoId, views, likes, desc)

            }


        }
    }
}

@Composable
```

```
fun VideoScreen(title: String, videoId: String, views: String, likes:
String, desc: String) {
    Column(
        Modifier
            .fillMaxSize(), horizontalAlignment =
Alignment.CenterHorizontally
    ) {
        YoutubeScreen(
            videoId = videoId, modifier = Modifier
                .fillMaxWidth()
                .height(300.dp)
        )
        Column(Modifier.padding(16.dp)) {
            Text(text = "Description", fontSize = 26.sp, fontWeight =
FontWeight.Bold)
            Spacer(modifier = Modifier.height(20.dp))
            Divider(color = Color.LightGray, thickness = 1.dp)
            Spacer(modifier = Modifier.height(20.dp))
            Text(text = title, fontSize = 26.sp, fontWeight =
FontWeight.Bold)
            Spacer(modifier = Modifier.height(20.dp))
            Row(
                horizontalArrangement = Arrangement.SpaceAround,
                modifier = Modifier.fillMaxWidth()
            ) {
                Column(
                    verticalArrangement = Arrangement.Center,
                    horizontalAlignment = Alignment.CenterHorizontally
                ) {
                    Text(text = likes, fontSize = 26.sp, fontWeight =
FontWeight.Bold)
                    Text(
                        text = "Likes", color = Color(
                            134,
                            134,
                            134,
                            255
                        )
                    )
                }

                Column(
```

```
                    verticalArrangement = Arrangement.Center,
                    horizontalAlignment = Alignment.CenterHorizontally
                ) {
                    Text(text = videoId, fontSize = 26.sp, fontWeight =
FontWeight.Bold)
                    Text(
                        text = "Video ID", color = Color(
                            134,
                            134,
                            134,
                            255
                        )
                    )
                }
                Column(
                    verticalArrangement = Arrangement.Center,
                    horizontalAlignment = Alignment.CenterHorizontally
                ) {
                    Text(text = views, fontSize = 26.sp, fontWeight =
FontWeight.Bold)
                    Text(
                        text = "Views", color = Color(
                            134,
                            134,
                            134,
                            255
                        )
                    )
                }
            }
            Spacer(modifier = Modifier.height(20.dp))
            Divider(color = Color.LightGray, thickness = 1.dp)
            Spacer(modifier = Modifier.height(20.dp))
            Text(
                text = desc, color = Color(
                    134,
                    134,
                    134,
                    255
                ), fontSize = 18.sp
            )
        }
```

```
    }
}

@Composable
fun Home(navController: NavController) {
    CustomScaffold(navController)
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun CustomScaffold(navController: NavController) {
    LocalContext.current
    Scaffold(topBar = { CustomTopBar() },
        content = { pad -> MainContent(pad, navController) }
    )
}

@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun CustomTopBar() {
    TopAppBar(
        title = {
            Row(
                verticalAlignment = Alignment.CenterVertically,
horizontalArrangement =
                Arrangement.Center, modifier = Modifier
                    .fillMaxWidth()
                    .padding(
                        end = 50
                            .dp
                    )
            ) {

                Image(
                    painterResource(id =
R.drawable.youtubelogotransparent),
                    contentDescription = "",
                    modifier = Modifier.size(110.dp)
                )


            }
```

```
        },
        modifier = Modifier.drawBehind {
            drawLine(
                Color.LightGray,
                Offset(0f, size.height),
                Offset(size.width, size.height),
                5f
            )
        }
    )
}

@OptIn(ExperimentalLayoutApi::class, ExperimentalMaterial3Api::class)
@Composable
fun MainContent(padding: PaddingValues, navController: NavController) {


    Column(
    ) {
        Column(
            modifier = Modifier
                .padding(padding)
                .consumedWindowInsets(padding),
        ) {
            VideoList(navController)
        }
    }
}


@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,
    val views: Int,
    val uploadDate: String,
    val likes: Int
)

private fun formatCount(count: Int): String {
    return when {
        count < 1000 -> count.toString()
        count < 1000000 -> "${(count / 1000)}K"
        else -> {
            val decimal = (count % 1000000) / 100000
            "${(count / 1000000)}.${decimal}M"
        }
    }
}


@OptIn(ExperimentalCoilApi::class)
@Composable
fun VideoItem(video: VideoItem, navController: NavController) {
    Column(
        Modifier
            .fillMaxSize()
            .clickable {
                navController.navigate(
                    Activities.VideoScreen.withArgs(
                        video.title,
                        formatCount(video.views),
                        video.videoId,
```

```
                        video.description.replace("/", "\u00B7"),
                        formatCount(video.likes),
                    )
                )
            }) {
        Image(
            painter = rememberImagePainter(video.thumbnailUrl),
            contentDescription = "",
            modifier = Modifier
                .fillMaxSize()
                .size(200.dp),
            contentScale = ContentScale.Crop
        )
        Column(Modifier.padding(16.dp, 10.dp, 16.dp, 35.dp)) {
            Text(text = video.title, fontSize = 19.sp, fontWeight =
FontWeight.SemiBold)
            Row() {
                Text(
                    text = "${formatCount(video.views)} views \u00B7 ",
color = Color(
                        134,
                        134,
                        134,
                        255
                    )
                )
                Text(
                    text = "${formatCount(video.likes)} likes", color =
Color(
                        134,
                        134,
                        134,
                        255
                    )
                )
            }

        }
    }
}

@Composable
```

```
fun VideoList(navController: NavController) {
    var videos by remember { mutableStateOf(emptyList<VideoItem>()) }

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

    Column(Modifier.fillMaxSize()) {
        LazyColumn {
            items(videos) { video ->
                VideoItem(video = video, navController)
            }
        }
    }
}

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 = "AIzaSyBpjOUPicEHpsQ_8XcBL3QIbA7AMWg_dQ4"
                    type = "video"
                    maxResults = 10 // Number of videos to fetch
                    channelId = "UCRcgy6GzDeccI7dkbbBna3Q"
                    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.high.url
```

```
                val views = getVideoViews(videoId, youtube)
                val uploadDate = video.snippet.publishedAt.toString()
                val likes = getVideoLikes(videoId, youtube)
                VideoItem(
                    videoId,
                    title,
                    description,
                    thumbnailUrl,
                    views,
                    uploadDate,
                    likes
                )
            }

            // Invoke the callback with the fetched videos
            onVideosFetched(videos)
        } catch (e: GoogleJsonResponseException) {
            // Handle API request errors
            e.printStackTrace()
            onVideosFetched(emptyList())
        }
    }
}

private suspend fun getVideoViews(videoId: String, youtube: YouTube): Int
{
    val videoListResponse = youtube.videos().list("statistics").apply {
        key = "AIzaSyBpjOUPicEHpsQ_8XcBL3QIbA7AMWg_dQ4"
        id = videoId
    }.execute()

    val video = videoListResponse.items.firstOrNull()
    return video?.statistics?.viewCount?.toInt() ?: 0
}

private suspend fun getVideoLikes(videoId: String, youtube: YouTube): Int
{
    val videoListResponse = youtube.videos().list("statistics").apply {
        key = "AIzaSyBpjOUPicEHpsQ_8XcBL3QIbA7AMWg_dQ4"
        id = videoId
    }.execute()
```

```
    val video = videoListResponse.items.firstOrNull()
    return video?.statistics?.likeCount?.toInt() ?: 0
}
```

Activities.kt

```
package com.zach.vit_20bce1690_youtube

sealed class Activities(val route: String) {
    object Home : Activities("home")
    object VideoScreen : Activities("video_screen")

    fun withArgs(vararg args: String): String {
        return buildString {
            append(route)
            args.forEach { arg ->
                append("/$arg")
            }
        }
    }
}
```

AndroidManifest.xml (Add the below lines):

```
<uses-permission android:name="android.permission.INTERNET" />
```

build.gradle (Add the below lines):

```
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"
implementation "androidx.navigation:navigation-compose:2.4.0-alpha04"
```