

WEEK-3

Vishesh.singhal2020@vitstudent.ac.in

detailScreen.kt:-

```
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
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.unit.dp
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import com.example.youtubeplayer.YoutubeScreen

@Composable
fun DetailScreen(navController: NavController, videoId:String?,
videoT:String?, videoD:String?) {

    var title by remember{
        mutableStateOf("")
    }
    var desc by remember{
        mutableStateOf("")
    }
    if(videoT!=null){
        title=videoT.toString()
    }
    if(videoT!=null){
        if (videoD != null) {
            desc=videoD.toString()
        }
    }
    if (videoId != null) {
        Column() {
            YoutubeScreen(videoId = videoId, modifier =
Modifier.fillMaxWidth().height(300.dp) )
            Text(title, fontSize=32.sp, modifier = Modifier.padding(16.dp))
            Text(desc, fontSize = 20.sp, modifier =
Modifier.padding(16.dp).background(color =
Color.LightGray))
        }
    }
}
```

MainActivity.kt

```
package com.example.youtubeplayer

import Navigation
import android.annotation.SuppressLint
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.runtime.*
import androidx.compose.ui.platform.ComposeView
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.ScrollState
import androidx.compose.foundation.background
import androidx.compose.foundation.clickable
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
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.layout.width
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material3.Card
import androidx.compose.material3.CardDefaults
import androidx.compose.material3.ExperimentalMaterial3Api
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Scaffold
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.runtime.LaunchedEffect
import androidx.compose.runtime.R
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.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.sp
import androidx.navigation.NavController
import coil.compose.rememberAsyncImagePainter
import com.example.youtubeplayer.ui.theme.YouTubePlayerTheme
```

```

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 kotlinx.coroutines.withContext

import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView {
            YoutubePlayerTheme {
                // A surface container using the 'background' color from
the theme
                Navigation()
            }
        }
    }
}
@OptIn(ExperimentalMaterial3Api::class)
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")

@Composable
fun TopBarsBuilt() {
    Column(modifier = Modifier
        .fillMaxWidth().height(80.dp).background(color = Color.Magenta)) {
        Text(text = "VisheshSinghal Fake YT", color = Color.White, fontSize =
32.sp, modifier =
        Modifier.padding(16.dp))
    }
}
@SuppressLint("UnusedMaterial3ScaffoldPaddingParameter")
@OptIn(ExperimentalMaterial3Api::class)
@Composable
fun MainScreen(navController: NavController) {
    Scaffold(topBar = {TopBarsBuilt()}, content = {
        Surface(
            modifier = Modifier.fillMaxSize(),
            color = MaterialTheme.colorScheme.background
        ) {
            VideoList(navController)
        }
    }) ;
}

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

@Composable
fun VideoItem(navController: NavController, video: VideoItem) {

    Card(elevation = CardDefaults.cardElevation(defaultElevation =

```

```

10.dp), modifier =
    Modifier
        .padding(10.dp)
        .clickable {
            navController.navigate(
Screen.detailScreen.withArgs(video.videoId, video.title, video.description)
        )
    }) {
    Row(Modifier.fillMaxWidth()) {

        Image(
            painter = rememberAsyncImagePainter(video.thumbnailUrl),
            contentDescription = "",
            modifier = Modifier.width(100.dp).height(100.dp),
            contentScale = ContentScale.Crop
        )
        Column(Modifier.padding(16.dp)) {
            Text(text = video.title)
//            Text(text = video.description)
        }
    }
}

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

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

    Column(
        Modifier
            .fillMaxSize()
            .verticalScroll(rememberScrollState()) {
        LazyColumn(modifier=Modifier.size(800.dp)) {
            items(videos) { video ->
                VideoItem(navController, video)
            }
        }
    }
}

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 = "AIzaSyCEJ3EJ09YWAYDkxt2OGFnwagvj1HHaDk0"
        type = "video"
        maxResults = 30 // Number of videos to fetch
        channelId = "UC8butISFwT-Wl7EV0hUK0BQ"
    }.execute()

    val videos = searchListResponse.items.map { video ->
        val title = video.snippet.title
        val description = video.snippet.description
        val thumbnailUrl = video.snippet.thumbnails.default.url
        val videoId = video.id.videoId
    //      val videoUrl = "https://www.youtube.com/watch?v=$videoId"
        VideoItem(title, description, thumbnailUrl, videoId)
    }

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

}

// Invoke the}

```

Navigation.kt

```

import androidx.compose.foundation.layout.Column
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
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.unit.dp
import androidx.compose.ui.unit.sp

import androidx.navigation.NavController
import androidx.navigation.NavHostController
import androidx.navigation.NavType
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable

import androidx.navigation.compose.rememberNavController
import androidx.navigation.navArgument
import com.example.youtubeplayer.MainScreen

import com.example.youtubeplayer.Screen
import com.example.youtubeplayer.YoutubeScreen

@Composable
fun Navigation() {
    val navController= rememberNavController()

```

```

NavHost(navController = navController, startDestination =
Screen.mainScreen.route ) {
    composable(route=Screen.mainScreen.route) {
        MainScreen(navController = navController)
    }
    composable(route=Screen.detailScreen
        .route+"/{videoId}/{videoTitle}/{videoDesc}",
        arguments = listOf(
            navArgument("videoId") {
                type= NavType.StringType
                defaultValue=" "
            },  navArgument("videoTitle") {
                type= NavType.StringType
                defaultValue=" "
            },  navArgument("videoDesc") {
                type= NavType.StringType
                defaultValue=" "
            },
        )
    ) {entry->
        DetailScreen(navController,
            videoId= entry.arguments?.getString("videoId"),
            videoT= entry.arguments?.getString("videoTitle"),
            videoD= entry.arguments?.getString("videoDesc"),
        )
    }
}
}
}

```

Screen.kt

```

package com.example.youtubeplayer

sealed class Screen(val route:String){

    object mainScreen : Screen("main_Screen")
    object detailScreen : Screen("detail_Screen")
    fun withArgs(vararg args:String):String{
        return buildString {
            append(route)
            args.forEach{
                arg->
                append("/$arg")
            }
        }
    }
}

```

videoPlaying.kt

```

package com.example.youtubeplayer

import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.platform.LocalContext
import androidx.compose.ui.viewinterop.AndroidView
import

```

```
com.pierfrancescosoffritti.androidyoutubeplayer.core.player.YouTubePlayer
import
com.pierfrancescosoffritti.androidyoutubeplayer.core.player.listeners.Abstr
actYouTubePlayerListener
import
com.pierfrancescosoffritti.androidyoutubeplayer.core.player.views.YouTubePl
ayerView

@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
    })
}
```