NEWS ARTICLE APPLICATION

//MainActivity.kt

package com.example.newsarticle  
  
import android.os.Bundle  
import androidx.activity.enableEdgeToEdge  
import androidx.appcompat.app.AppCompatActivity  
import androidx.core.view.ViewCompat  
import androidx.core.view.WindowInsetsCompat  
import android.content.Context  
import android.util.Log  
import android.view.View  
import android.widget.EditText  
import android.widget.ImageButton  
import android.widget.ProgressBar  
import android.widget.Toast  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import retrofit2.Call  
import retrofit2.Callback  
import retrofit2.Response  
  
class MainActivity : AppCompatActivity() {  
  
 private lateinit var recyclerView: RecyclerView  
 private lateinit var progressBar: ProgressBar  
 private lateinit var searchEditText: EditText  
 private lateinit var searchButton: ImageButton  
 private lateinit var adapter: NewsAdapter  
 private val articles = *mutableListOf*<Article>()  
 private val likedArticles = *mutableSetOf*<String>()  
 private val sharedPref by *lazy* **{** getPreferences(Context.*MODE\_PRIVATE*) **}** override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main)  
  
 recyclerView = findViewById(R.id.recyclerView)  
 progressBar = findViewById(R.id.progressBar)  
 searchEditText = findViewById(R.id.searchEditText)  
 searchButton = findViewById(R.id.searchButton)  
  
 // Load liked articles from SharedPreferences  
 likedArticles.addAll(sharedPref.getStringSet("liked\_articles", *setOf*()) ?: *setOf*())  
  
 setupRecyclerView()  
 loadTopHeadlines()  
  
 searchButton.setOnClickListener **{** val query = searchEditText.*text*.toString()  
 if (query.*isNotEmpty*()) {  
 searchNews(query)  
 } else {  
 loadTopHeadlines()  
 }  
 **}** }  
  
 private fun setupRecyclerView() {  
 adapter = NewsAdapter(  
 this,  
 articles,  
 onLikeClickListener = **{** article **->** if (likedArticles.contains(article.url)) {  
 likedArticles.remove(article.url)  
 Toast.makeText(this, "Removed from favorites", Toast.*LENGTH\_SHORT*).show()  
 } else {  
 likedArticles.add(article.url)  
 Toast.makeText(this, "Added to favorites", Toast.*LENGTH\_SHORT*).show()  
 }  
 // Save to SharedPreferences  
 sharedPref.edit().putStringSet("liked\_articles", likedArticles).apply()  
 adapter.notifyDataSetChanged()  
 **}**,  
 likedArticles  
 )  
  
 recyclerView.layoutManager = LinearLayoutManager(this)  
 recyclerView.adapter = adapter  
 }  
  
 private fun loadTopHeadlines() {  
 progressBar.*visibility* = View.*VISIBLE* RetrofitClient.instance.getTopHeadlines("us", "6b4958f940344ca497fb183e1a384d42").enqueue(object : Callback<NewsResponse> {  
 override fun onResponse(call: Call<NewsResponse>, response: Response<NewsResponse>) {  
 progressBar.*visibility* = View.*GONE* Log.d("API\_ERROR", "Response code: ${response.code()}, error: ${response.errorBody()?.string()}")  
 if (response.*isSuccessful*) {  
 response.body()?.articles?.*let* **{** articles.clear()  
 articles.addAll(**it**)  
 adapter.notifyDataSetChanged()  
 **}** } else {  
 Toast.makeText(this@MainActivity, "Failed to load news", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
  
 override fun onFailure(call: Call<NewsResponse>, t: Throwable) {  
 progressBar.*visibility* = View.*GONE* Toast.makeText(this@MainActivity, "Error: ${t.message}", Toast.*LENGTH\_SHORT*).show()  
 }  
 })  
 }  
  
 private fun searchNews(query: String) {  
 progressBar.*visibility* = View.*VISIBLE* RetrofitClient.instance.searchNews(query, "6b4958f940344ca497fb183e1a384d42").enqueue(object : Callback<NewsResponse> {  
 override fun onResponse(call: Call<NewsResponse>, response: Response<NewsResponse>) {  
 progressBar.*visibility* = View.*GONE* if (response.*isSuccessful*) {  
 response.body()?.articles?.*let* **{** articles.clear()  
 articles.addAll(**it**)  
 adapter.notifyDataSetChanged()  
 **}** } else {  
 Toast.makeText(this@MainActivity, "Failed to search news", Toast.*LENGTH\_SHORT*).show()  
 }  
 }  
  
 override fun onFailure(call: Call<NewsResponse>, t: Throwable) {  
 progressBar.*visibility* = View.*GONE* Toast.makeText(this@MainActivity, "Error: ${t.message}", Toast.*LENGTH\_SHORT*).show()  
 }  
 })  
 }

//NewsAdapter.kt

package com.example.newsarticle  
  
import android.content.Context  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.ImageButton  
import android.widget.ImageView  
import android.widget.TextView  
import androidx.recyclerview.widget.RecyclerView  
import com.bumptech.glide.Glide  
  
class NewsAdapter(  
 private val context: Context,  
 private val articles: List<Article>,  
 private val onLikeClickListener: (Article) -> Unit,  
 private val likedArticles: Set<String>  
) : RecyclerView.Adapter<NewsAdapter.NewsViewHolder>() {  
  
 class NewsViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {  
 val newsImage: ImageView = itemView.findViewById(R.id.newsImage)  
 val newsTitle: TextView = itemView.findViewById(R.id.newsTitle)  
 val newsDescription: TextView = itemView.findViewById(R.id.newsDescription)  
 val newsSource: TextView = itemView.findViewById(R.id.newsSource)  
 val newsDate: TextView = itemView.findViewById(R.id.newsDate)  
 val likeButton: ImageButton = itemView.findViewById(R.id.likeButton)  
 }  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): NewsViewHolder {  
 val view = LayoutInflater.from(context).inflate(R.layout.item\_news, parent, false)  
 return NewsViewHolder(view)  
 }  
  
 override fun onBindViewHolder(holder: NewsViewHolder, position: Int) {  
 val article = articles[position]  
  
 holder.newsTitle.text = article.title  
 holder.newsDescription.text = article.description ?: "No description available"  
 holder.newsSource.text = article.source.name  
 holder.newsDate.text = article.publishedAt.substring(0, 10) // Just show date part  
  
 // Load image with Glide  
 if (article.urlToImage != null) {  
 Glide.with(context)  
 .load(article.urlToImage)  
 .placeholder(R.drawable.ic\_news\_placeholder)  
 .into(holder.newsImage)  
 } else {  
 holder.newsImage.setImageResource(R.drawable.ic\_news\_placeholder)  
 }  
  
 // Set like button state  
 if (likedArticles.contains(article.url)) {  
 holder.likeButton.setImageResource(R.drawable.ic\_liked)  
 } else {  
 holder.likeButton.setImageResource(R.drawable.ic\_like)  
 }  
  
 holder.likeButton.setOnClickListener **{** onLikeClickListener(article)  
 **}** }  
  
 override fun getItemCount(): Int = articles.size  
}

//NewsApiService

package com.example.newsarticle  
  
import retrofit2.Call  
import retrofit2.http.GET  
import retrofit2.http.Query  
  
interface NewsApiService {  
 @GET("top-headlines")  
 fun getTopHeadlines(  
 @Query("country") country: String,  
 @Query("apiKey") apiKey: String  
 ): Call<NewsResponse>  
  
 @GET("everything")  
 fun searchNews(  
 @Query("q") query: String,  
 @Query("apiKey") apiKey: String  
 ): Call<NewsResponse>  
}

//NewsApiResponse

package com.example.newsarticle  
  
data class NewsResponse(  
 val status: String,  
 val totalResults: Int,  
 val articles: List<Article>  
)

//Retrofit Client

package com.example.newsarticle  
import okhttp3.OkHttpClient  
import retrofit2.Retrofit  
import retrofit2.converter.gson.GsonConverterFactory  
  
object RetrofitClient {  
  
 private val client = OkHttpClient.Builder()  
 .addInterceptor **{** chain **->** val newRequest = chain.request().newBuilder()  
 .addHeader("User-Agent", "Mozilla/5.0")  
 .build()  
 chain.proceed(newRequest)  
 **}** .build()  
  
 private val retrofit = Retrofit.Builder()  
 .baseUrl("https://newsapi.org/v2/")  
 .client(client)  
 .addConverterFactory(GsonConverterFactory.create())  
 .build()  
  
 val instance: NewsApiService = retrofit.create(NewsApiService::class.*java*)  
}

//Article.kt

package com.example.newsarticle  
  
data class Article(  
 val source: Source,  
 val author: String?,  
 val title: String,  
 val description: String?,  
 val url: String,  
 val urlToImage: String?,  
 val publishedAt: String,  
 val content: String?  
)  
  
data class Source(  
 val id: String?,  
 val name: String  
)

//item\_news.xml

<?xml version="1.0" encoding="utf-8"?>  
<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_margin="8dp"  
 app:cardCornerRadius="8dp"  
 app:cardElevation="4dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical">  
  
 <ImageView  
 android:id="@+id/newsImage"  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:scaleType="centerCrop"  
 android:src="@drawable/ic\_news\_placeholder" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:padding="8dp">  
  
 <LinearLayout  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:orientation="vertical">  
  
 <TextView  
 android:id="@+id/newsTitle"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:textColor="@android:color/black"  
 android:textSize="16sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/newsDescription"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:maxLines="2"  
 android:textColor="@android:color/darker\_gray"  
 android:textSize="14sp" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:orientation="horizontal">  
  
 <TextView  
 android:id="@+id/newsSource"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textColor="@color/purple\_500"  
 android:textSize="12sp"  
 android:textStyle="bold" />  
  
 <TextView  
 android:id="@+id/newsDate"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:textColor="@android:color/darker\_gray"  
 android:textSize="12sp" />  
 </LinearLayout>  
 </LinearLayout>  
  
 <ImageButton  
 android:id="@+id/likeButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center\_vertical"  
 android:background="?attr/selectableItemBackgroundBorderless" />  
 </LinearLayout>  
 </LinearLayout>  
</androidx.cardview.widget.CardView>

//main\_activity.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 tools:context=".MainActivity">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:padding="16dp">  
  
 <EditText  
 android:id="@+id/searchEditText"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:hint="Search news..."  
 android:imeOptions="actionSearch"  
 android:inputType="text" />  
  
 <ImageButton  
 android:id="@+id/searchButton"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="8dp"  
 android:src="@drawable/ic\_search" />  
 </LinearLayout>  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:visibility="gone" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 android:id="@+id/recyclerView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent" />  
  
</LinearLayout>