



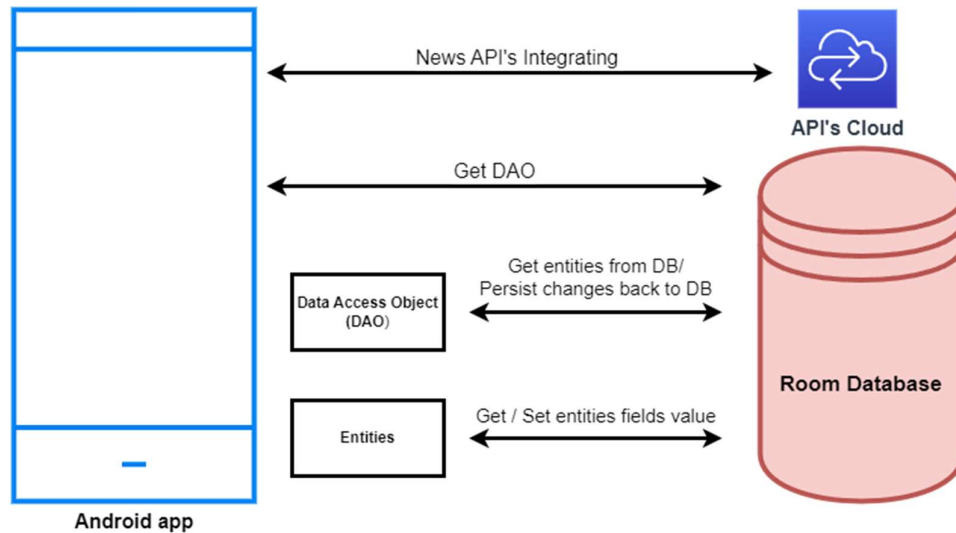
## **An Android Application for Keeping Up with the Latest Headlines**

**Project Based Experiential Learning Program**

# An Android Application for Keeping Up with the Latest Headlines :

The app's main feature is displaying a list of news articles, each with a title, image, and brief description. Users can scroll through the list of articles and tap on an article to view more details. The app uses the Jetpack Compose UI toolkit to build the UI and it uses the coil library to load images. The app fetches data from a remote server using Retrofit library and demonstrates how to use the Jetpack Compose UI toolkit for Android development.

## Architecture



## Learning Outcomes :

By end of this project:

- You'll be able to work on Android studio and build an app.
- You'll be able to integrate the database accordingly.
- You'll be able to integrate the API's accordingly.

## Project Workflow:

- Users register into the application.
- After registration , user logs into the application.
- User enters into the main page

## Tasks:

1. Creating a new project.
2. Adding required dependencies.
3. Adding permissions
4. Creating the database classes.
5. Creating API Service and required classes for integrating API
6. Building application UI and connecting to database.
7. Modifying AndroidManifest.xml
8. Running the application

### Task 1 : Creating a new project.

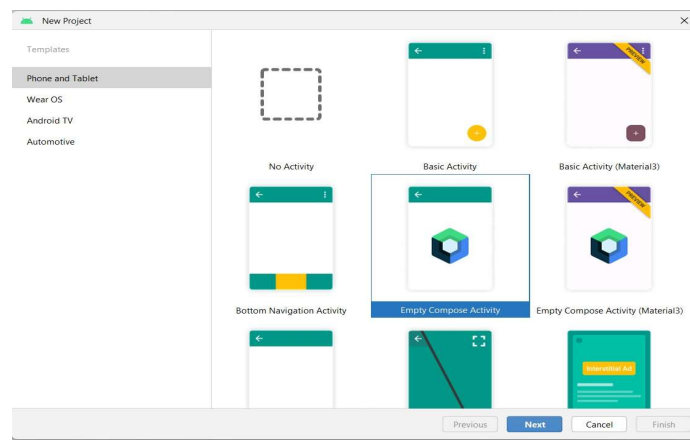
Step 1 : Android studio > File > New > New Project > Empty Compose Activity

Step 2 : Click on Next button.

Step 3 : Give name to the new project.

Step 4 : Give the Minimum SDK value

Step 5 : Click Finish



## Task 2 : Adding required dependencies.

Step 1 : Gradle scripts > build.gradle(Module :app)

Step 2 : Adding room dependencies.

Step 3 : Adding Retrofit dependencies

Step 4 : Adding Coil dependencies

Step 5 : Click on Sync now

```
49
50 dependencies {
51
52     implementation 'androidx.core:core-ktx:1.7.0'
53     implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
54     implementation 'androidx.activity:activity-compose:1.3.1'
55     implementation "androidx.compose.ui:ui:$compose_ui_version"
56     implementation "androidx.compose.ui:ui-tooling-preview:$compose_ui_version"
57     implementation 'androidx.compose.material:material:1.2.0'
58     implementation 'androidx.room:room-common:2.5.0'
59     implementation 'androidx.room:room-ktx:2.5.0'
60     implementation 'com.google.firebase:firebase-auth-ktx:22.1.0'
61     testImplementation 'junit:junit:4.13.2'
62     androidTestImplementation 'androidx.test.ext:junit:1.1.5'
63     androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
64     androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose_ui_version"
65     debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
66     debugImplementation "androidx.compose.ui:ui-test-manifest:$compose_ui_version"
67
68
69     // Retrofit
70     implementation 'com.squareup.retrofit2:retrofit:2.9.0'
71     implementation "com.squareup.okhttp3:okhttp:5.0.0-alpha.2"
72     implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
73
74
75     implementation("io.coil-kt:coil-compose:1.4.0")
76
77 }
```

## Task 3: Adding permissions

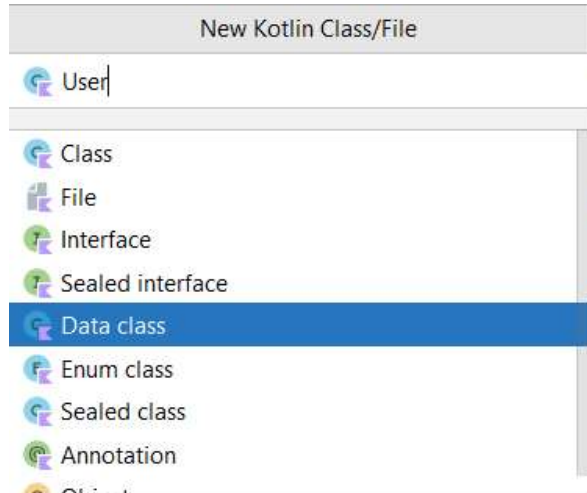
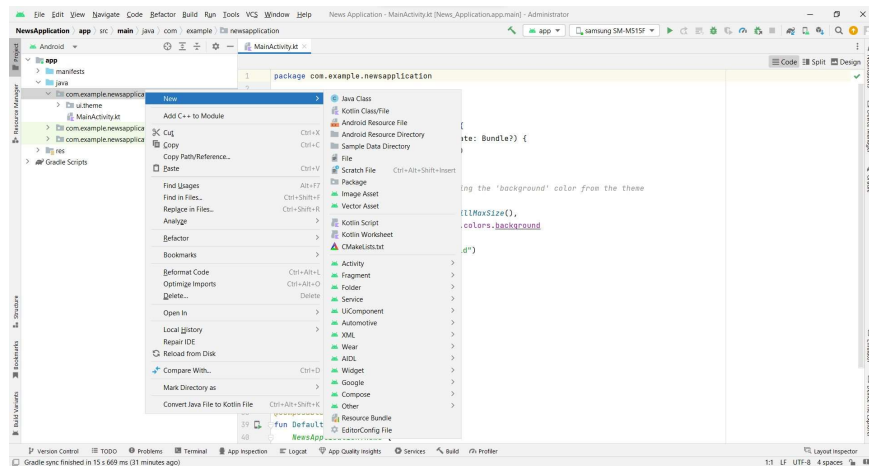
Step 1: Open AndroidManifest.xml

Step 2: Add permission to access wifi and internet

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/newsappiconmy"
        android:label="News Headlines"
        android:supportsRtl="true"
        android:theme="@style/Theme.NewsHeadlines"
        tools:targetApi="31">
        <activity
            android:name=".DisplayNews"
            android:exported="false"
            android:label="DisplayNews"
            android:theme="@style/Theme.NewsHeadlines" />
        <activity
            android:name=".RegistrationActivity"
            android:exported="false"
            android:label="RegistrationActivity"
            android:theme="@style/Theme.NewsHeadlines" />
        <activity
            android:name=".MainPage"
            android:exported="false"
            android:label="MainPage"
            android:theme="@style/Theme.NewsHeadlines" />
        <activity
            android:name=".LoginActivity"
```

## Task 4: Creating the database classes

### Step 1 : Create User data class

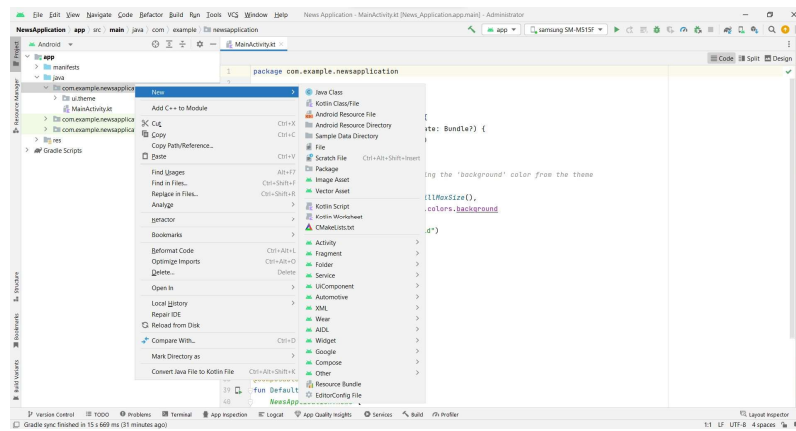


User class code:

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/User.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadlines/User.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/User.kt)

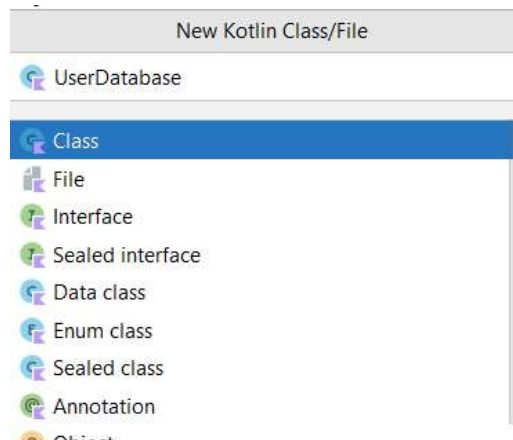
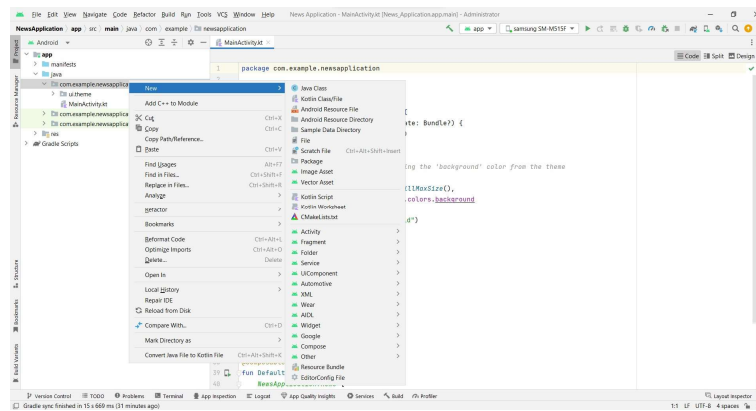
## Step 2 : Create an UserDao interface



UserDao interface code:

<https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/UserDao.kt>

### Step 3 : Create an UserDatabase class

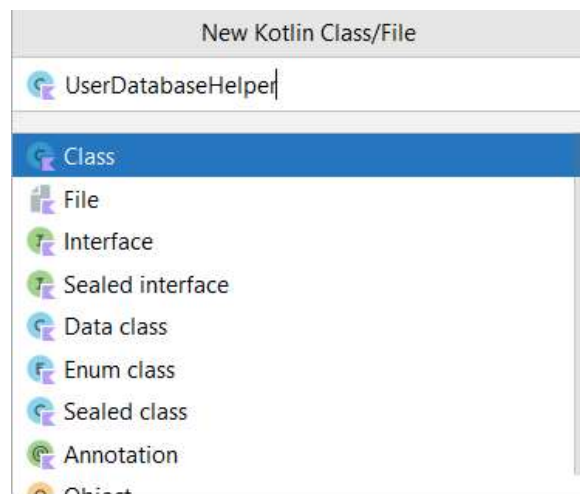
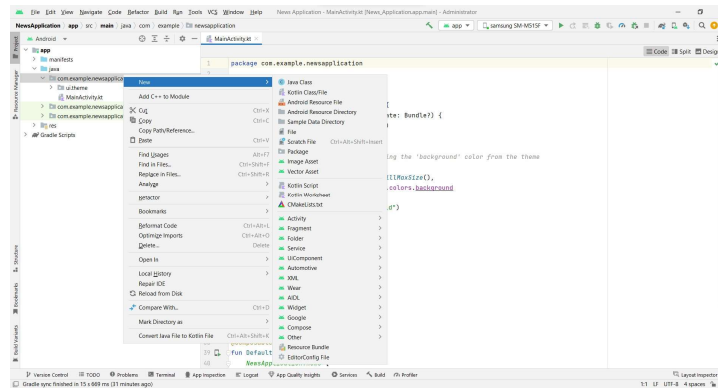


UserDatabase class code:

<https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/UserDatabase.kt>



Step 4 : Create an UserDatabaseHelper class:

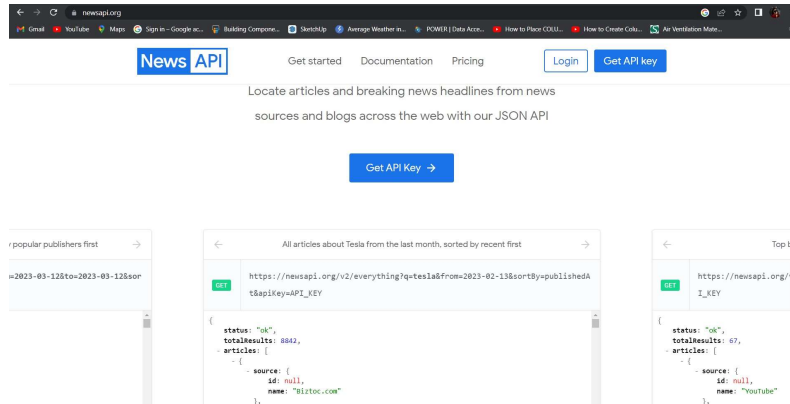


UserDatabaseHelper class code :

<https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/UserDatabaseHelper.kt>

## Task 6 :Creating API Service and required classes for integrating API

### Step 1: Create a API key for the required API

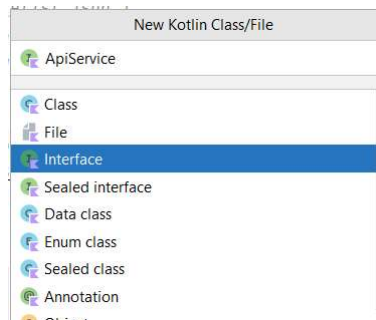


Code which is needed to be copied:

<https://newsapi.org/v2/top-headlines?country=in&apiKey=aeff87a82995467d95eb4350a7349a18>

### Database for news integration into project

### Step 3: Create ApiService interface

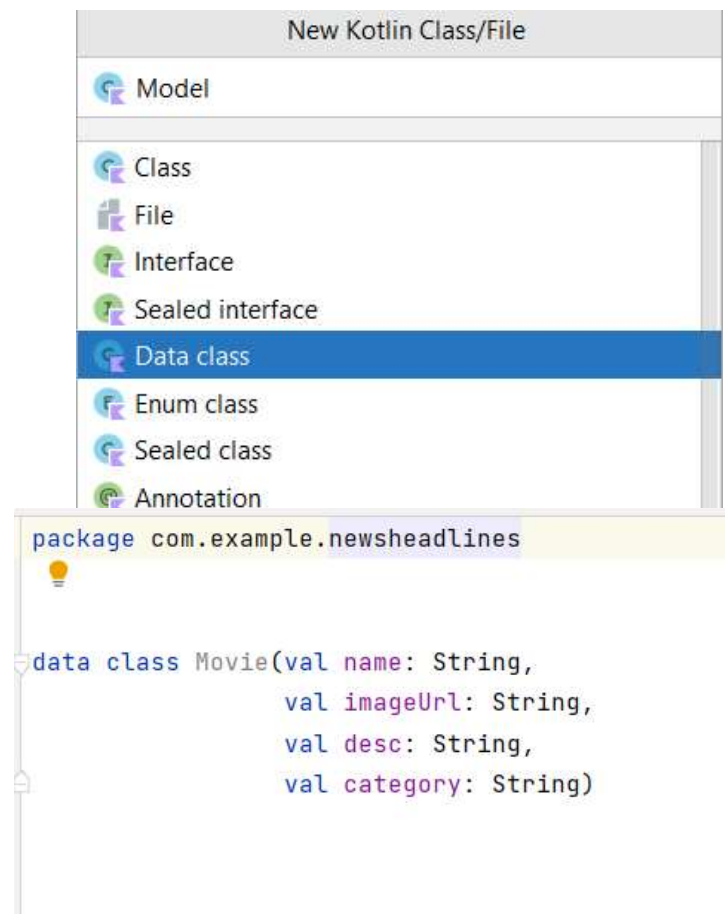
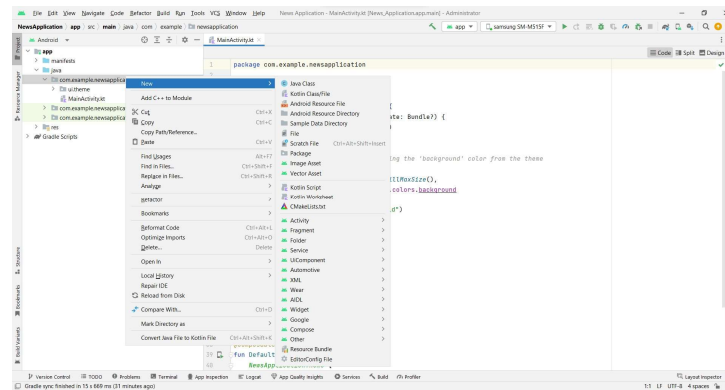


```
1 package com.example.newsHeadLines
2
3 import ...
4
5 interface ApiService {
6
7     //GET("newsList.json")
8
9     //GET("https://newsapi.org/v2/top-headlines?country=in&apiKey=aeff87a82995467d95eb4350a7349a18")
10    //GET("search?query=tesla")
11    suspend fun getNews() : News
12
13
14    companion object {
15        var apiService: ApiService? = null
16        fun getInstance() : ApiService {
17            if (apiService == null) {
18                apiService = Retrofit.Builder().baseUrl(
19                    // baseUrl("https://newsapi.org/v2/")
20                    .baseUrl("https://newsapi.org/v2/")
21                    // baseUrl("https://newsapi.org/v2/")
22                    // baseUrl("https://newsapi.org/v2/")
23                ).addConverterFactory(GsonConverterFactory.create())
24                .build().create(ApiService::class.java)
25            }
26            return apiService!!
27        }
28    }
29
30 }
```

ApiService interface code:

<https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/ApiService.kt>

## Step 4: Create Model data class

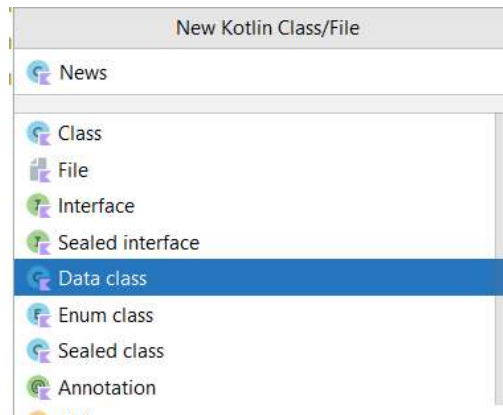
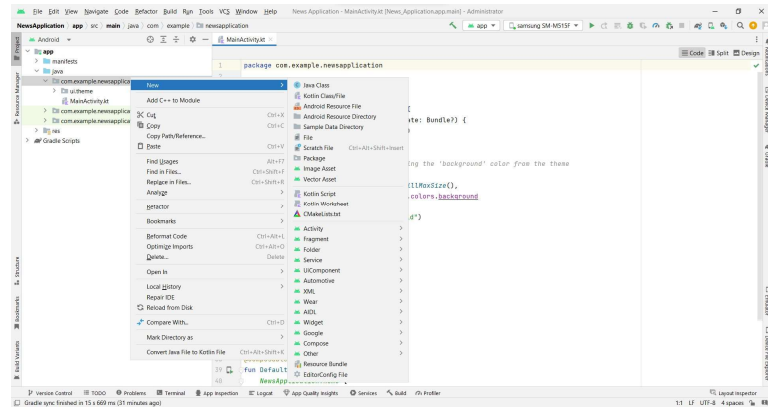


**Model class code:**

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Model.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Model.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Model.kt)

## Step 5: Create News data class

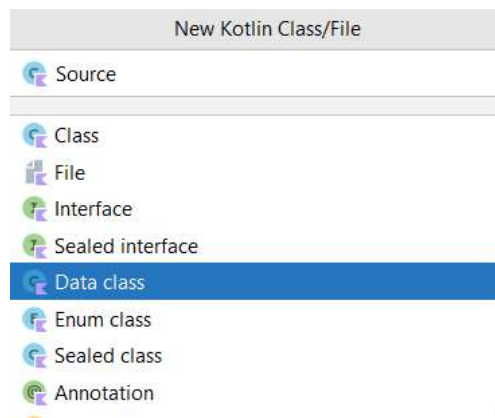
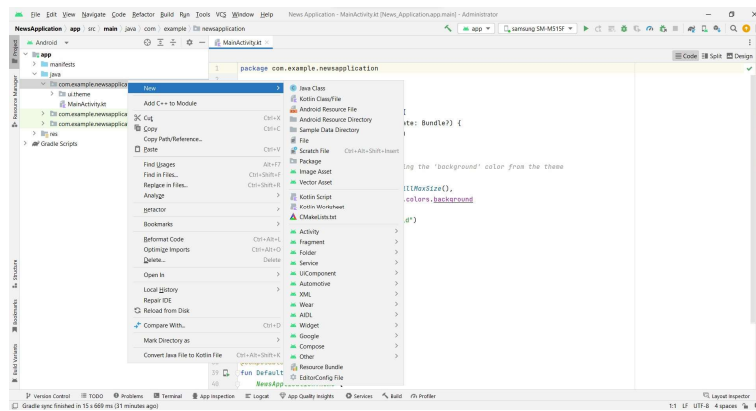


**News Data Class Code:**

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/News.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadlines/News.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/News.kt)

## Step 6: Create Source data class:

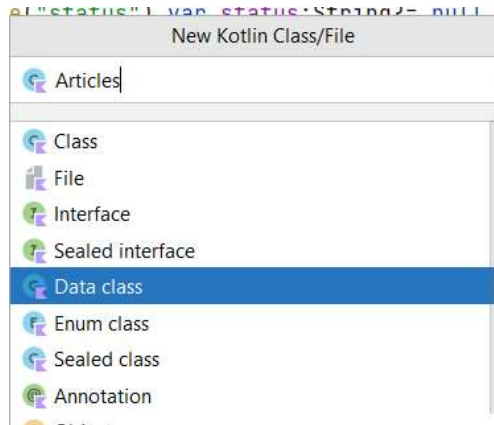
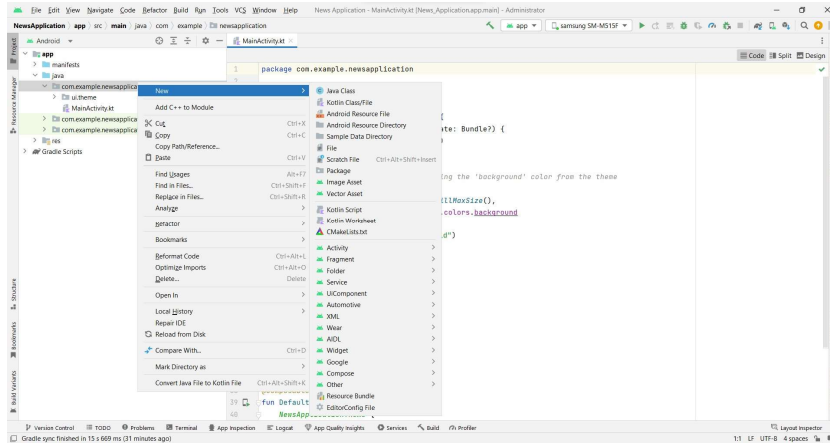


Source data class code:

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Source.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Source.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Source.kt)

## Step 7: Create Articles data class

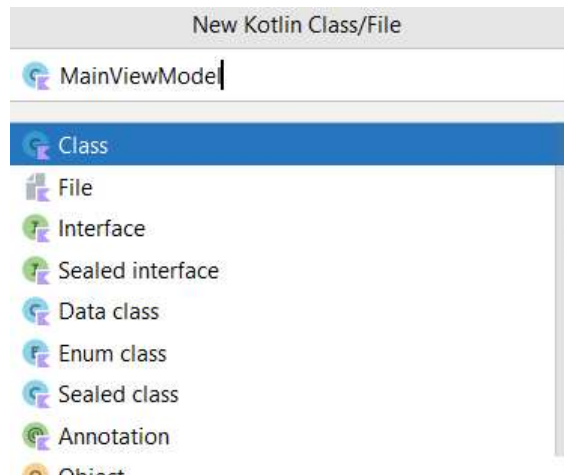
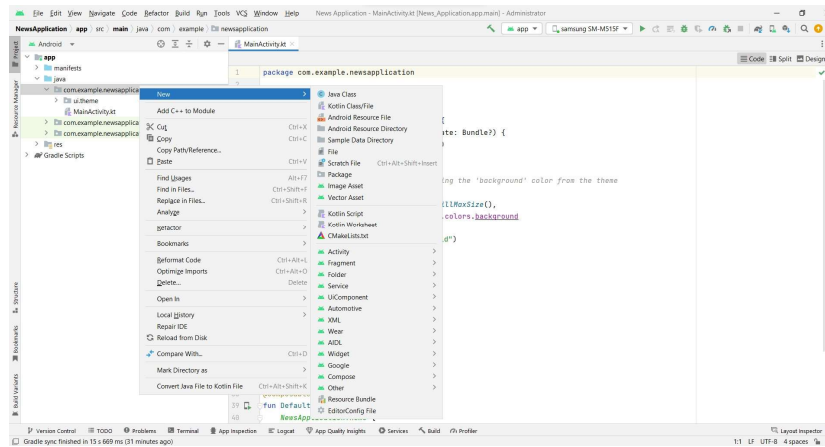


Articles data code:

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Articles.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Articles.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/Articles.kt)

## Step 8: Create MainViewModel class



```
package com.example.newsheadlines

import ...

class MainViewModel : ViewModel() {
    var movieListResponse: List<Articles> by mutableStateOf(listOf())
    var errorMessage: String by mutableStateOf( value: "")
    fun getMovieList() {
        viewModelScope.launch { this: CoroutineScope
            val apiService = ApiService.getInstance()
            try {
                val movieList = apiService.getMovies()
                movieListResponse = movieList.articles
            }
            catch (e: Exception) {
                errorMessage = e.message.toString()
            }
        }
    }
}
```

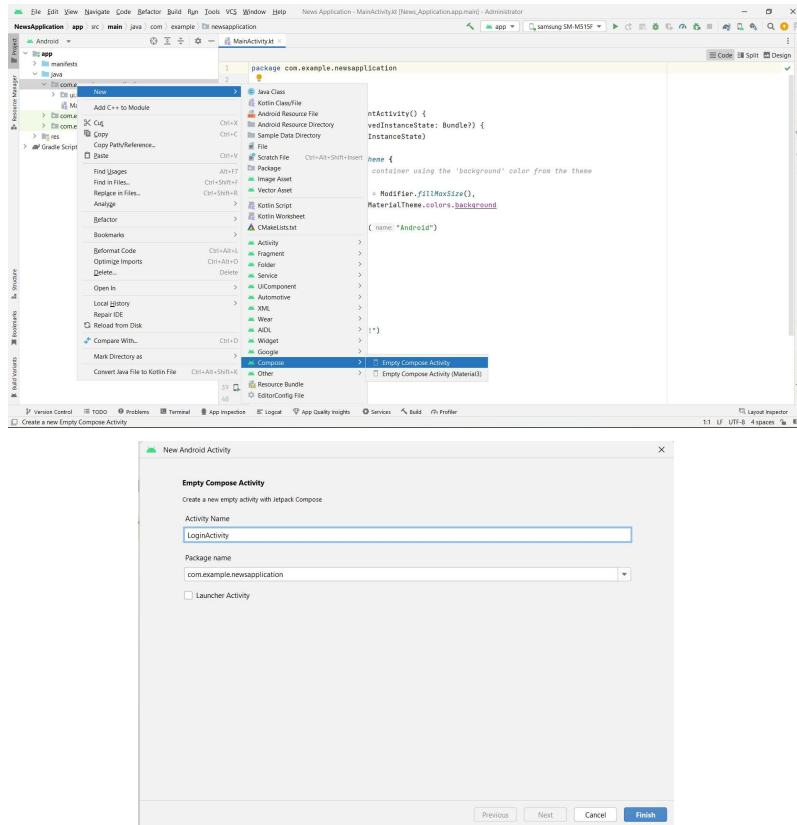
Main view model class code:

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/MainViewModel.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadlines/MainViewModel.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/MainViewModel.kt)

## Task 7: Building application UI and connecting to database.

### Step 1: Creating LoginActivity.kt with database



```
package com.example.newsheadlines

import ...

class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(context = this)
        setContent {
            LoginScreen(context = this, databaseHelper)
        }
    }
}

@Composable
fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {
    var username by remember { mutableStateOf<String>("") }
    var password by remember { mutableStateOf<String>("") }
    var error by remember { mutableStateOf<String>("") }

    Column(
        Modifier
            .fillMaxHeight()
            .fillMaxWidth()
            .padding(28.dp),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        // this ColumnScope
    }
}
```

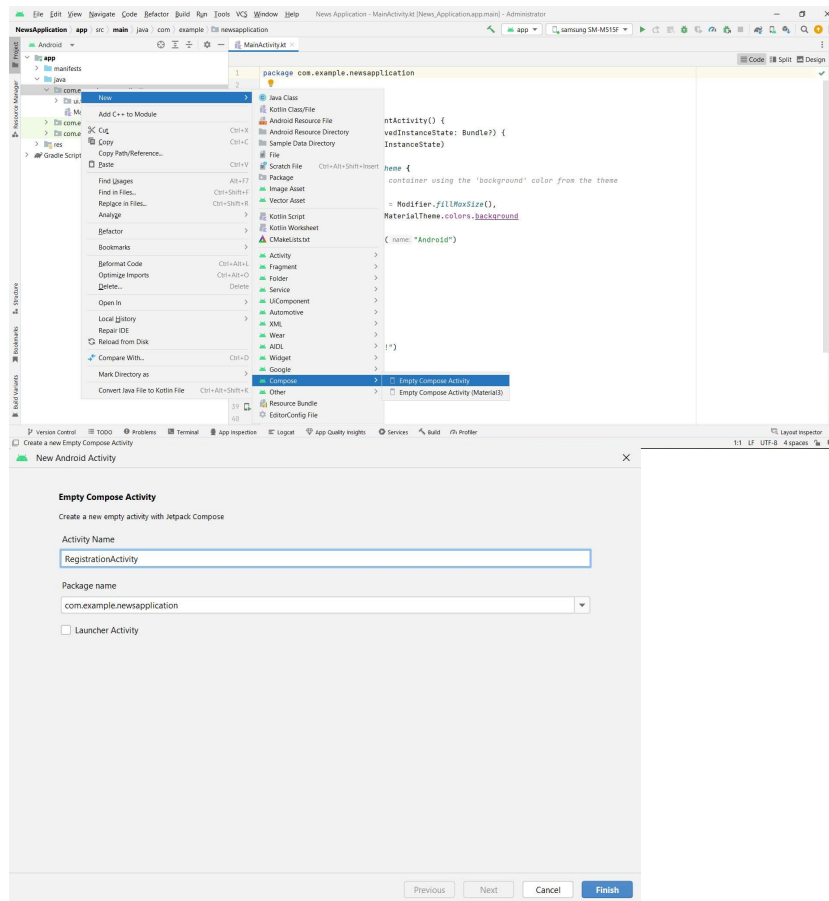
Loginactivity code:

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/LoginActivity.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadlines/LoginActivity.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/LoginActivity.kt)



## Step 2 : Creating RegistrationActivity.kt with database



```
package com.example.newsheadLines

import ...

class RegistrationActivity : ComponentActivity() {
    private lateinit var dbHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        dbHelper = UserDatabaseHelper(context, this)
        setContent {
            RegistrationScreen(context, dbHelper)
        }
    }
}

@Composable
fun RegistrationScreen(context: Context, dbHelper: UserDatabaseHelper) {
    var username by remember { mutableStateOf<String>("") }
    var password by remember { mutableStateOf<String>("") }
    var email by remember { mutableStateOf<String>("") }
    var error by remember { mutableStateOf<String>("") }

    Column {
        Modifier
            .background(Color.White)
            .fillMaxHeight()
            .fillMaxWidth()
            .horizontalAlignment = Alignment.CenterHorizontally,
```

Registration Activity code:

[https://github.com/vindifier/Latest-](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadLines/RegistrationActivity.kt)

[Headlines/blob/master/app/src/main/java/com/example/newsheadLines/RegistrationActivity.kt](https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadLines/RegistrationActivity.kt)

## Step 3 : Creating MainActivity.kt file

In MainActivity.kt file the main application is developed

```
package com.example.newsheadlines

import ...

class MainPage : ComponentActivity() {
    val mainViewModel by viewModels<MainViewModel>()
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            NewsHeadlinesTheme {
                // A surface container using the 'background' color from the theme
                Surface(color = MaterialTheme.colors.background) {
                    Column() { this: ColumnScope

                        Text(text = "Latest NEWS", fontSize = 32.sp, modifier = Modifier.fillMaxWidth(), textAlign = TextAlign.Center)

                        MovieList(applicationContext, movieList = mainViewModel.movieListResponse)
                        mainViewModel.getMovieList()

                    }
                }
            }
        }
    }
}

@Composable
fun MovieList(context: Context, movieList: List<Articles>) {
    var selectedIndex by remember { mutableStateOf<Int>(-1) }
    LazyColumn { this: LazyListScope

        itemsIndexed(items = movieList) { this: LazyItemScope
```

Main page activity code:

<https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/MainPage.kt>

## Step 4 : Creating DisplayNews.kt file

```
package com.example.newsheadlines

import ...

class DisplayNews : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            NewsHeadlinesTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.secondaryVariant
                ) {
                    var desk = getIntent().getStringExtra("desk")
                    var title = getIntent().getStringExtra("title")
                    var urlImage = getIntent().getStringExtra("urlToImage")
                    Log.i(tag = "test123abc", msg = "MovieItem: $desk")

                    Column(modifier = Modifier.background(Color.LightGray).padding(20.dp), horizontalAlignment = HorizontalAlignment.Center) {
                        Text(text = "$title", fontSize = 32.sp)
                        HtmlText(html = desk.toString())
                        /* AsyncImage(
                            model = "https://example.com/image.jpg",
                            contentDescription = "Translated description of what the image contains"
                        )*/
                        Image(
                            painter = rememberImagePainter(urlImage),
                            contentDescription = "My content description",
                        )
                    }
                }
            }
        }
    }
}
```

Display news code:

<https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/java/com/example/newsheadlines/DisplayNews.kt>

## Task 8: Modifying AndroidManifest.xml

```
<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@drawable/newsappiconmy"
    android:label="News Headlines"
    android:supportRtl="true"
    android:theme="@style/Theme.NewsHeadlines"
    tools:targetApi="31">
    <activity
        android:name=".DisplayNews"
        android:exported="false"
        android:label="DisplayNews"
        android:theme="@style/Theme.NewsHeadlines" />
    <activity
        android:name=".RegistrationActivity"
        android:exported="false"
        android:label="RegistrationActivity"
        android:theme="@style/Theme.NewsHeadlines" />
    <activity
        android:name=".MainPage"
        android:exported="false"
        android:label="@string/title_activity_main_page"
        android:theme="@style/Theme.NewsHeadlines" />
    <activity
        android:name=".LoginActivity"
        android:exported="true"
        android:label="News Headlines"
        android:theme="@style/Theme.NewsHeadlines">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
```

**Complete Code is given:**

<https://github.com/vindifier/Latest-Headlines/blob/master/app/src/main/AndroidManifest.xml>

## Task 9 : Running app on the hardware

**For the tutorial click the below link:**

<https://developer.android.com/studio/run/device>

**Run Application on the emulator:**

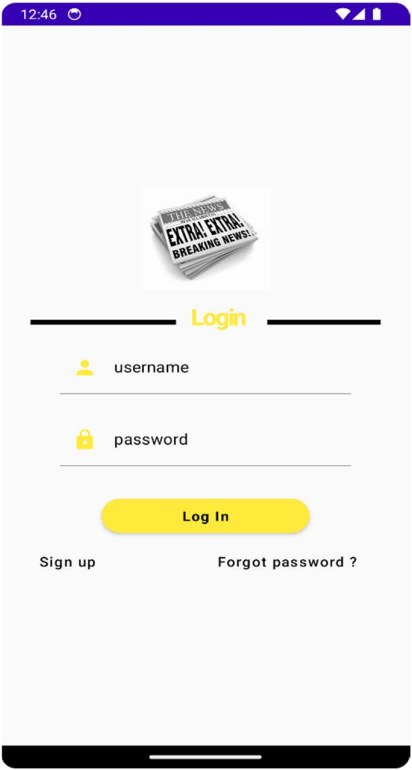


**For Complete Project Link:**

<https://github.com/vindifier/Latest-Headlines>

Login Page:

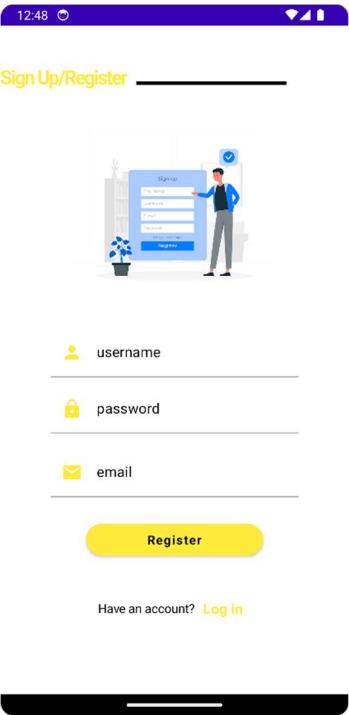
12:46



The login page features a purple status bar at the top with the time 12:46 and standard Android icons. The main content area is white. At the top center is an illustration of a newspaper with the headline 'EXTRA! EXTRA! BREAKING NEWS!'. Below this is a horizontal line with the word 'Login' in yellow text in the center. Underneath are two input fields: the first is labeled 'username' with a yellow person icon, and the second is labeled 'password' with a yellow lock icon. Below these fields is a yellow rounded button with the text 'Log In'. At the bottom, there are two links: 'Sign up' on the left and 'Forgot password ?' on the right. The page is framed by a black bottom bar.

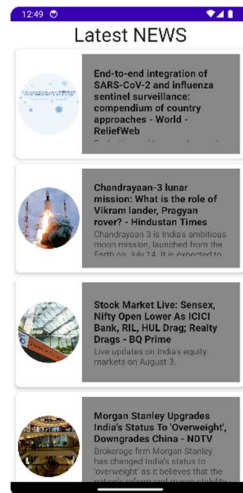
Signup Page:

12:48



The signup page features a purple status bar at the top with the time 12:48 and standard Android icons. The main content area is white. At the top left is the text 'Sign Up/Register' in yellow, followed by a horizontal line. Below this is an illustration of a person standing next to a sign that says 'Sign Up' with a list of fields and a 'Register' button. Underneath are three input fields: the first is labeled 'username' with a yellow person icon, the second is labeled 'password' with a yellow lock icon, and the third is labeled 'email' with a yellow envelope icon. Below these fields is a yellow rounded button with the text 'Register'. At the bottom, there is a link 'Have an account? Log in' where 'Log in' is in yellow. The page is framed by a black bottom bar.

Main Page:



News View Page:



Submitted By:

Team ID : LTVIP2023TMID04251

Team Size : 4

Team Leader : Bynagari Vinaykumar

Team member : Chennuru Saiprathap

Team member : Chinnappaiah Sivasankar

Team member : Chowda Chandrasekhar