Project Manual

Date	06 November 2023
Team ID	SI-GuidedProject-587104-1696656485
Project Name	An Android Application for Keeping Up with the Latest Headlines
Team Members	Aarush Jain, Hemantha Kumar Karthikeya, Sanjana Sivakumar, Siddhi Kadam
Maximum Marks	

Android Application for Keeping Up with the Latest Headlines

In today's era of the internet, a multitude of news articles and gossip pieces are easily accessed, by a simple click of a button. But how many of these news articles are written by verified news outlets, and how many sources of the gossip pieces have verifiable sources? Sometimes, the amount of unorganized news floating through the web seems to be too much to handle, and this discourages people from even knowing the events occurring all around the world.

An Android application for keeping up with the latest headlines, can be just the solution for this problem. The application organizes news depending on the news outlets, and the topics it contains. You can now read news straight from your most dependable news outlets.

The application's main feature is to fetch news articles from different news outlets. The Home Page, News Page and Specific News Page are all scrollable. The app is available in dark mode, and in light mode. The app also stores user data when they first sign up.

Architecture

Learning Outcomes

Once you have successfully completed creating an android application that allows users to keep up with the latest headlines, you will be able to:

- Work on Android Studio to create any application
- Integrate databases to your application to store data
- Integrate API's into your application

Project Workflow

1. First, the user will open the app's login page and type in their login information (username and password). This information is stored in the database.
2. If the user does not have an account, they can click on "Not Registered? Click Here". This will navigate them to the sign up page, where they can type their username, password and email account. Once they have signed up, they will be redirected to the log-in page. If the user already has an account, they can click on "Already Registered? Click Here" to redirect them to the login page.
3. After logging in successfully, the user is directed to the Home Page. Here, there is a number of news outlets that the user can choose to get their news from.
4. Once they have selected a news outlet, they will be directed to the News Page. Here, there are a variety of news pieces from said news outlet.
5. When the user has selected a news piece, they will be directed to a Specific News Page, where a detailed essay about the new piece that they have chosen will be visible.

TASKS

1. **Start with the essential first actions.**
2. **Begin a new project.**
3. **Include necessary dependencies.**
4. **Set up authorization and access permissions.**
5. **Develop the database structure.**
6. **Establish an API Service and the relevant classes to incorporate external services.**
7. **Construct the user interface of the application and establish connections with the database.**
8. **Make adjustments to the AndroidManifest.xml file.**
9. **Execute the application to test it.**

TASK 1: Start with essential first actions

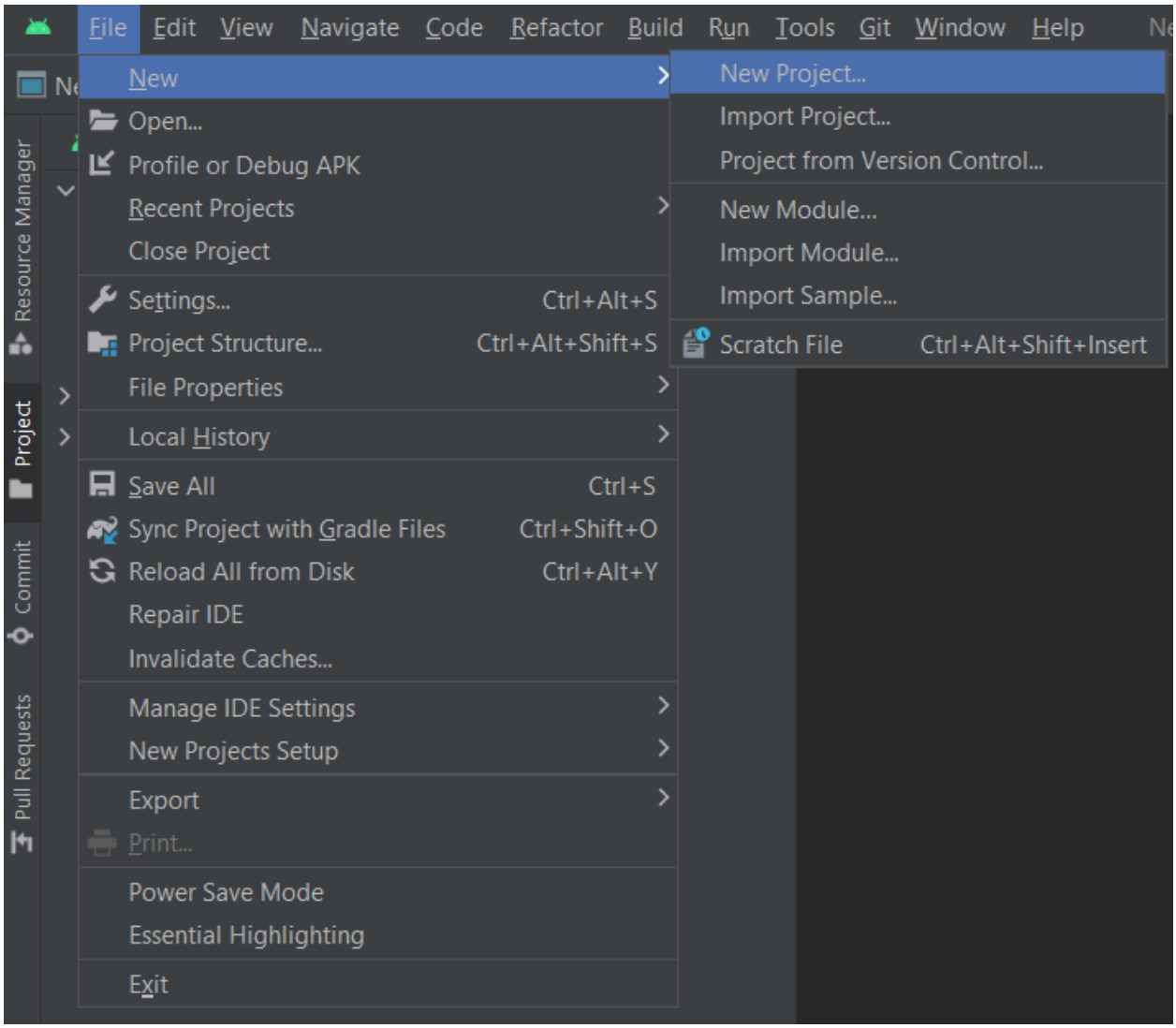
1. Download the latest version of Android Studio.

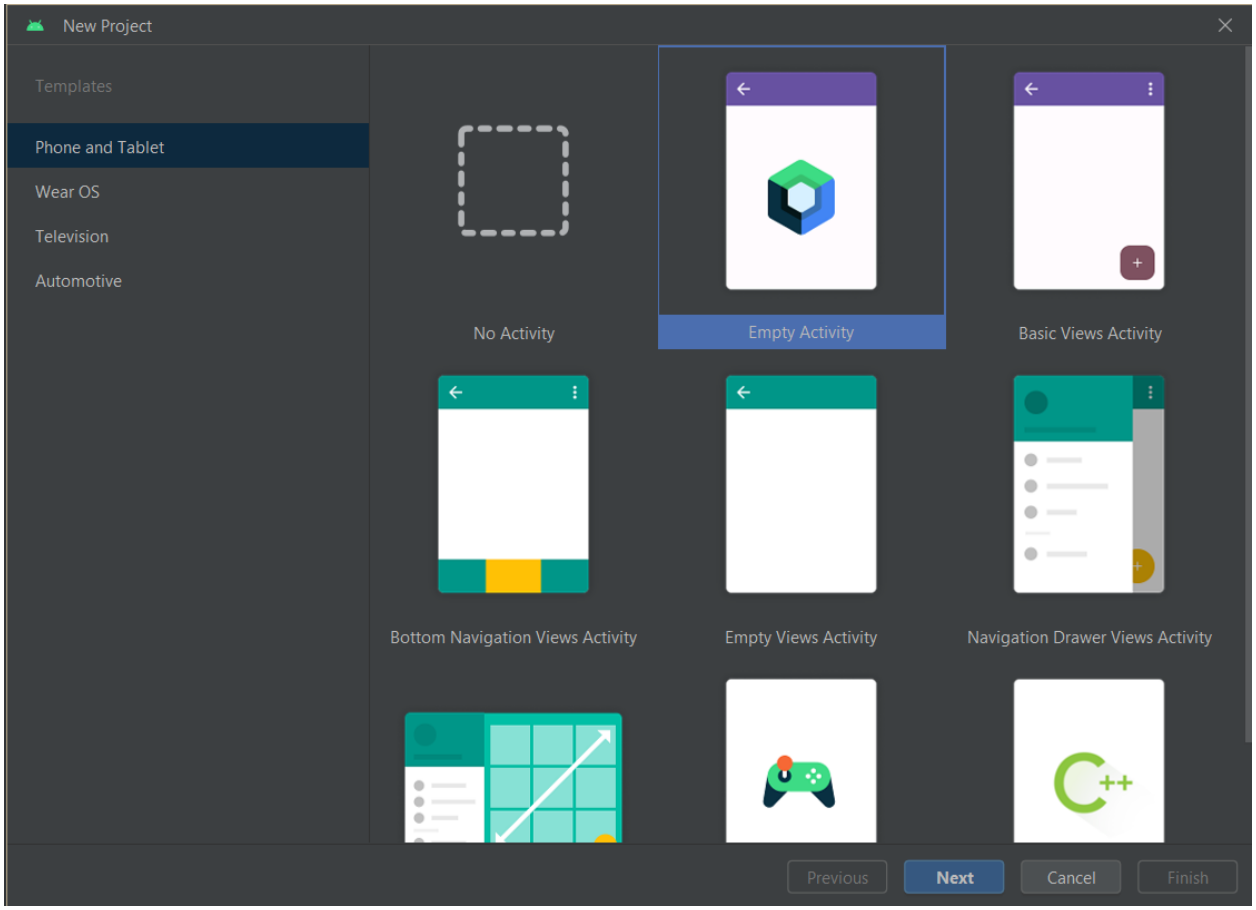
https://developer.android.com/studio

2. Open Android Studio.

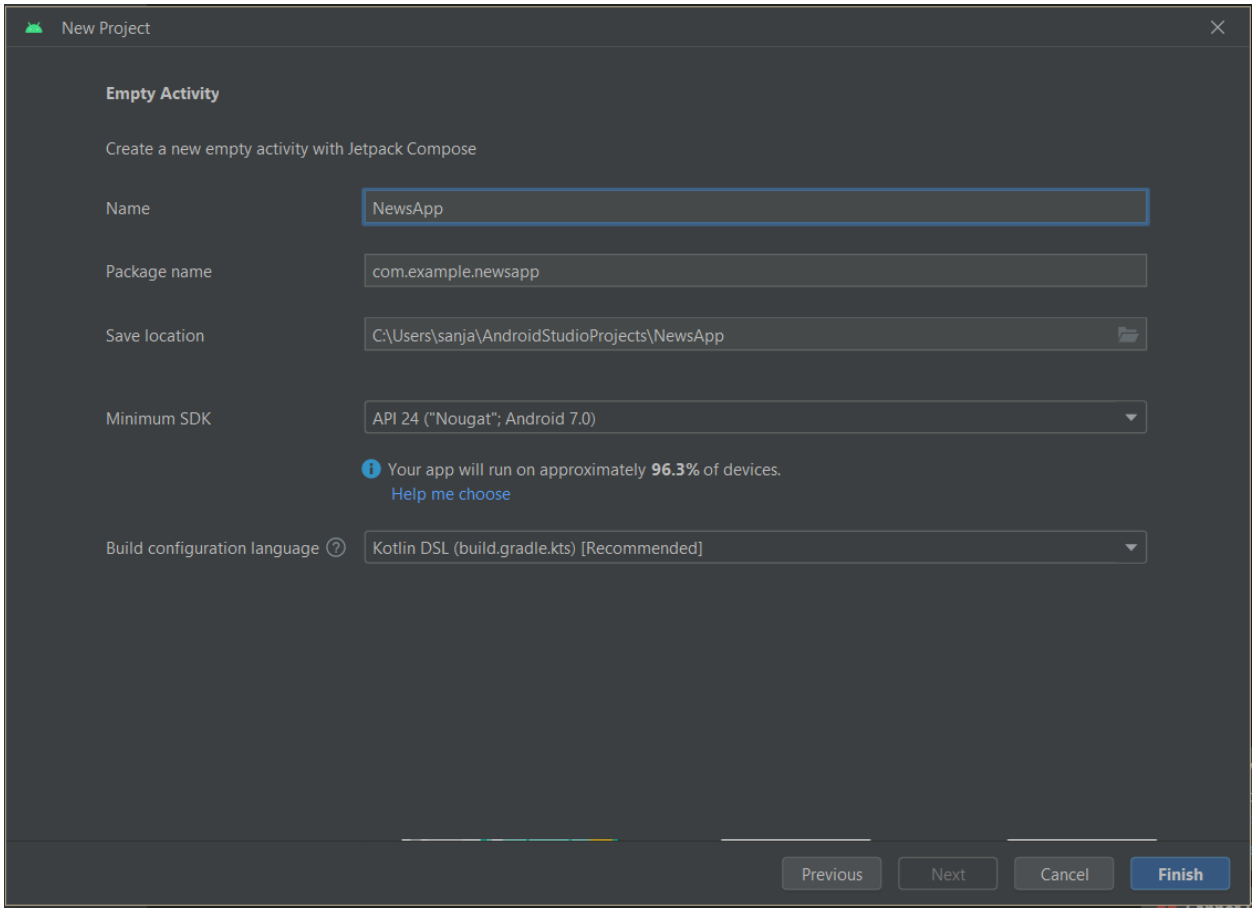
TASK 2: Begin a new project

1. In Android Studio, go to File, New, New Project and then Empty Activity.


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
New
Open...
Profile or Debug APK
Recent Projects
Close Project
Settings... Ctrl+Alt+S
Project Structure... Ctrl+Alt+Shift+S
File Properties
Local History
Save All Ctrl+S
Sync Project with Gradle Files Ctrl+Shift+O
Reload All from Disk Ctrl+Alt+Y
Repair IDE
Invalidate Caches...
Manage IDE Settings
New Projects Setup
Export
Print...
Power Save Mode
Essential Highlighting
Exit
New Project...
Import Project...
Project from Version Control...
New Module...
Import Module...
Import Sample...
Scratch File Ctrl+Alt+Shift+Insert
Resource Manager
Project
Commit
Pull Requests



2. Give a name to the project and give the minimum SDK value.



TASK 3: Include necessary dependencies

1. Go to Gradle Scripts on the left panel, and go to build.gradle(Module : app).
2. Add the required dependencies.
3. Click on Sync now.

build.gradle (:app)

```
plugins {
    id 'com.android.application'
    id 'dagger.hilt.android.plugin'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-android'
    id 'kotlin-kapt'
}

android {
    namespace 'com.aarush.newsapp'
    compileSdk 33

    defaultConfig {
        applicationId "com.aarush.newsapp"
        minSdk 24
        targetSdk 33
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
```

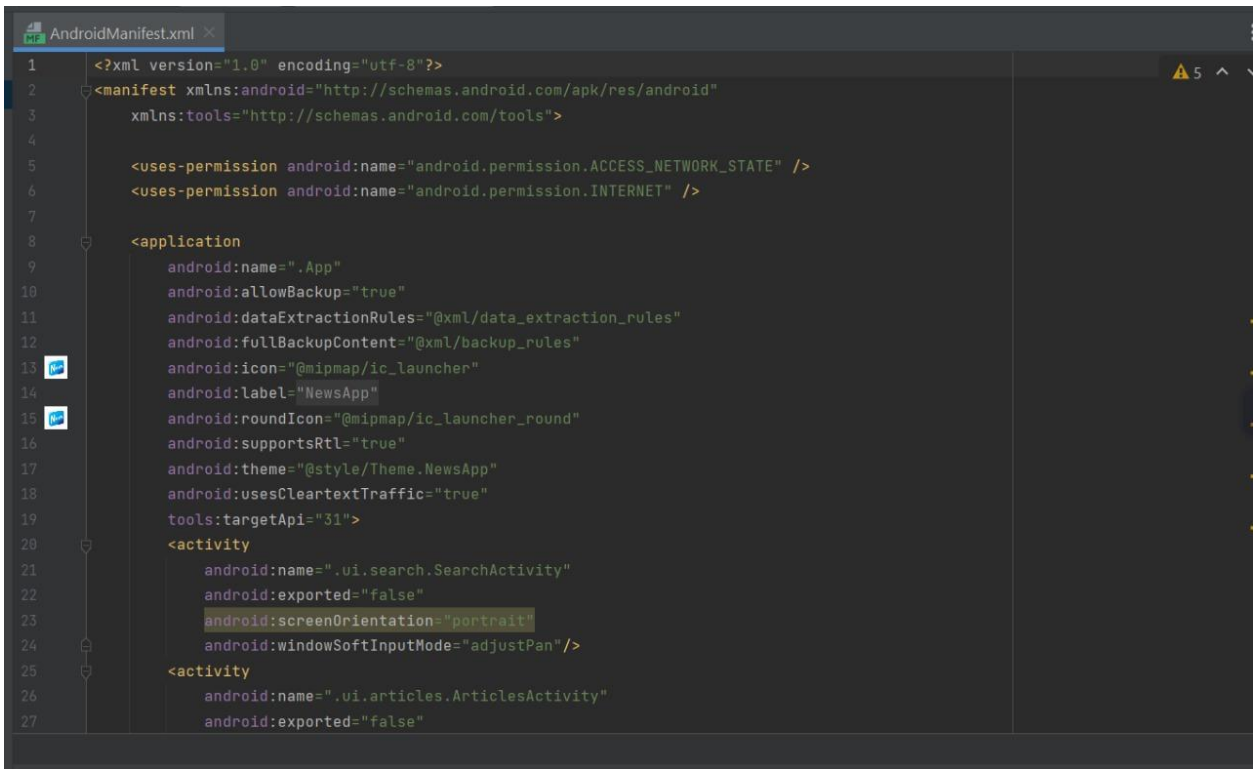
build.gradle (:app)

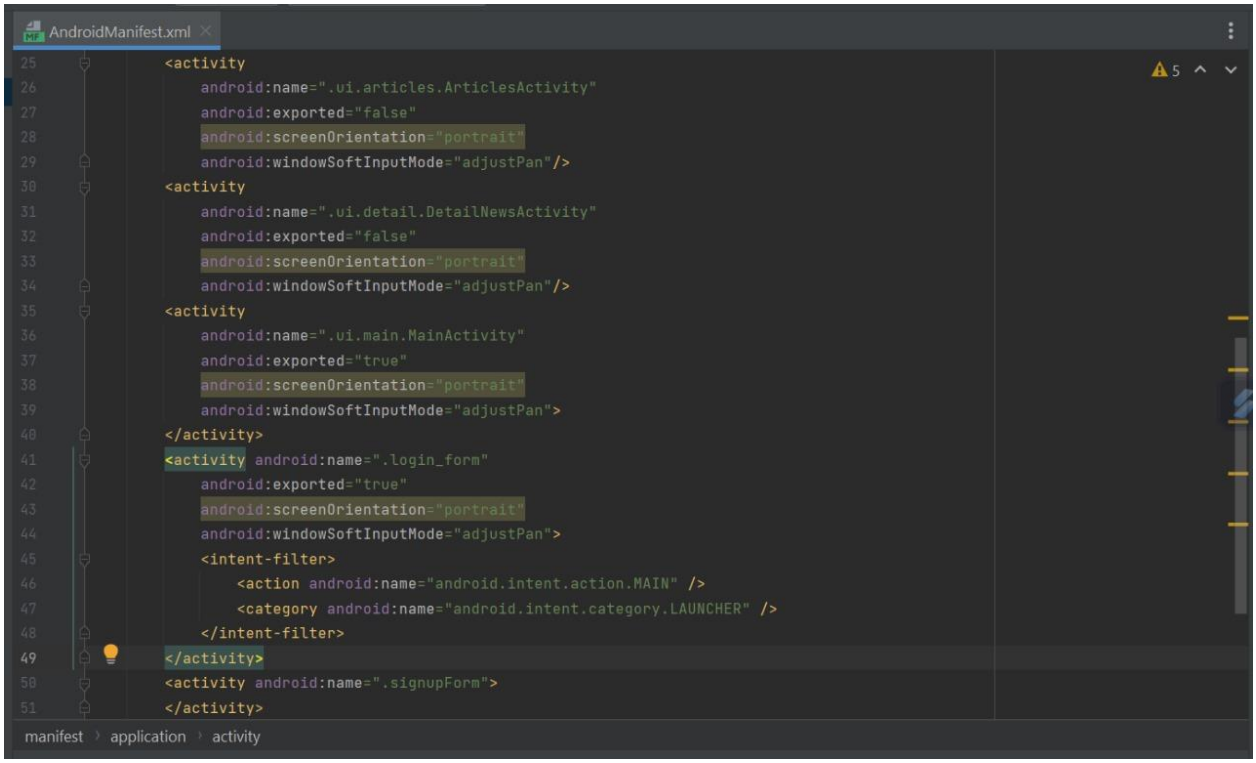
```
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
    buildFeatures {
        viewBinding true
    }
}

apply from: '../shared_dependencies.gradle'

dependencies {
    implementation project(":core")
    implementation 'androidx.appcompat:appcompat:1.6.1'
    implementation 'com.google.android.material:material:1.5.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'androidx.test.ext:junit-ktx:1.1.5'
    implementation 'androidx.test:monitor:1.6.1'
    androidTestImplementation 'junit:junit:4.12'
    androidTestImplementation 'junit:junit:4.12'
}
```

dependencies{}

TASK 4: Set up authorization and access permissions

1. Go to app on the left panel, go to manifests, and click on AndroidManifest.xml.
2. Add permissions.





TASK 5: Develop the database structure

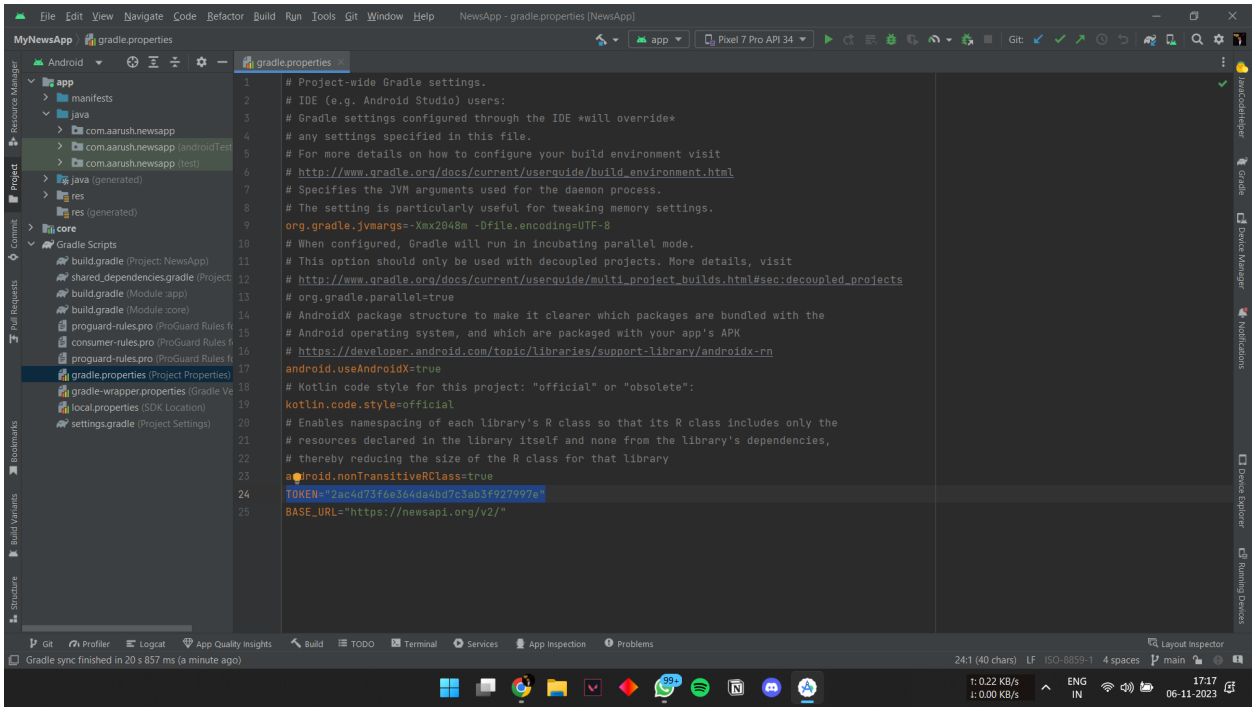
1. In your XML layout files, you design the user interface components for the login and registration screens. You create input fields for the username and password, along with buttons to handle login and registration actions. These XML files define the visual structure of your app, allowing users to interact with the authentication system.

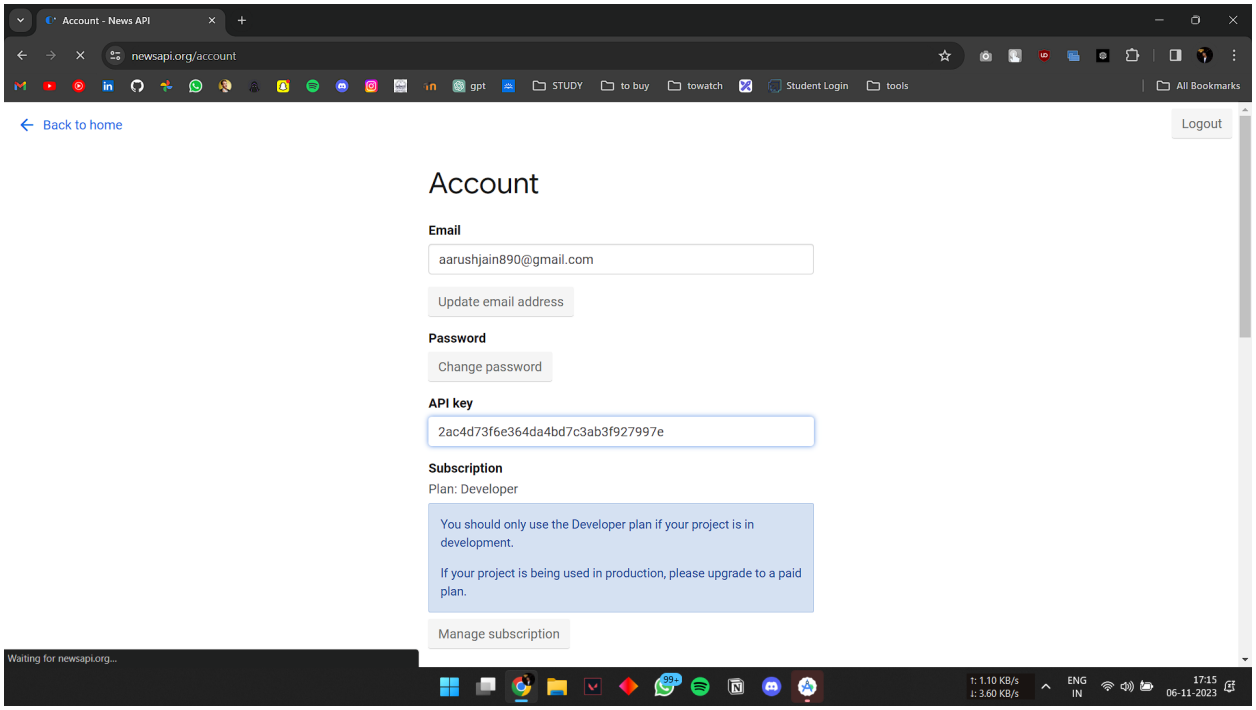
2. You implement the SQLite database in the backend using Java. You create a DatabaseHelper class that extends SQLiteOpenHelper. This class manages the database creation, version management, and provides methods for performing CRUD (Create, Read, Update, Delete) operations on the user data. Within this class, you define the database schema, including the table structure for storing usernames and passwords securely.
3. In SQLite, the database is stored locally on your device. DB_class stores data (from signup page) in a relational database

TASK 6: Establish an API Service and the relevant classes to incorporate external services

https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485/releases/tag/apk1.0.1

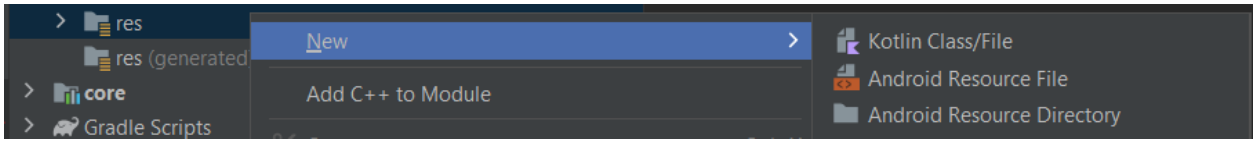
1. ArticlesViewModel.kt, ArcticlesActivity.kt and ArticlesAdapter.kt are used for the API to fetch news articles and data from respective news articles

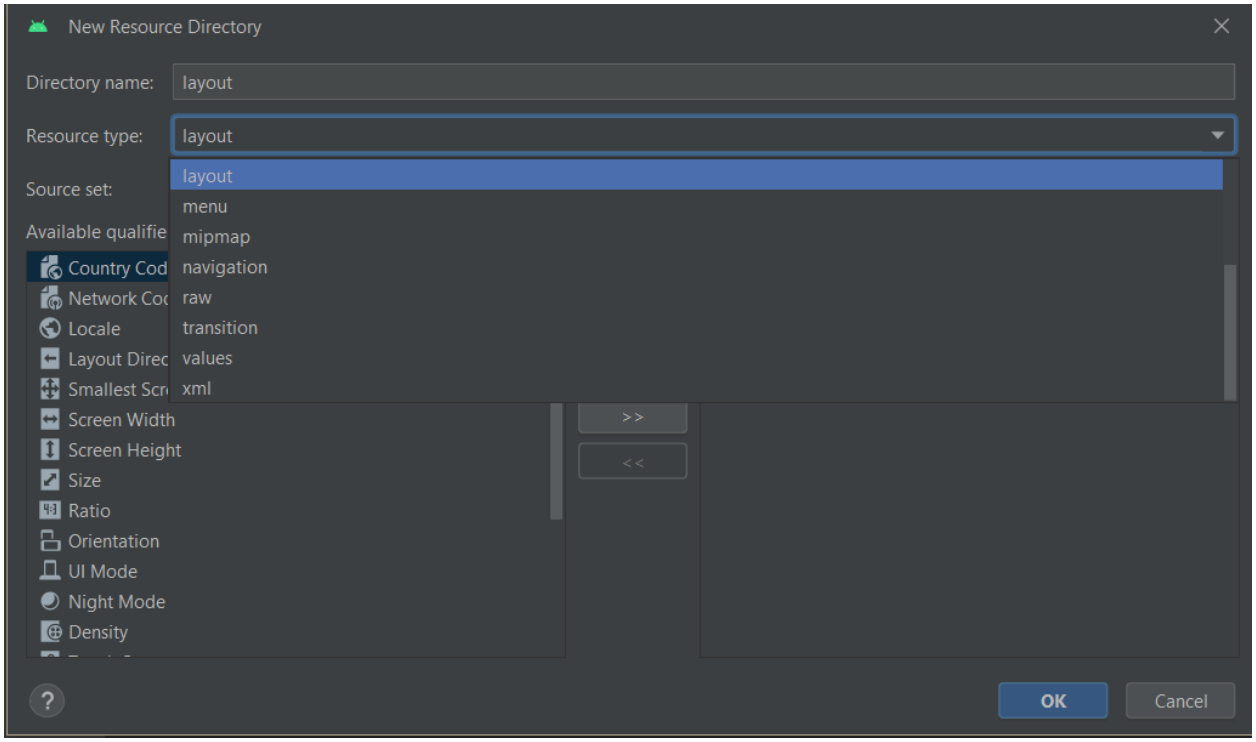


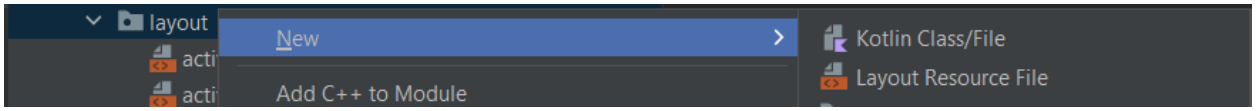


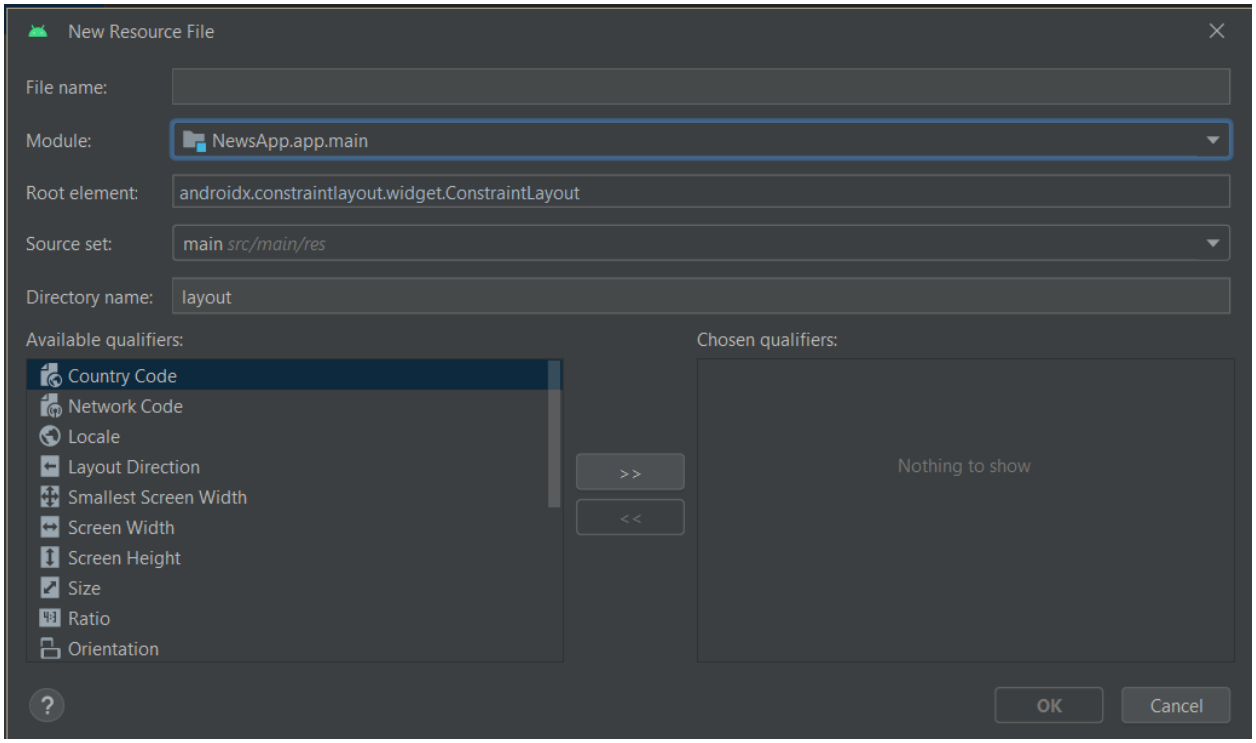
TASK 7: Construct the user interface of the application and establish connections with the database

1. Use xml files to construct user interfaces for each screen, and they must be stored in an Android Resource Directory.
2. Right click on the resource folder, click on New, and click on Android Resource Directory.
3. Change the directory name to layout and change resource type to layout.
4. There will be an xml file for (). To create xml file, right click on resource folder, click on New, and click on the Layout resource file. Change the file name, depending on what screen the xml file will represent.
5. Use design and code to create a UI.




New Resource Directory
Directory name:
layout
Resource type:
layout
Source set:
Available qualifie
layout
menu
mipmap
navigation
raw
transition
values
xml
Country Cod
Network Coc
Locale
Layout Direc
Smallest Scr
Screen Width
Screen Height
Size
Ratio
Orientation
UI Mode
Night Mode
Density
>>
<<
OK
Cancel


layout
New
Add C++ to Module
Kotlin Class/File
Layout Resource File

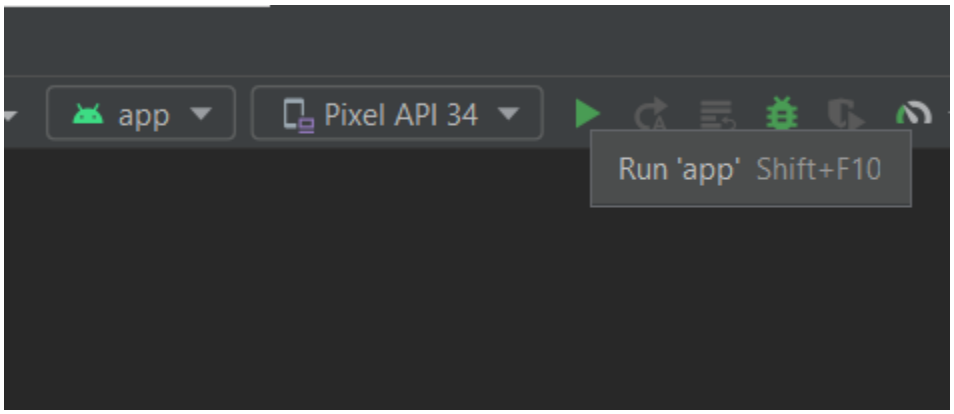

New Resource File
File name:
Module:
NewsApp.app.main
Root element:
androidx.constraintlayout.widget.ConstraintLayout
Source set:
main src/main/res
Directory name:
layout
Available qualifiers:
Chosen qualifiers:
Country Code
Network Code
Locale
Layout Direction
Smallest Screen Width
Screen Width
Screen Height
Size
Ratio
Orientation
>>
<<
Nothing to show
OK
Cancel

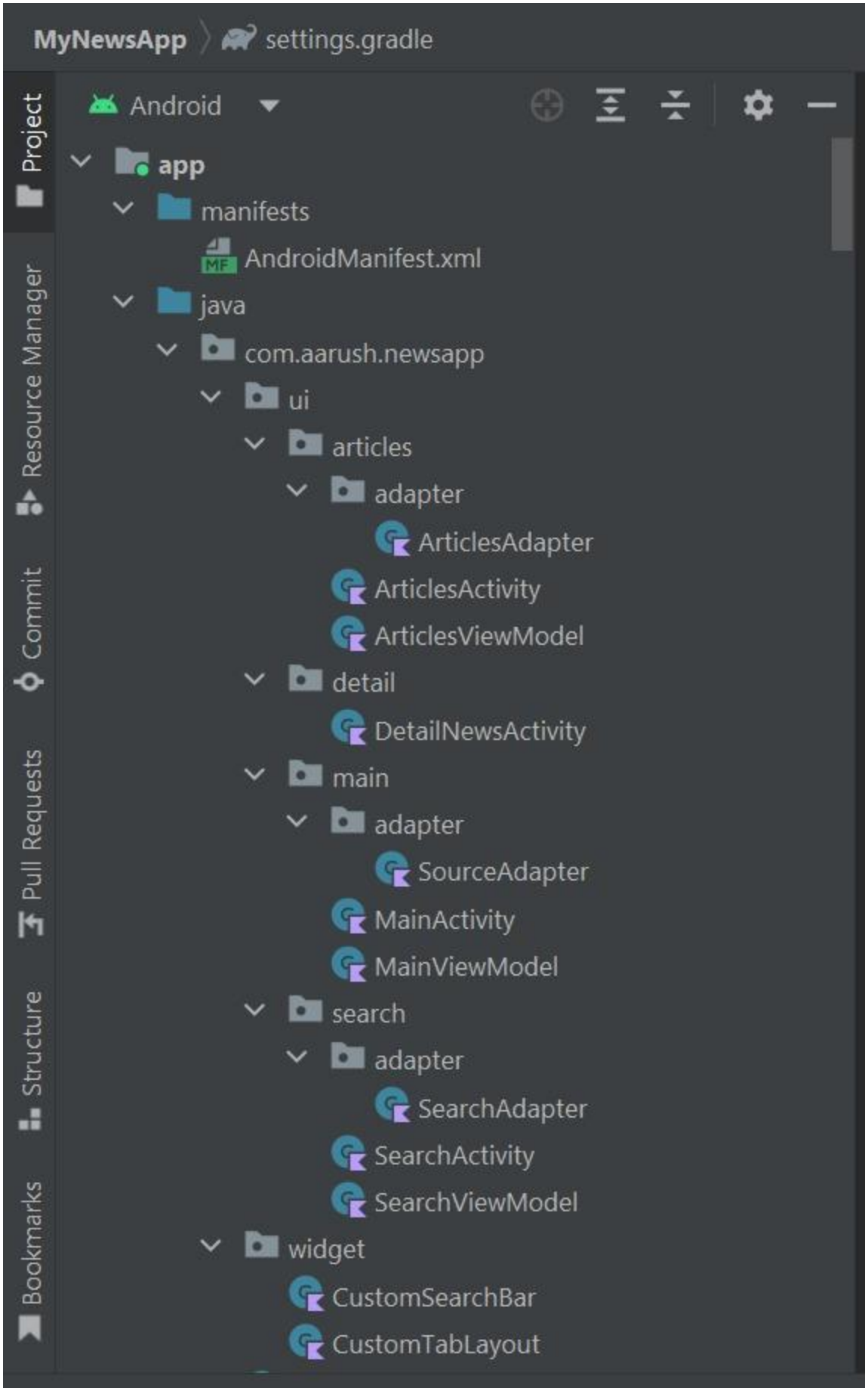
TASK 8: Construct the logic for the application and establish connections with the database

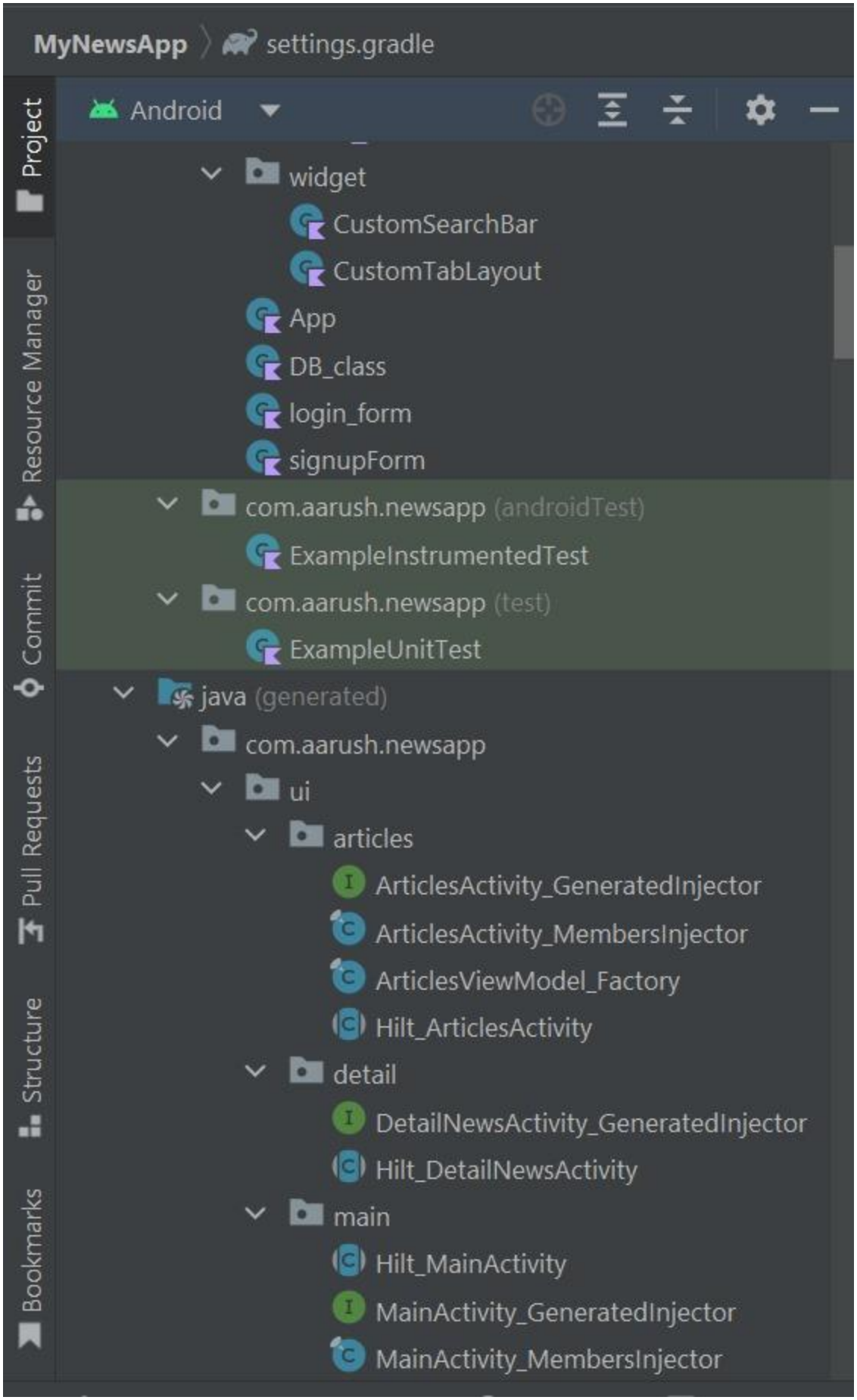
1. Use Kotlin files to construct logic for each screen.
2. Right click on the desired location (folder), click on New and click on Kotlin file.
3. Code the logic that would be used for each screen and concept of the application.

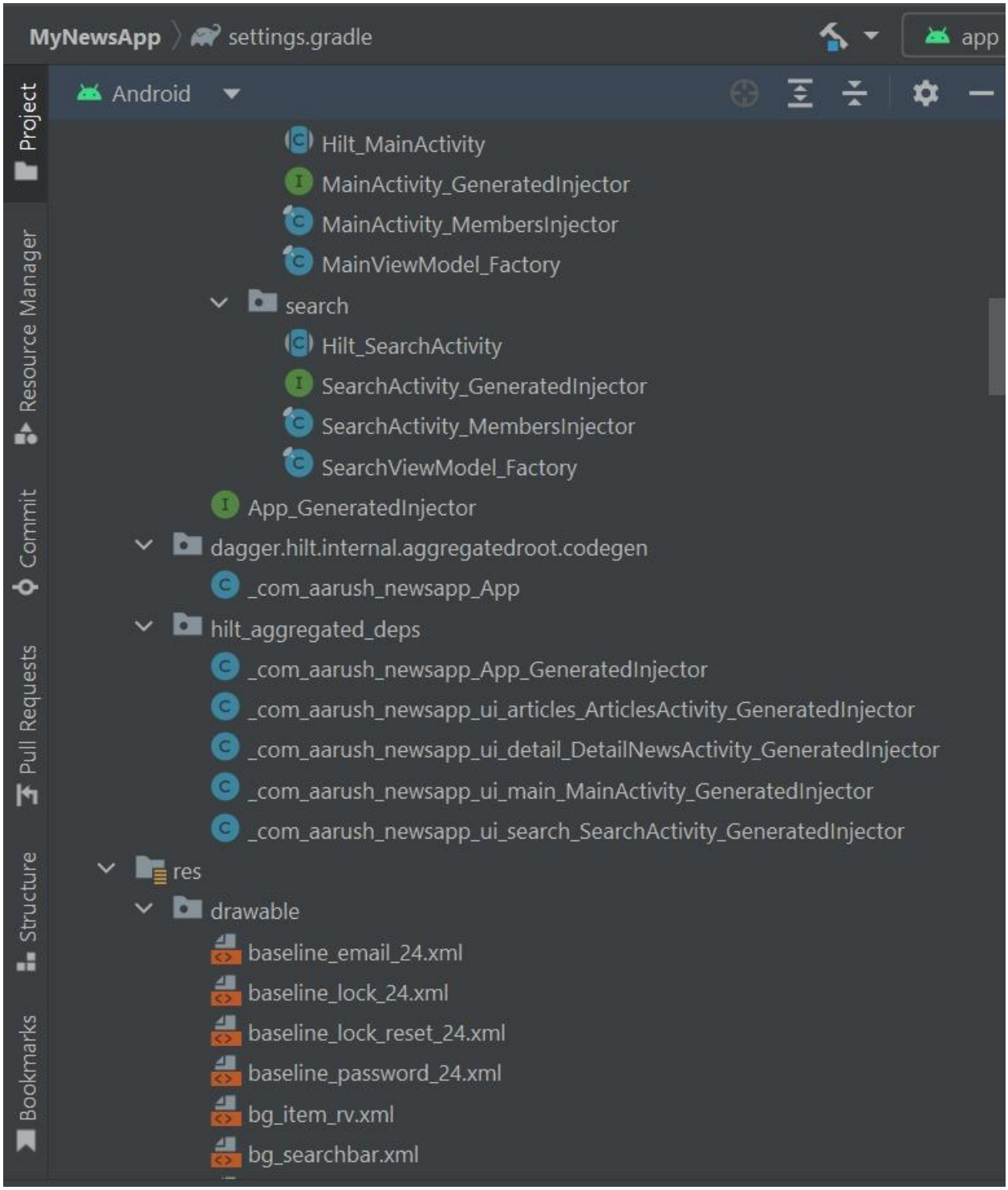
TASK 9: Execute the application to test it

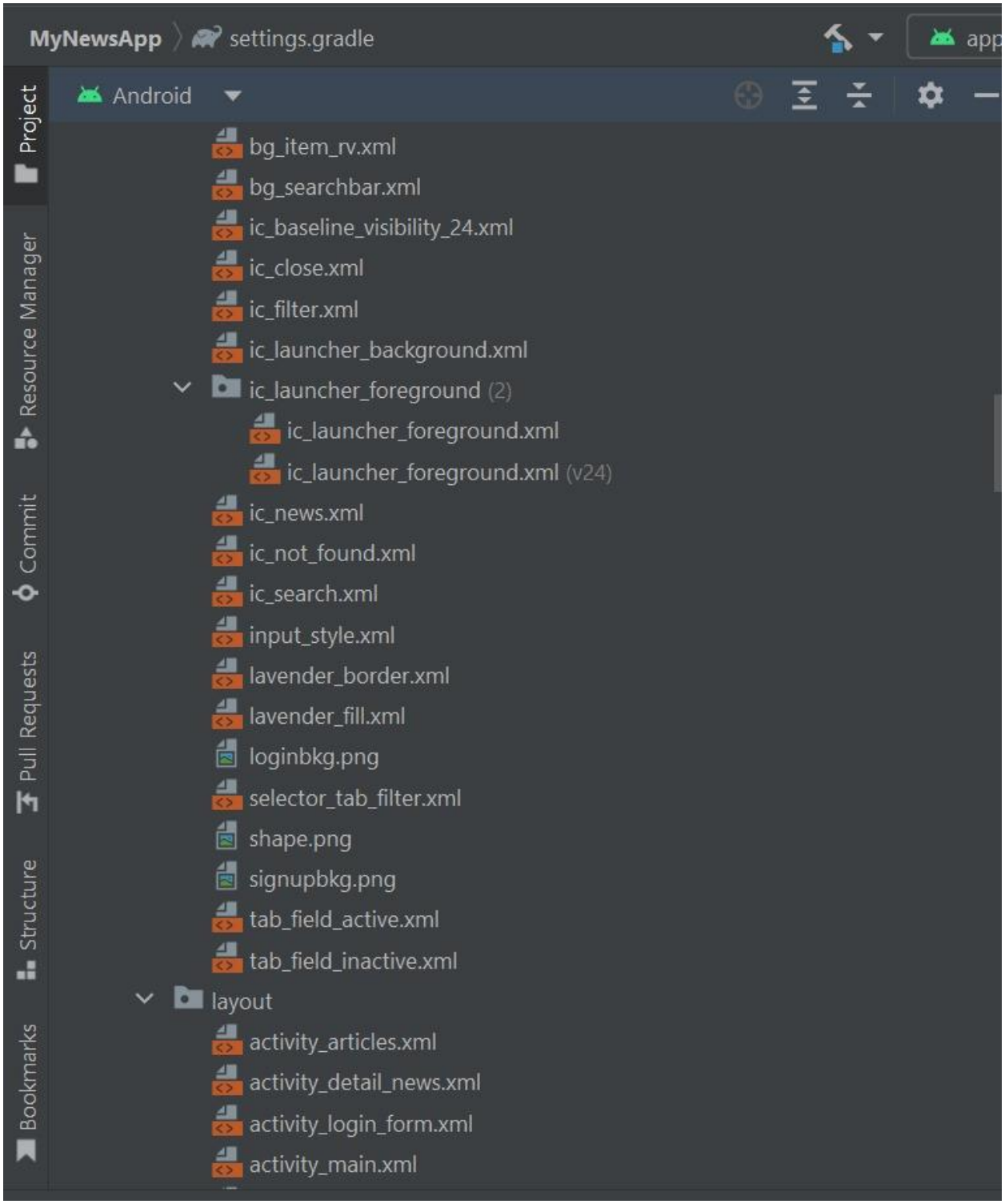
1. Create or find the right emulator.
2. Build and run the application.

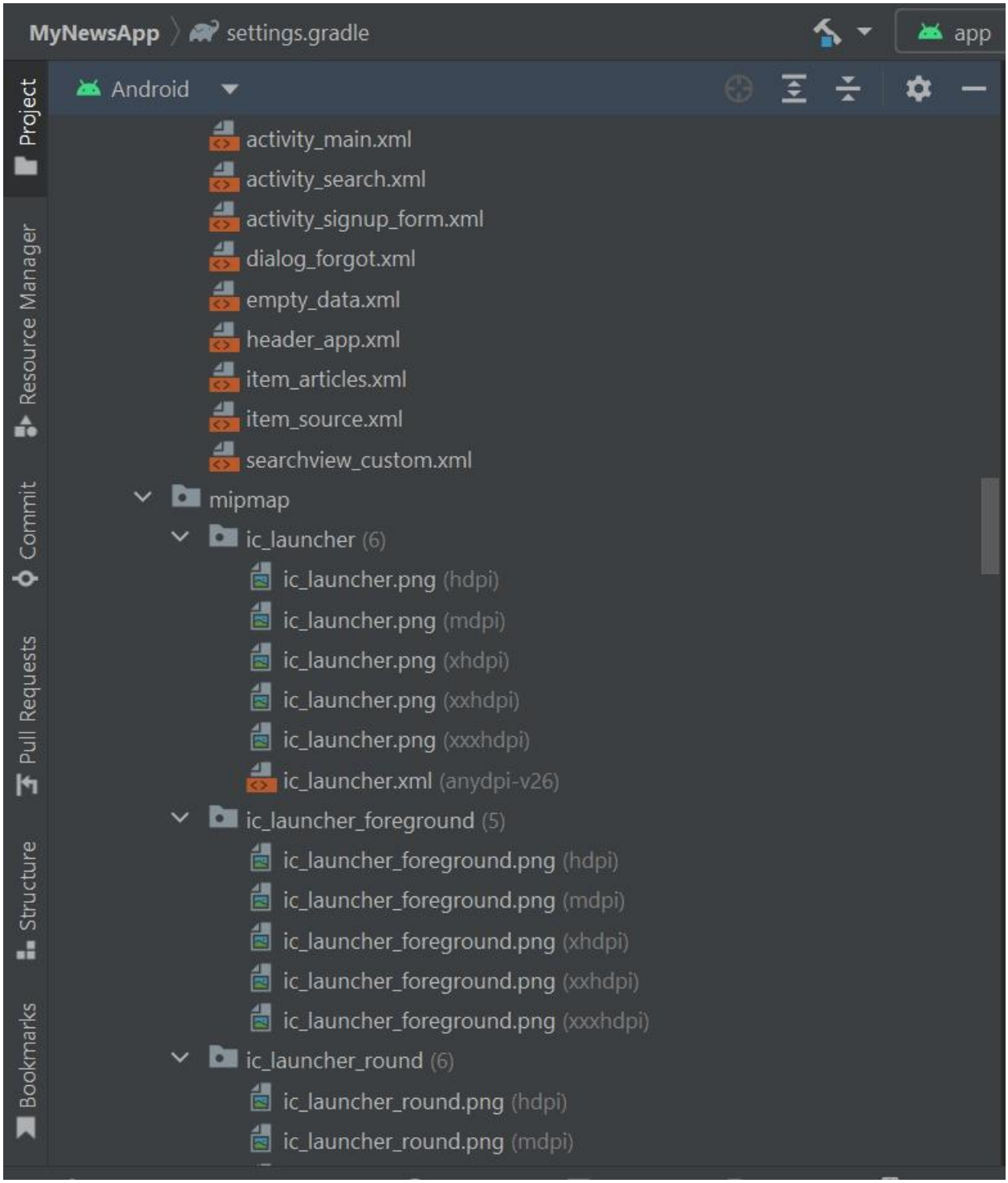


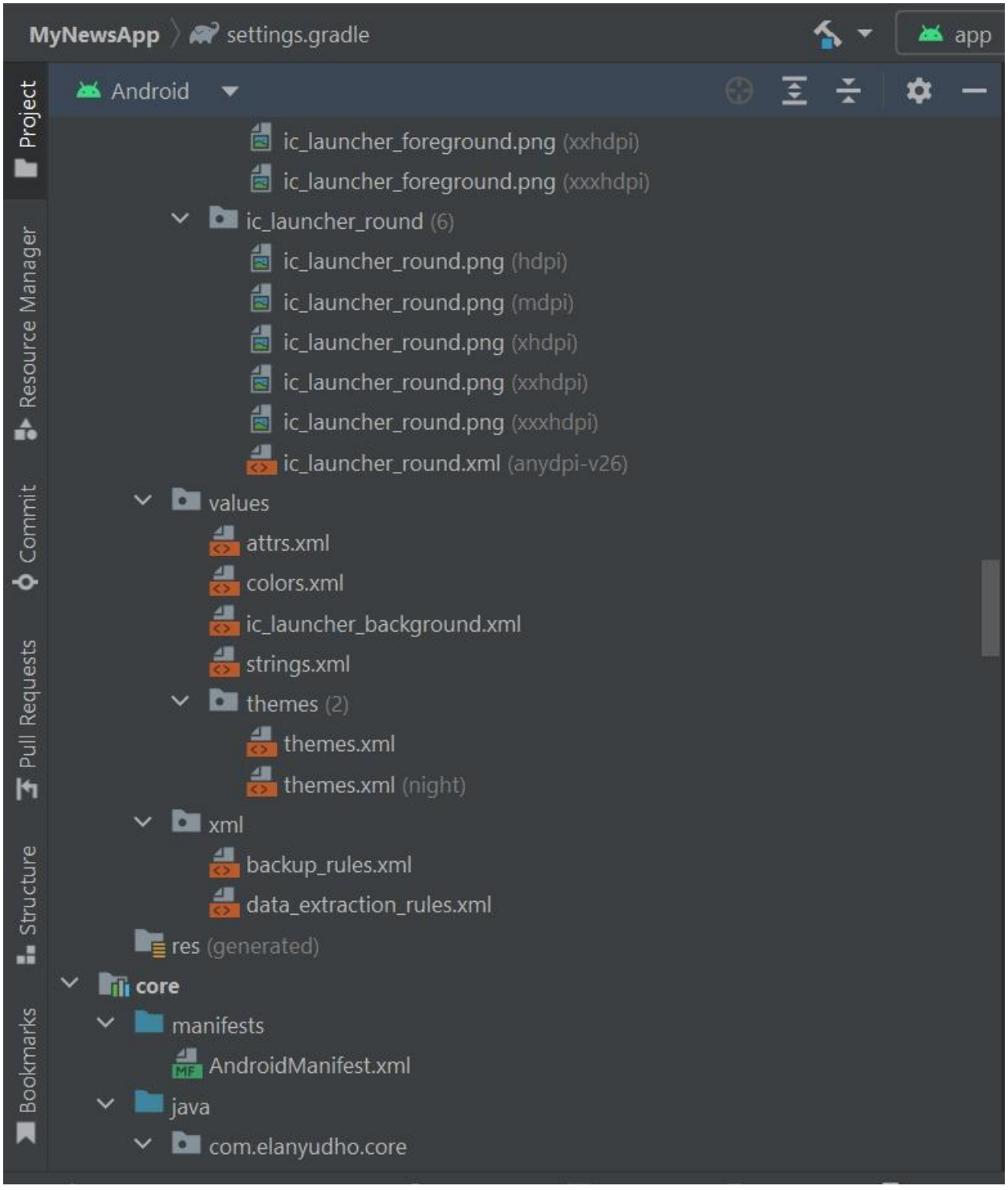

MyNewsApp
settings.gradle
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
app
manifests
AndroidManifest.xml
java
com.aarush.newsapp
ui
articles
adapter
ArticlesAdapter
ArticlesActivity
ArticlesViewModel
detail
DetailNewsActivity
main
adapter
SourceAdapter
MainActivity
MainViewModel
search
adapter
SearchAdapter
SearchActivity
SearchViewModel
widget
CustomSearchBar
CustomTabLayout

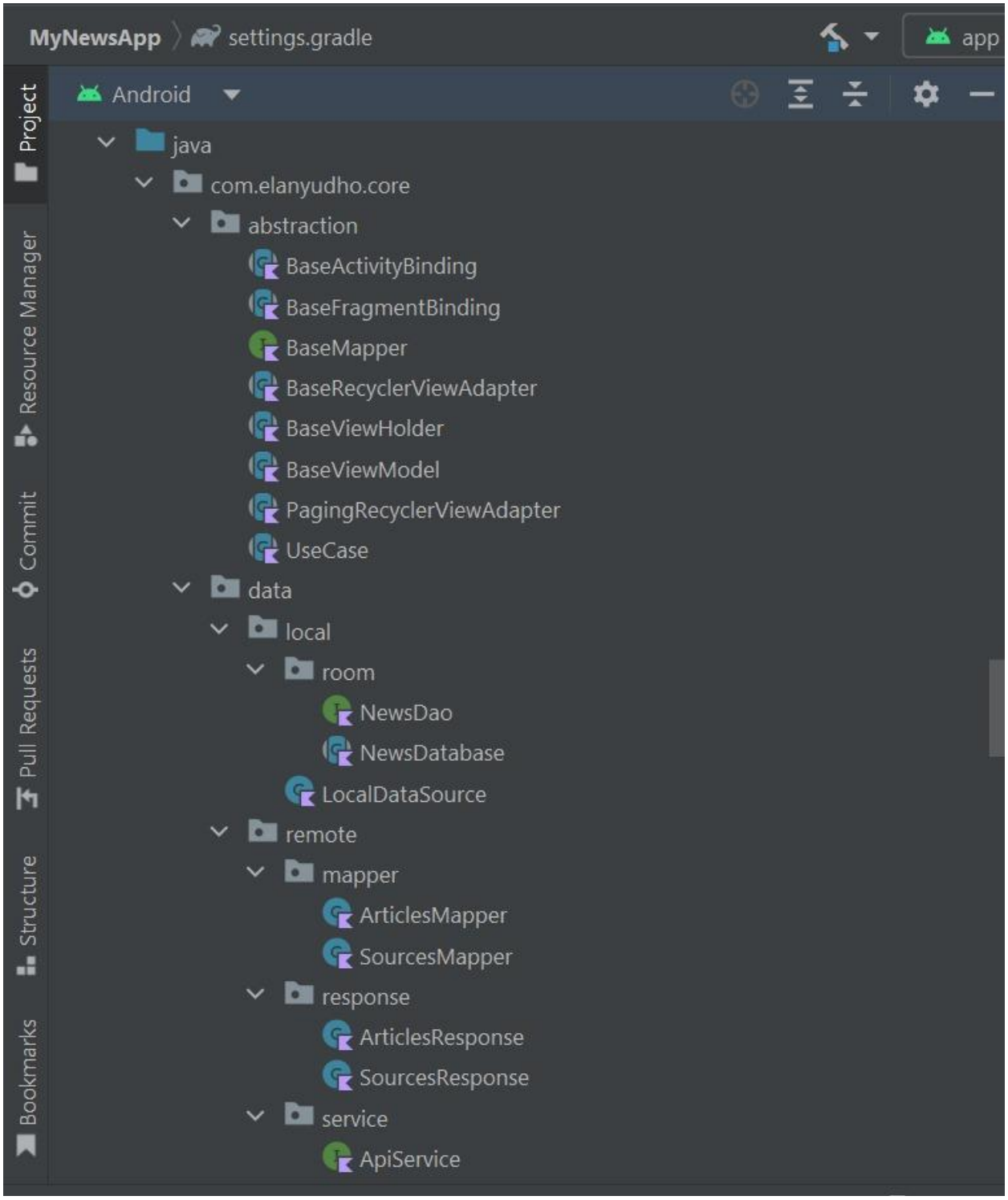

MyNewsApp
settings.gradle
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
widget
CustomSearchBar
CustomTabLayout
App
DB_class
login_form
signupForm
com.aarush.newsapp (androidTest)
ExampleInstrumentedTest
com.aarush.newsapp (test)
ExampleUnitTest
java (generated)
com.aarush.newsapp
ui
articles
ArticlesActivity_GeneratedInjector
ArticlesActivity_MembersInjector
ArticlesViewModel_Factory
Hilt_ArticlesActivity
detail
DetailNewsActivity_GeneratedInjector
Hilt_DetailNewsActivity
main
Hilt_MainActivity
MainActivity_GeneratedInjector
MainActivity_MembersInjector

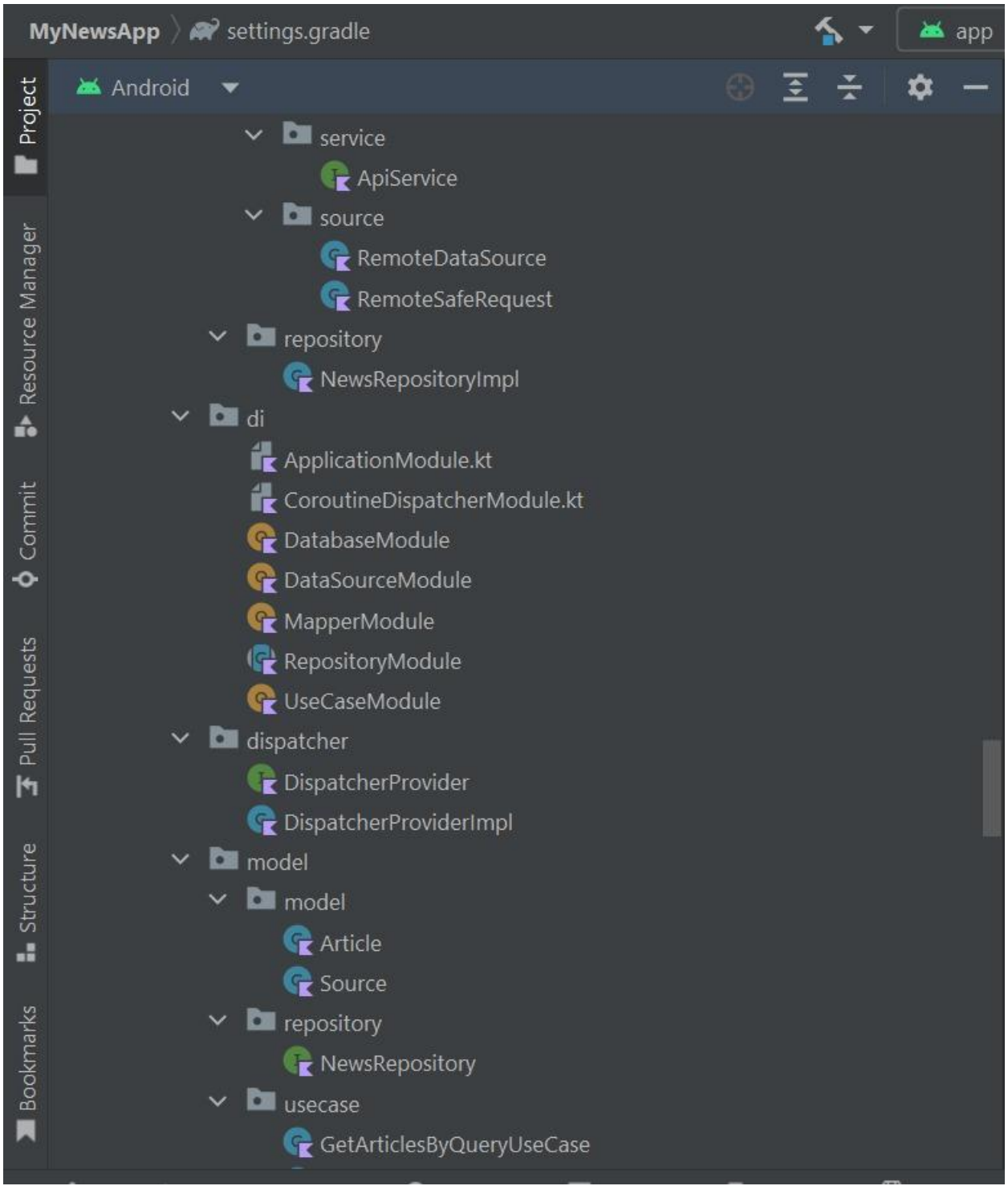

MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
Hilt_MainActivity
MainActivity_GeneratedInjector
MainActivity_MembersInjector
MainViewModel_Factory
search
Hilt_SearchActivity
SearchActivity_GeneratedInjector
SearchActivity_MembersInjector
SearchViewModel_Factory
App_GeneratedInjector
dagger.hilt.internal.aggregatedroot.codegen
_com_aarush_newsapp_App
hilt_aggregated_deps
_com_aarush_newsapp_App_GeneratedInjector
_com_aarush_newsapp_ui_articles_ArticlesActivity_GeneratedInjector
_com_aarush_newsapp_ui_detail_DetailNewsActivity_GeneratedInjector
_com_aarush_newsapp_ui_main_MainActivity_GeneratedInjector
_com_aarush_newsapp_ui_search_SearchActivity_GeneratedInjector
res
drawable
baseline_email_24.xml
baseline_lock_24.xml
baseline_lock_reset_24.xml
baseline_password_24.xml
bg_item_rv.xml
bg_searchbar.xml

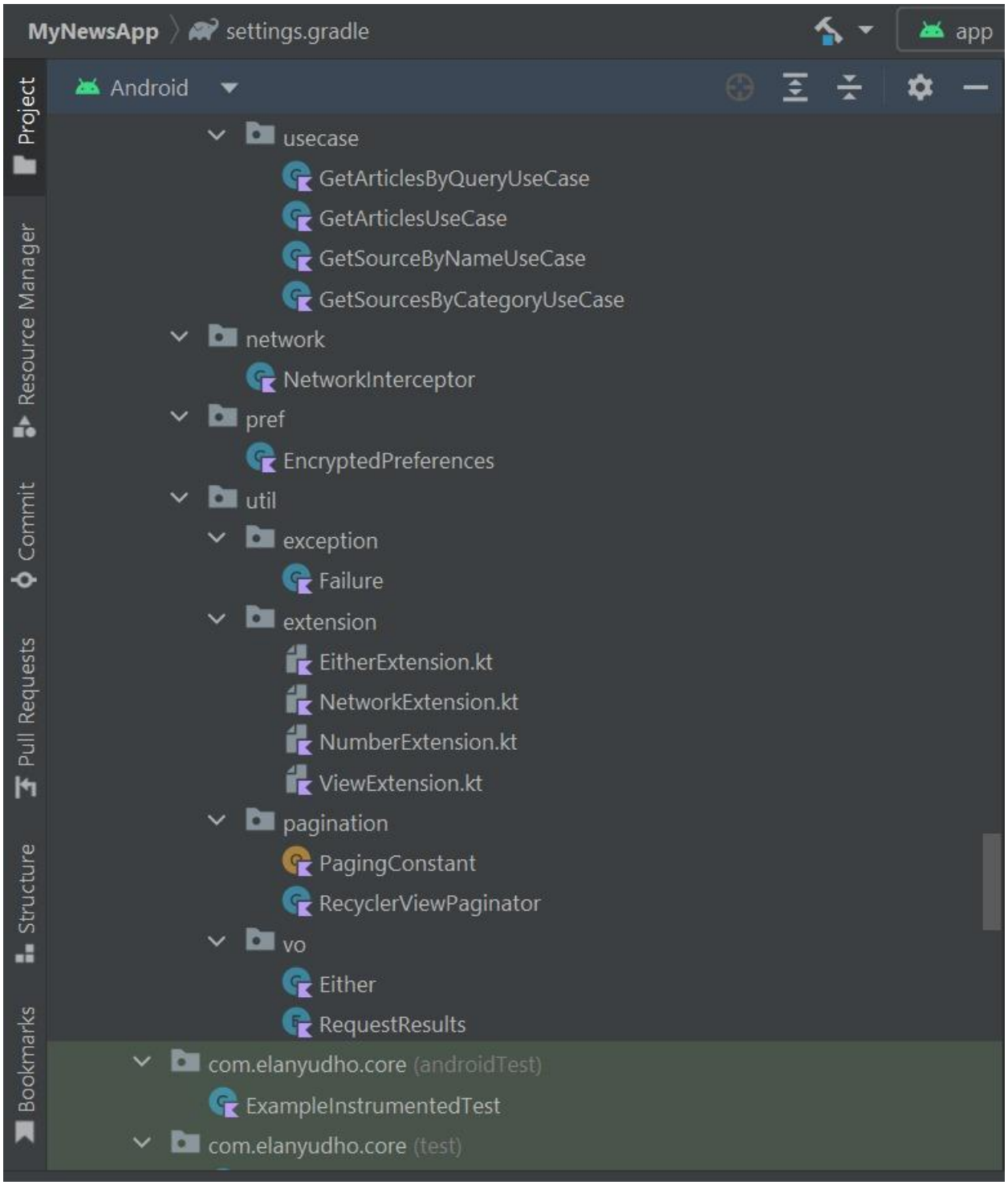

MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
bg_item_rv.xml
bg_searchbar.xml
ic_baseline_visibility_24.xml
ic_close.xml
ic_filter.xml
ic_launcher_background.xml
ic_launcher_foreground (2)
ic_launcher_foreground.xml
ic_launcher_foreground.xml (v24)
ic_news.xml
ic_not_found.xml
ic_search.xml
input_style.xml
lavender_border.xml
lavender_fill.xml
loginbkg.png
selector_tab_filter.xml
shape.png
signupbkg.png
tab_field_active.xml
tab_field_inactive.xml
layout
activity_articles.xml
activity_detail_news.xml
activity_login_form.xml
activity_main.xml

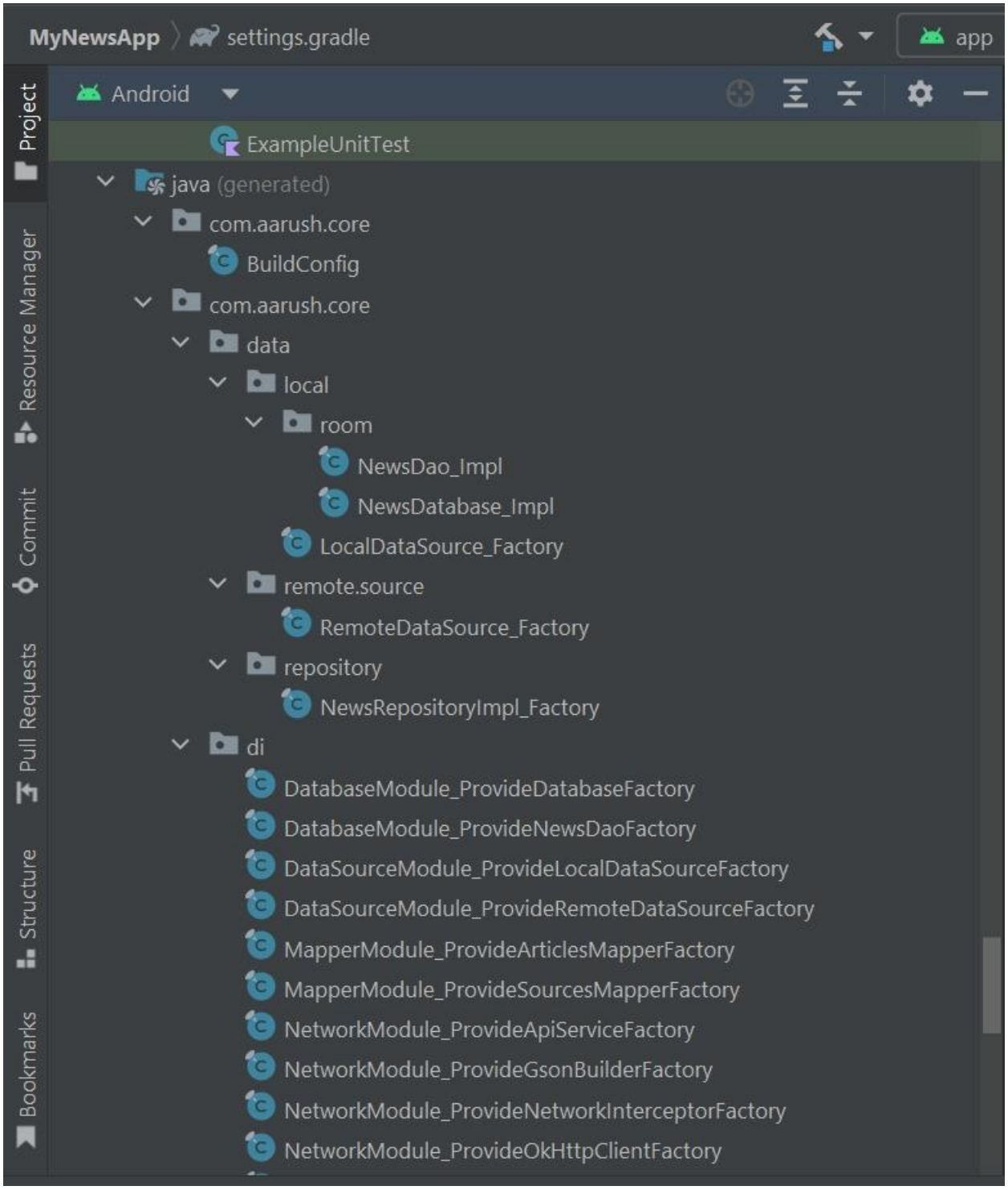

MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
activity_main.xml
activity_search.xml
activity_signup_form.xml
dialog_forgot.xml
empty_data.xml
header_app.xml
item_articles.xml
item_source.xml
searchview_custom.xml
mipmap
ic_launcher (6)
ic_launcher.png (hdpi)
ic_launcher.png (mdpi)
ic_launcher.png (xhdpi)
ic_launcher.png (xxhdpi)
ic_launcher.png (xxxhdpi)
ic_launcher.xml (anydpi-v26)
ic_launcher_foreground (5)
ic_launcher_foreground.png (hdpi)
ic_launcher_foreground.png (mdpi)
ic_launcher_foreground.png (xhdpi)
ic_launcher_foreground.png (xxhdpi)
ic_launcher_foreground.png (xxxhdpi)
ic_launcher_round (6)
ic_launcher_round.png (hdpi)
ic_launcher_round.png (mdpi)

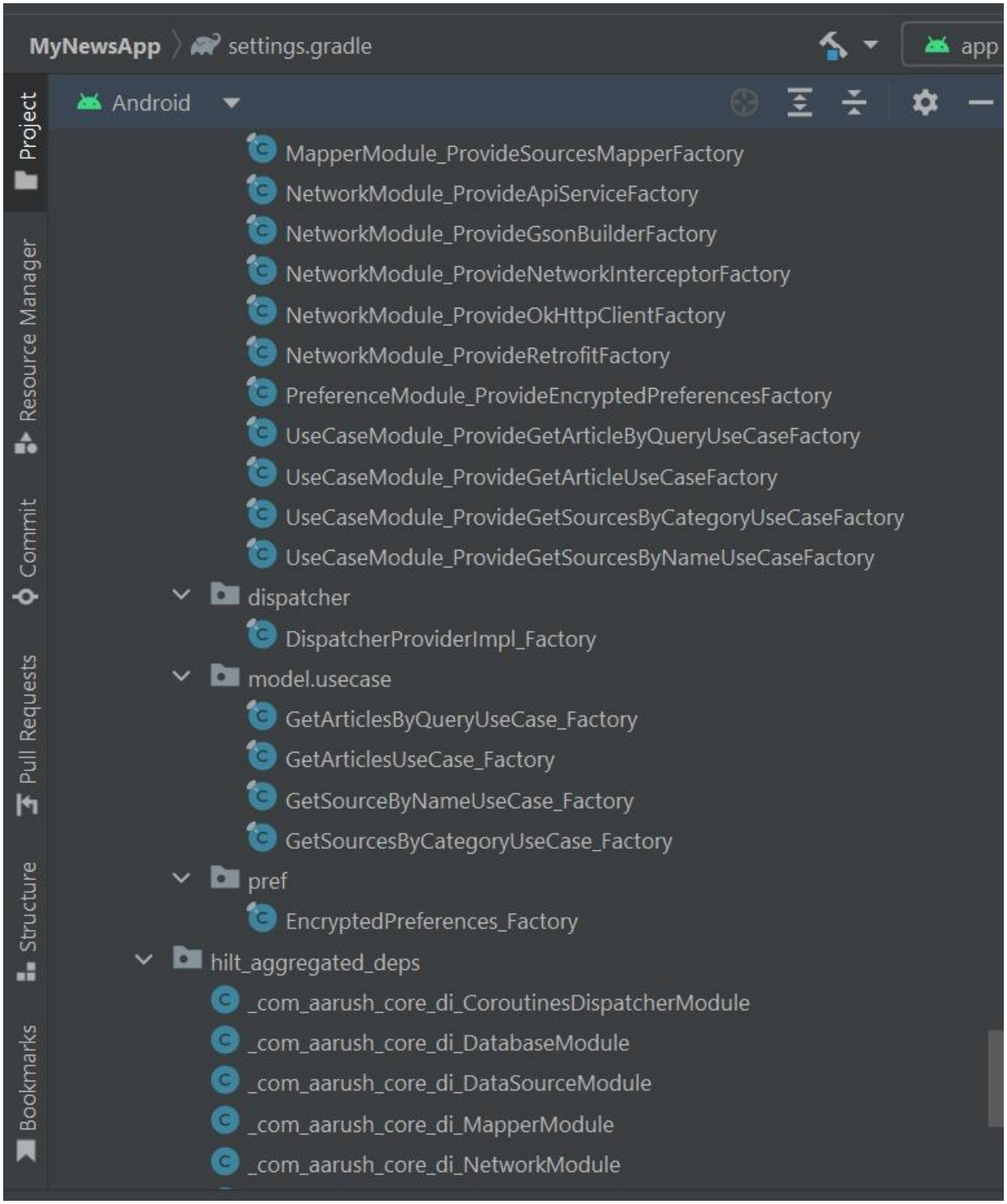

MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
ic_launcher_foreground.png (xxhdpi)
ic_launcher_foreground.png (xxxhdpi)
ic_launcher_round (6)
ic_launcher_round.png (hdpi)
ic_launcher_round.png (mdpi)
ic_launcher_round.png (xhdpi)
ic_launcher_round.png (xxhdpi)
ic_launcher_round.png (xxxhdpi)
ic_launcher_round.xml (anydpi-v26)
values
attrs.xml
colors.xml
ic_launcher_background.xml
strings.xml
themes (2)
themes.xml
themes.xml (night)
xml
backup_rules.xml
data_extraction_rules.xml
res (generated)
core
manifests
AndroidManifest.xml
java
com.elanyudho.core

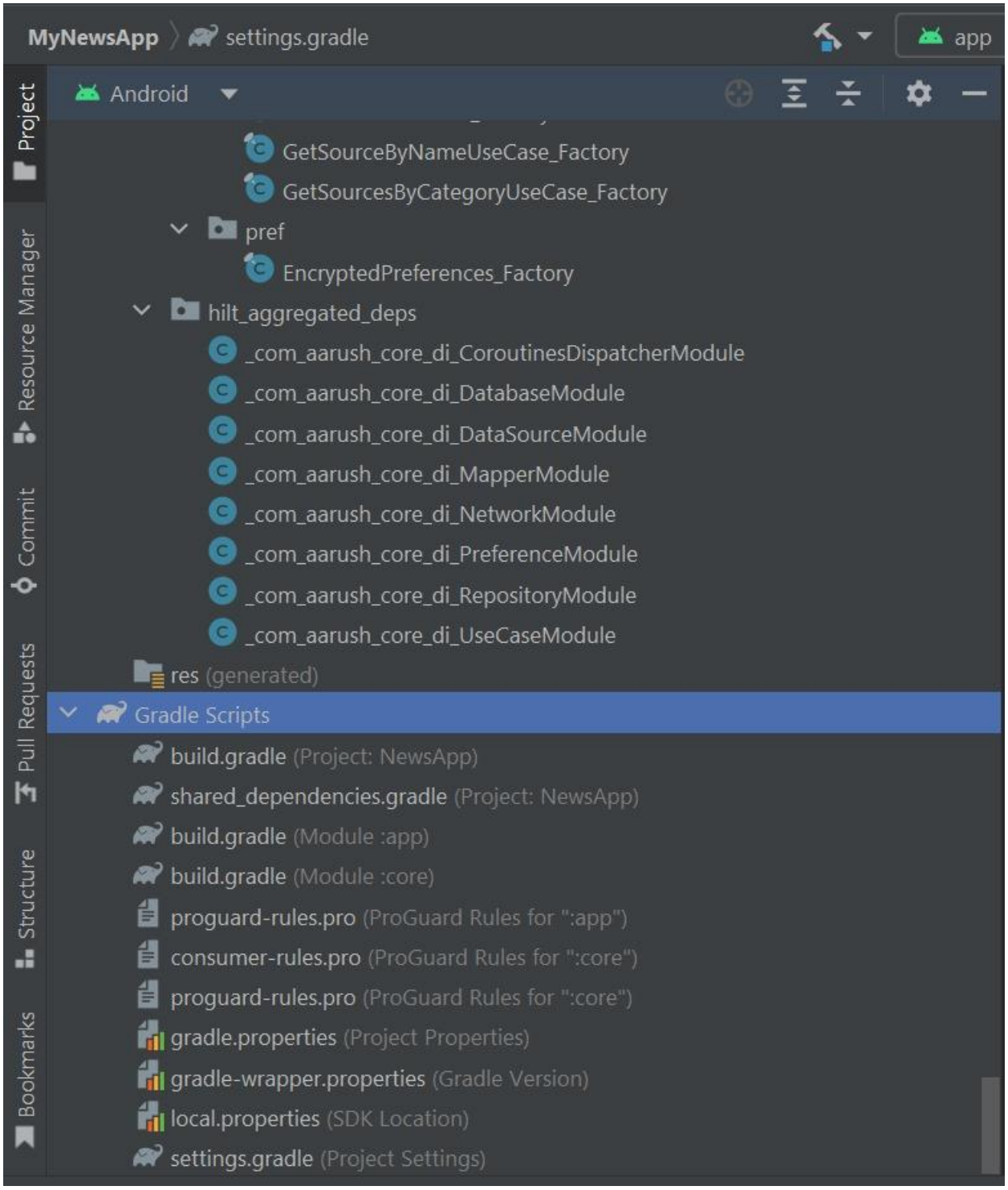

MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
java
com.elanyudho.core
abstraction
BaseActivityBinding
BaseFragmentBinding
BaseMapper
BaseRecyclerViewAdapter
BaseViewHolder
BaseViewModel
PagingRecyclerViewAdapter
UseCase
data
local
room
NewsDao
NewsDatabase
LocalDataSource
remote
mapper
ArticlesMapper
SourcesMapper
response
ArticlesResponse
SourcesResponse
service
ApiService


MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
service
ApiService
source
RemoteDataSource
RemoteSafeRequest
repository
NewsRepositoryImpl
di
ApplicationModule.kt
CoroutineDispatcherModule.kt
DatabaseModule
DataSourceModule
MapperModule
RepositoryModule
UseCaseModule
dispatcher
DispatcherProvider
DispatcherProviderImpl
model
model
Article
Source
repository
NewsRepository
usecase
GetArticlesByQueryUseCase


MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
usecase
GetArticlesByQueryUseCase
GetArticlesUseCase
GetSourceByNameUseCase
GetSourcesByCategoryUseCase
network
NetworkInterceptor
pref
EncryptedPreferences
util
exception
Failure
extension
EitherExtension.kt
NetworkExtension.kt
NumberExtension.kt
ViewExtension.kt
pagination
PagingConstant
RecyclerViewPaginator
vo
Either
RequestResults
com.elanyudho.core (androidTest)
ExampleInstrumentedTest
com.elanyudho.core (test)


MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
ExampleUnitTest
java (generated)
com.aarush.core
BuildConfig
com.aarush.core
data
local
room
NewsDao_Impl
NewsDatabase_Impl
LocalDataSource_Factory
remote.source
RemoteDataSource_Factory
repository
NewsRepositoryImpl_Factory
di
DatabaseModule_ProvideDatabaseFactory
DatabaseModule_ProvideNewsDaoFactory
DataSourceModule_ProvideLocalDataSourceFactory
DataSourceModule_ProvideRemoteDataSourceFactory
MapperModule_ProvideArticlesMapperFactory
MapperModule_ProvideSourcesMapperFactory
NetworkModule_ProvideApiServiceFactory
NetworkModule_ProvideGsonBuilderFactory
NetworkModule_ProvideNetworkInterceptorFactory
NetworkModule_ProvideOkHttpClientFactory

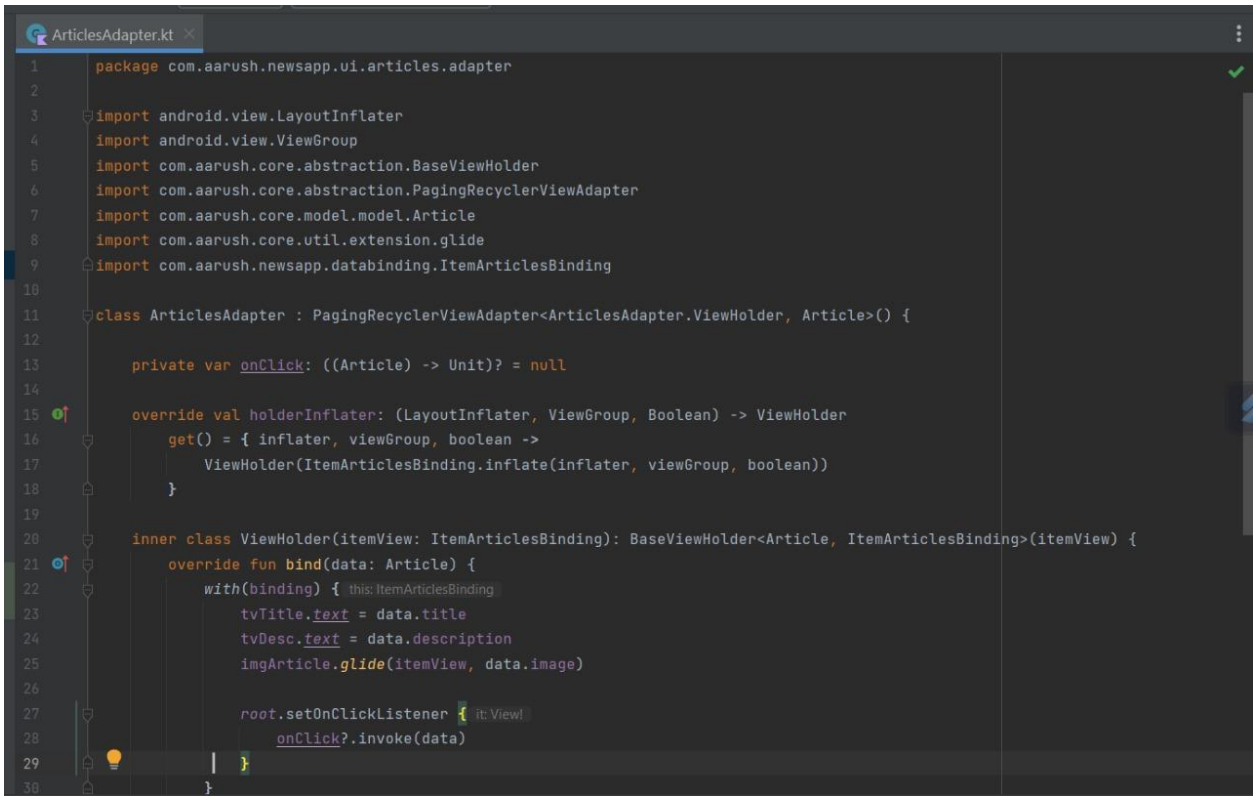

MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
MapperModule_ProvideSourcesMapperFactory
NetworkModule_ProvideApiServiceFactory
NetworkModule_ProvideGsonBuilderFactory
NetworkModule_ProvideNetworkInterceptorFactory
NetworkModule_ProvideOkHttpClientFactory
NetworkModule_ProvideRetrofitFactory
PreferenceModule_ProvideEncryptedPreferencesFactory
UseCaseModule_ProvideGetArticleByQueryUseCaseFactory
UseCaseModule_ProvideGetArticleUseCaseFactory
UseCaseModule_ProvideGetSourcesByCategoryUseCaseFactory
UseCaseModule_ProvideGetSourcesByNameUseCaseFactory
dispatcher
DispatcherProviderImpl_Factory
model.usecase
GetArticlesByQueryUseCase_Factory
GetArticlesUseCase_Factory
GetSourceByNameUseCase_Factory
GetSourcesByCategoryUseCase_Factory
pref
EncryptedPreferences_Factory
hilt_aggregated_deps
_com_aarush_core_di_CoroutinesDispatcherModule
_com_aarush_core_di_DatabaseModule
_com_aarush_core_di_DataSourceModule
_com_aarush_core_di_MapperModule
_com_aarush_core_di_NetworkModule


MyNewsApp
settings.gradle
app
Android
Project
Resource Manager
Commit
Pull Requests
Structure
Bookmarks
GetSourceByNameUseCase_Factory
GetSourcesByCategoryUseCase_Factory
pref
EncryptedPreferences_Factory
hilt_aggregated_deps
_com_aarush_core_di_CoroutinesDispatcherModule
_com_aarush_core_di_DatabaseModule
_com_aarush_core_di_DataSourceModule
_com_aarush_core_di_MapperModule
_com_aarush_core_di_NetworkModule
_com_aarush_core_di_PreferenceModule
_com_aarush_core_di_RepositoryModule
_com_aarush_core_di_UseCaseModule
res (generated)
Gradle Scripts
build.gradle (Project: NewsApp)
shared_dependencies.gradle (Project: NewsApp)
build.gradle (Module :app)
build.gradle (Module :core)
proguard-rules.pro (ProGuard Rules for ":app")
consumer-rules.pro (ProGuard Rules for ":core")
proguard-rules.pro (ProGuard Rules for ":core")
gradle.properties (Project Properties)
gradle-wrapper.properties (Gradle Version)
local.properties (SDK Location)
settings.gradle (Project Settings)

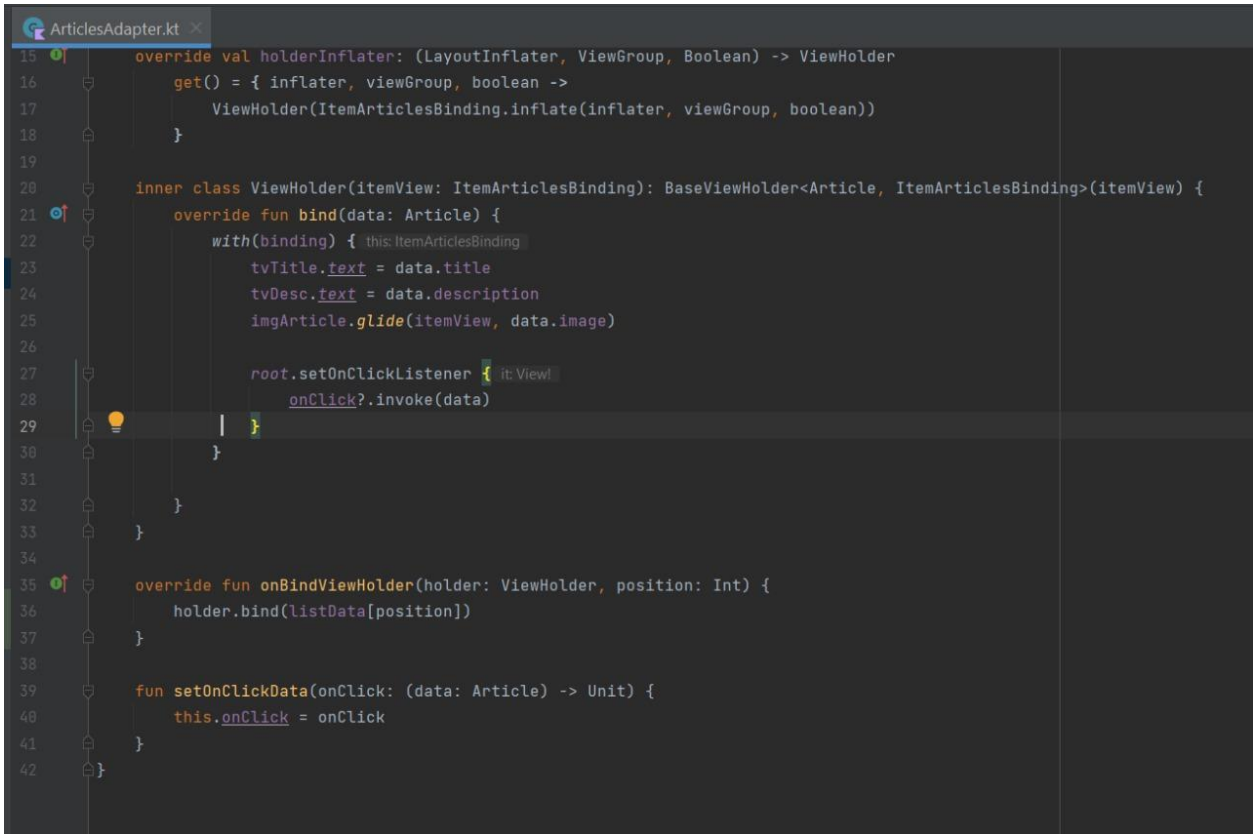
CODES:

1. **ArticlesAdapter.kt**
2. **ArticlesActivity.kt**
3. **ArticlesViewModel.kt**
4. **DetailNewsActivity.kt**
5. **SourceAdapter.kt**
6. **MainActivity.kt**
7. **MainViewModel.kt**
8. **SearchAdapter.kt**
9. **SearchActivity.kt**
10. **SearchViewModel.kt**
11. **App.kt**
12. **DB_class.kt**
13. **login_form.kt**
14. **signupForm.kt**
15. **CustomSearchBar.kt**
16. **CustomTabLayout.kt**
17. **ExampleUnitTest.kt**
18. **activity_articles.xml**
19. **activity_detail_news.xml**
20. **activity_login_form.xml**
21. **activity_main.xml**
22. **activity_search.xml**
23. **activity_signup_form.xml**
24. **empty_data.xml**
25. **header_app.xml**
26. **item.xml**
27. **searchview_custom.xml**

CODE 1: ArticlesAdapter.kt



```kotlin
package com.aarush.newsapp.ui.articles.adapter

import android.view.LayoutInflater
import android.view.ViewGroup
import com.aarush.core.abstraction.BaseViewHolder
import com.aarush.core.abstraction.PagingRecyclerViewAdapter
import com.aarush.core.model.model.Article
import com.aarush.core.util.extension.glide
import com.aarush.newsapp.databinding.ItemArticlesBinding

class ArticlesAdapter : PagingRecyclerViewAdapter<ArticlesAdapter.ViewHolder, Article>() {

    private var onClick: ((Article) -> Unit)? = null

    override val holderInflater: (LayoutInflater, ViewGroup, Boolean) -> ViewHolder
        get() = { inflater, viewGroup, boolean ->
            ViewHolder(ItemArticlesBinding.inflate(inflater, viewGroup, boolean))
        }

    inner class ViewHolder(itemView: ItemArticlesBinding): BaseViewHolder<Article, ItemArticlesBinding>(itemView) {
        override fun bind(data: Article) {
            with(binding) { this: ItemArticlesBinding
                tvTitle.text = data.title
                tvDesc.text = data.description
                imgArticle.glide(itemView, data.image)

                root.setOnClickListener { it: View!
                    onClick?.invoke(data)
                }
            }
```



CODE 2: ArticlesActivity.kt

ArticlesActivity.kt

```kotlin
package com.aarush.newsapp.ui.articles

import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.widget.Toast
import androidx.lifecycle.Observer
import androidx.recyclerview.widget.LinearLayoutManager
import com.aarush.core.abstraction.BaseActivityBinding
import com.aarush.core.util.exception.Failure
import com.aarush.core.util.pagination.RecyclerViewPaginator
import com.aarush.newsapp.R
import com.aarush.newsapp.databinding.ActivityArticlesBinding
import com.aarush.newsapp.ui.articles.adapter.ArticlesAdapter
import com.aarush.newsapp.ui.detail.DetailNewsActivity
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class ArticlesActivity : BaseActivityBinding<ActivityArticlesBinding>(), Observer<ArticlesViewModel.ArticlesUiState> {

    @Inject
    lateinit var articlesViewModel: ArticlesViewModel

    private val articlesAdapter: ArticlesAdapter by lazy { ArticlesAdapter() }

    private var paginator: RecyclerViewPaginator? = null

    private lateinit var sourceId: String
```

ArticlesActivity.kt

```kotlin
    private lateinit var sourceId: String

    override val bindingInflater: (LayoutInflater) -> ActivityArticlesBinding
        get() = { ActivityArticlesBinding.inflate(layoutInflater) }

    override fun setupView() {
        getDataIntent()
        initViewModel()
        setHeader()
        setAdapter()
        setPagination()

    }

    override fun onChanged(state: ArticlesViewModel.ArticlesUiState?) {
        when(state) {
            is ArticlesViewModel.ArticlesUiState.ArticlesLoaded -> {
                stopLoading()
                if (state.data.isEmpty() && paginator?.isFirstGet == true) {
                    showEmptyData()
                }else {
                    hideEmptyData()
                    articlesAdapter.appendList(state.data)
                }
            }
            is ArticlesViewModel.ArticlesUiState.InitialLoading -> {
                initialLoading()
            }
            is ArticlesViewModel.ArticlesUiState.PagingLoading -> {
```

ArticlesActivity.kt

```
                initialLoading()
            }
            is ArticlesViewModel.ArticlesUiState.PagingLoading -> {
                pagingLoading()
            }
            is ArticlesViewModel.ArticlesUiState.FailedLoaded -> {
                stopLoading()
                handleFailure(state.failure)
            }
            else -> {}
        }
    }

    private fun getDataIntent() {
        sourceId = intent.getStringExtra( name: "sourceId") ?: ""
    }

    private fun initViewModel() {
        articlesViewModel.uiState.observe( owner: this,  observer: this)
        articlesViewModel.getArticles(sourceId,  page: 1)
    }

    private fun setHeader() {
        with(binding) { this: ActivityArticlesBinding
            headerDetail.tvHeader.text = "Articles"
            headerDetail.btnBack.setOnClickListener { onBackPressedDispatcher.onBackPressed() }
        }
    }

    private fun setAdapter() {
```

ArticlesActivity.kt

```
    }

    private fun setAdapter() {
        with(binding.rvNews) { this: RecyclerView
            adapter = articlesAdapter
            setHasFixedSize(true)

            articlesAdapter.setOnClickData { it: Article
                val intent = Intent( packageContext: this@ArticlesActivity, DetailNewsActivity::class.java)
                intent.putExtra( name: "sourceNews", it.source)
                intent.putExtra( name: "urlNews", it.url)
                startActivity(intent)
            }
        }
    }

    private fun setPagination() {
        paginator = RecyclerViewPaginator(binding.rvNews.layoutManager as LinearLayoutManager)
        paginator?.setOnLoadMoreListener { page ->
            paginator?.isFirstGet = false
            articlesViewModel.getArticles(sourceId, page)
        }
        paginator?.let { binding.rvNews.addOnScrollListener(it) }
    }
    private fun initialLoading() {
        binding.progressCircular.visibility = View.VISIBLE
        binding.rvNews.visibility = View.GONE
    }

    private fun pagingLoading() {
```

ArticlesActivity.kt

```
        paginator?.let { binding.rvNews.addOnScrollListener(it) }
    }
    private fun initialLoading() {
        binding.progressCircular.visibility = View.VISIBLE
        binding.rvNews.visibility = View.GONE
    }

    private fun pagingLoading() {
        binding.progressCircular.visibility = View.VISIBLE
    }

    private fun stopLoading() {
        binding.rvNews.visibility = View.VISIBLE
        binding.progressCircular.visibility = View.GONE
    }

    private fun showEmptyData() {
        binding.emptyDataView.parent.visibility = View.VISIBLE
    }

    private fun hideEmptyData() {
        binding.emptyDataView.parent.visibility = View.GONE
    }

    private fun handleFailure(failure: Failure) {
        Toast.makeText( context: this, failure.throwable.message, Toast.LENGTH_SHORT).show()
    }
}
```

CODE 3: ArticlesViewModel.kt

ArticlesViewModel.kt

```
package com.aarush.newsapp.ui.articles

import androidx.lifecycle.viewModelScope
import com.aarush.core.abstraction.BaseViewModel
import com.aarush.core.dispatcher.DispatcherProvider
import com.aarush.core.model.model.Article
import com.aarush.core.model.usecase.GetArticlesUseCase
import com.aarush.core.util.exception.Failure
import com.aarush.core.util.extension.onError
import com.aarush.core.util.extension.onSuccess
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import javax.inject.Inject

class ArticlesViewModel @Inject constructor(
    private val dispatcherProvider: DispatcherProvider,
    private val getArticlesUseCase: GetArticlesUseCase,
) : BaseViewModel<ArticlesViewModel.ArticlesUiState>() {

    sealed class ArticlesUiState {
        object InitialLoading: ArticlesUiState()
        object PagingLoading: ArticlesUiState()
        data class ArticlesLoaded(val data: List<Article>) : ArticlesUiState()
        data class FailedLoaded(val failure: Failure) : ArticlesUiState()
    }

    fun getArticles(source: String, page: Long){
        _uiState.value = if (page == 1L) {
            ArticlesUiState.InitialLoading
        }else{
```

```kotlin
ArticlesViewModel.kt
        object InitialLoading: ArticlesUiState()
        object PagingLoading: ArticlesUiState()
        data class ArticlesLoaded(val data: List<Article>) : ArticlesUiState()
        data class FailedLoaded(val failure: Failure) : ArticlesUiState()
    }

    fun getArticles(source: String, page: Long){
        _uiState.value = if (page == 1L) {
            ArticlesUiState.InitialLoading
        }else{
            ArticlesUiState.PagingLoading
        }
        viewModelScope.launch(dispatcherProvider.io) { this: CoroutineScope
            getArticlesUseCase.run(GetArticlesUseCase.Params(source, page.toString()))
                .onSuccess { it: List<Article>
                    withContext(dispatcherProvider.main) { this: CoroutineScope
                        _uiState.value = ArticlesUiState.ArticlesLoaded(it)
                    }
                }
                .onError { it: Failure
                    withContext(dispatcherProvider.main) { this: CoroutineScope
                        _uiState.value = ArticlesUiState.FailedLoaded(it)
                    }
                }
        }
    }

}
```

CODE 4: DetailNewsActivity.kt

```kotlin
DetailNewsActivity.kt
package com.aarush.newsapp.ui.detail

import android.annotation.SuppressLint
import android.view.LayoutInflater
import android.view.View
import android.webkit.WebChromeClient
import android.webkit.WebSettings
import android.webkit.WebView
import androidx.lifecycle.lifecycleScope
import com.aarush.core.abstraction.BaseActivityBinding
import com.aarush.newsapp.R
import com.aarush.newsapp.databinding.ActivityDetailNewsBinding
import dagger.hilt.android.AndroidEntryPoint
import kotlinx.coroutines.Dispatchers
import kotlinx.coroutines.launch
import java.lang.ref.WeakReference

@AndroidEntryPoint
class DetailNewsActivity : BaseActivityBinding<ActivityDetailNewsBinding>() {

    private lateinit var urlNews: String
    private lateinit var sourceNews: String

    override val bindingInflater: (LayoutInflater) -> ActivityDetailNewsBinding
        get() = { ActivityDetailNewsBinding.inflate(layoutInflater) }

    override fun setupView() {
        getDataIntent()
        setHeader()
        setWebView()
```

DetailNewsActivity.kt

```kotlin
        getDataIntent()
        setHeader()
        setWebView()
    }

    private fun getDataIntent() {
        urlNews = intent.getStringExtra( name: "urlNews") ?: ""
        sourceNews = intent.getStringExtra( name: "sourceNews") ?: ""
    }

    private fun setHeader() {
        with(binding) { this: ActivityDetailNewsBinding
            headerDetail.tvHeader.text = sourceNews
            headerDetail.btnBack.setOnClickListener { onBackPressedDispatcher.onBackPressed() }
        }
    }

    @SuppressLint("SetJavaScriptEnabled")
    private fun setWebView() {
        val webSettings: WebSettings = binding.webview.settings
        webSettings.javaScriptEnabled = true
        webSettings.mediaPlaybackRequiresUserGesture = false
        binding.webview.webChromeClient = object : WebChromeClient() {
            override fun onProgressChanged(view: WebView?, newProgress: Int) {
                super.onProgressChanged(view, newProgress)
                if (newProgress == 100) {
                    if (!this@DetailNewsActivity.isDestroyed) {
                        binding.progressCircular.visibility = View.GONE
                    }
                }
```

DetailNewsActivity.kt

```kotlin
    }

    private fun setHeader() {
        with(binding) { this: ActivityDetailNewsBinding
            headerDetail.tvHeader.text = sourceNews
            headerDetail.btnBack.setOnClickListener { onBackPressedDispatcher.onBackPressed() }
        }
    }

    @SuppressLint("SetJavaScriptEnabled")
    private fun setWebView() {
        val webSettings: WebSettings = binding.webview.settings
        webSettings.javaScriptEnabled = true
        webSettings.mediaPlaybackRequiresUserGesture = false
        binding.webview.webChromeClient = object : WebChromeClient() {
            override fun onProgressChanged(view: WebView?, newProgress: Int) {
                super.onProgressChanged(view, newProgress)
                if (newProgress == 100) {
                    if (!this@DetailNewsActivity.isDestroyed) {
                        binding.progressCircular.visibility = View.GONE
                    }
                }
            }
        }
        binding.webview.loadUrl(urlNews)
    }

}
```

CODE 5: SourceAdapter.kt

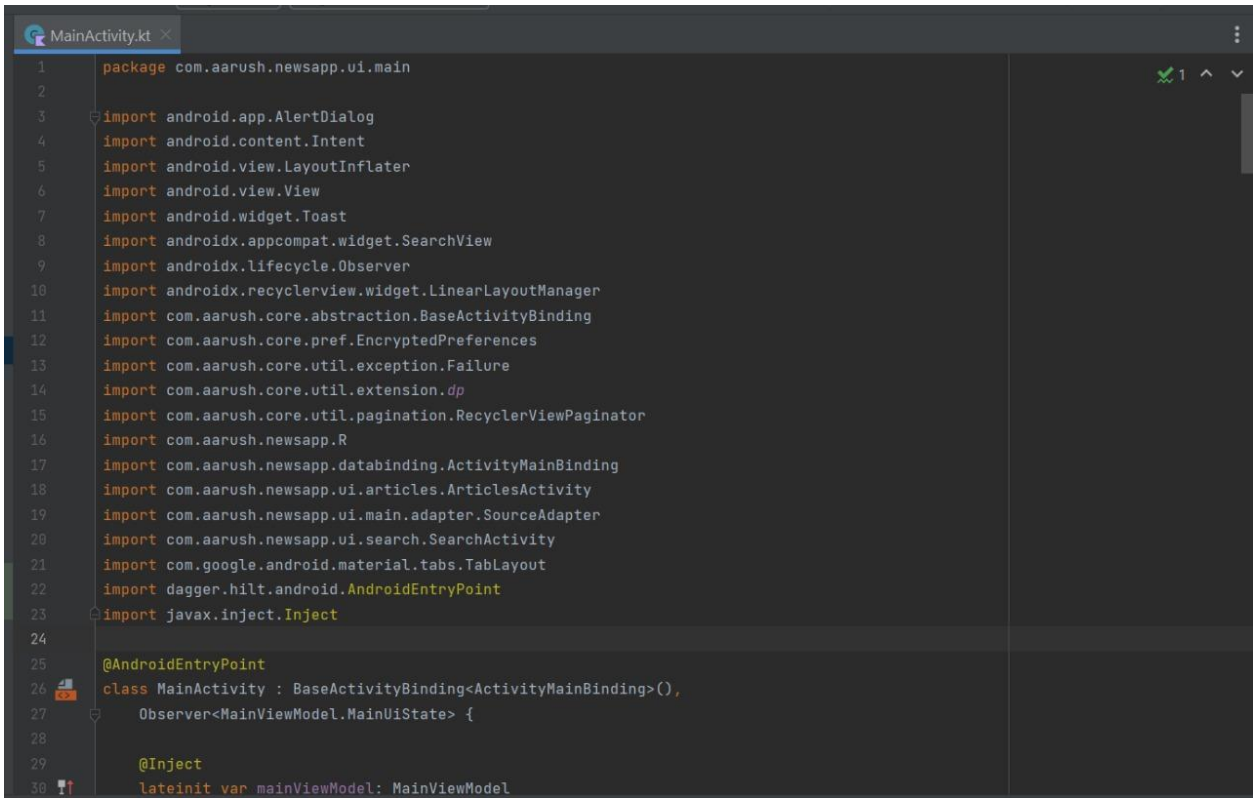
SourceAdapter.kt

```kotlin
package com.aarush.newsapp.ui.main.adapter

import android.view.LayoutInflater
import android.view.ViewGroup
import com.aarush.core.abstraction.BaseViewHolder
import com.aarush.core.abstraction.PagingRecyclerViewAdapter
import com.aarush.core.model.model.Source
import com.aarush.newsapp.databinding.ItemSourceBinding

class SourceAdapter : PagingRecyclerViewAdapter<SourceAdapter.ViewHolder, Source>() {

    private var onClick: ((Source) -> Unit)? = null

    override val holderInflater: (LayoutInflater, ViewGroup, Boolean) -> ViewHolder
        get() = { inflater, viewGroup, boolean ->
            ViewHolder(ItemSourceBinding.inflate(inflater, viewGroup, boolean))
        }

    inner class ViewHolder(itemView: ItemSourceBinding) :
        BaseViewHolder<Source, ItemSourceBinding>(itemView) {
        override fun bind(data: Source) {
            with(binding) { this: ItemSourceBinding
                tvSource.text = data.name
                tvDesc.text = data.description

                root.setOnClickListener { it: View!
                    onClick?.invoke(data)
                }
            }
```

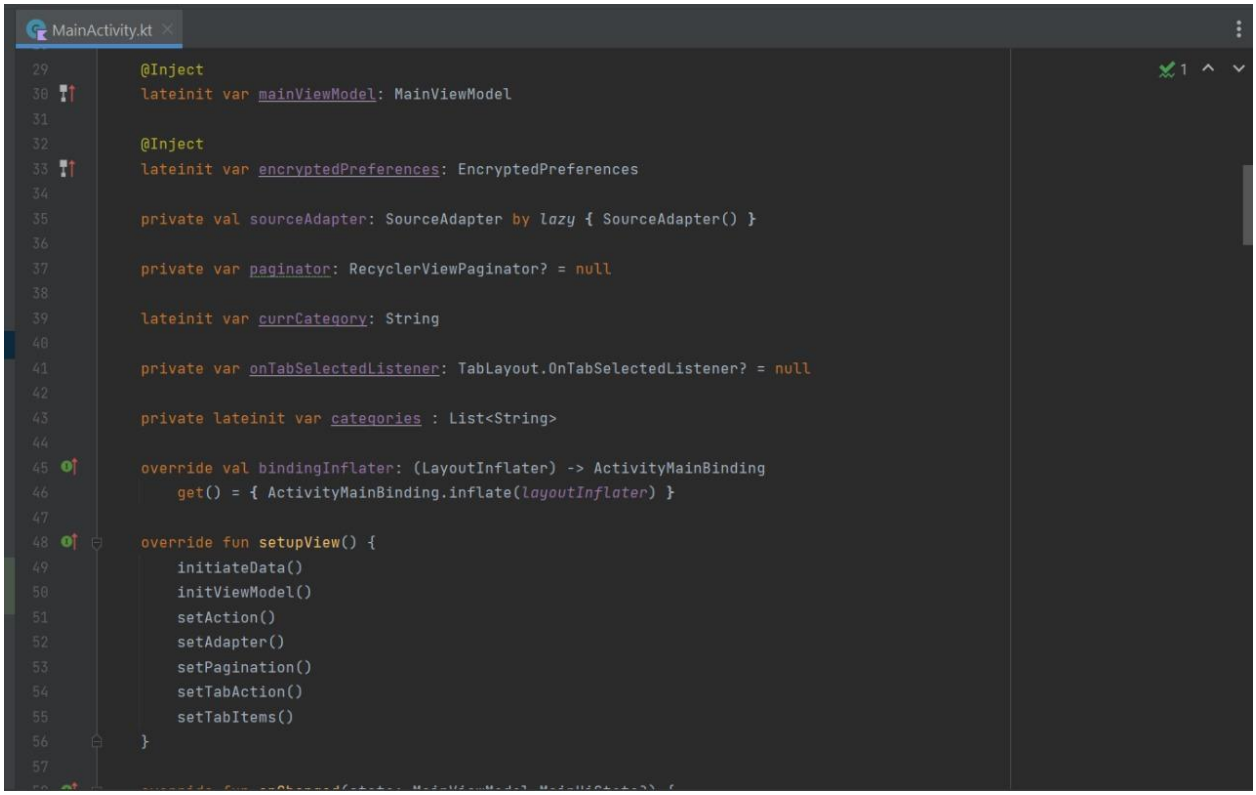
SourceAdapter.kt

```kotlin
    override val holderInflater: (LayoutInflater, ViewGroup, Boolean) -> ViewHolder
        get() = { inflater, viewGroup, boolean ->
            ViewHolder(ItemSourceBinding.inflate(inflater, viewGroup, boolean))
        }

    inner class ViewHolder(itemView: ItemSourceBinding) :
        BaseViewHolder<Source, ItemSourceBinding>(itemView) {
        override fun bind(data: Source) {
            with(binding) { this: ItemSourceBinding
                tvSource.text = data.name
                tvDesc.text = data.description

                root.setOnClickListener { it: View!
                    onClick?.invoke(data)
                }
            }

        }
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.bind(listData[position])
    }

    fun setOnClickData(onClick: (data: Source) -> Unit) {
        this.onClick = onClick
    }
}
```

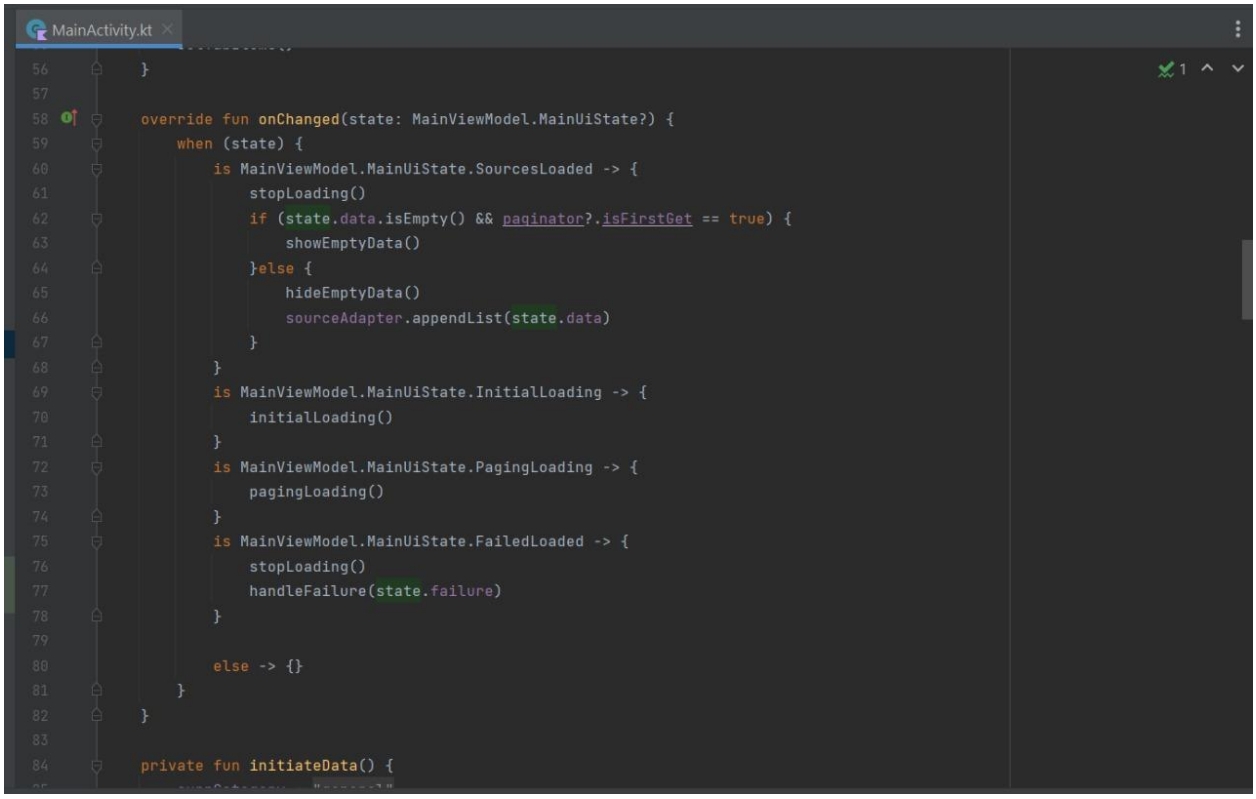
CODE 6: MainActivity.kt



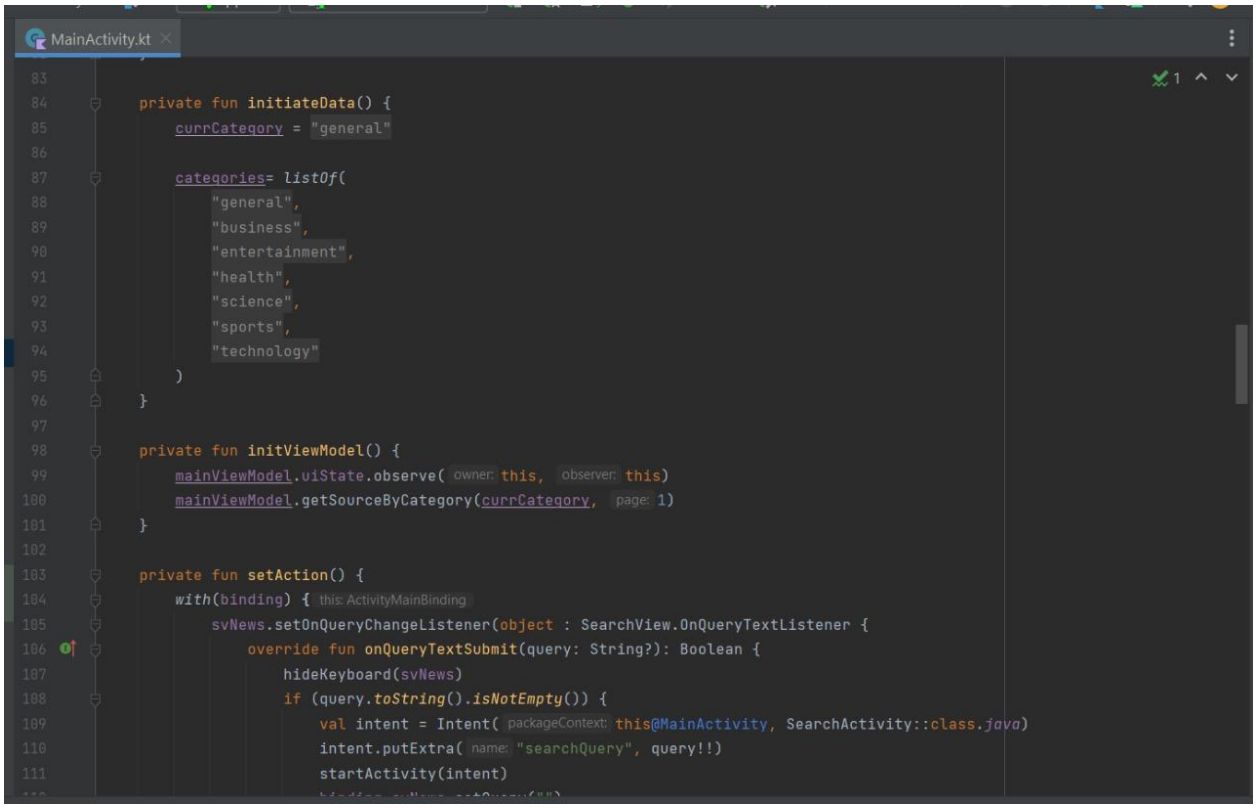
```kotlin
package com.aarush.newsapp.ui.main

import android.app.AlertDialog
import android.content.Intent
import android.view.LayoutInflater
import android.view.View
import android.widget.Toast
import androidx.appcompat.widget.SearchView
import androidx.lifecycle.Observer
import androidx.recyclerview.widget.LinearLayoutManager
import com.aarush.core.abstraction.BaseActivityBinding
import com.aarush.core.pref.EncryptedPreferences
import com.aarush.core.util.exception.Failure
import com.aarush.core.util.extension.dp
import com.aarush.core.util.pagination.RecyclerViewPaginator
import com.aarush.newsapp.R
import com.aarush.newsapp.databinding.ActivityMainBinding
import com.aarush.newsapp.ui.articles.ArticlesActivity
import com.aarush.newsapp.ui.main.adapter.SourceAdapter
import com.aarush.newsapp.ui.search.SearchActivity
import com.google.android.material.tabs.TabLayout
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class MainActivity : BaseActivityBinding<ActivityMainBinding>(),
    Observer<MainViewModel.MainUiState> {

    @Inject
    lateinit var mainViewModel: MainViewModel
```



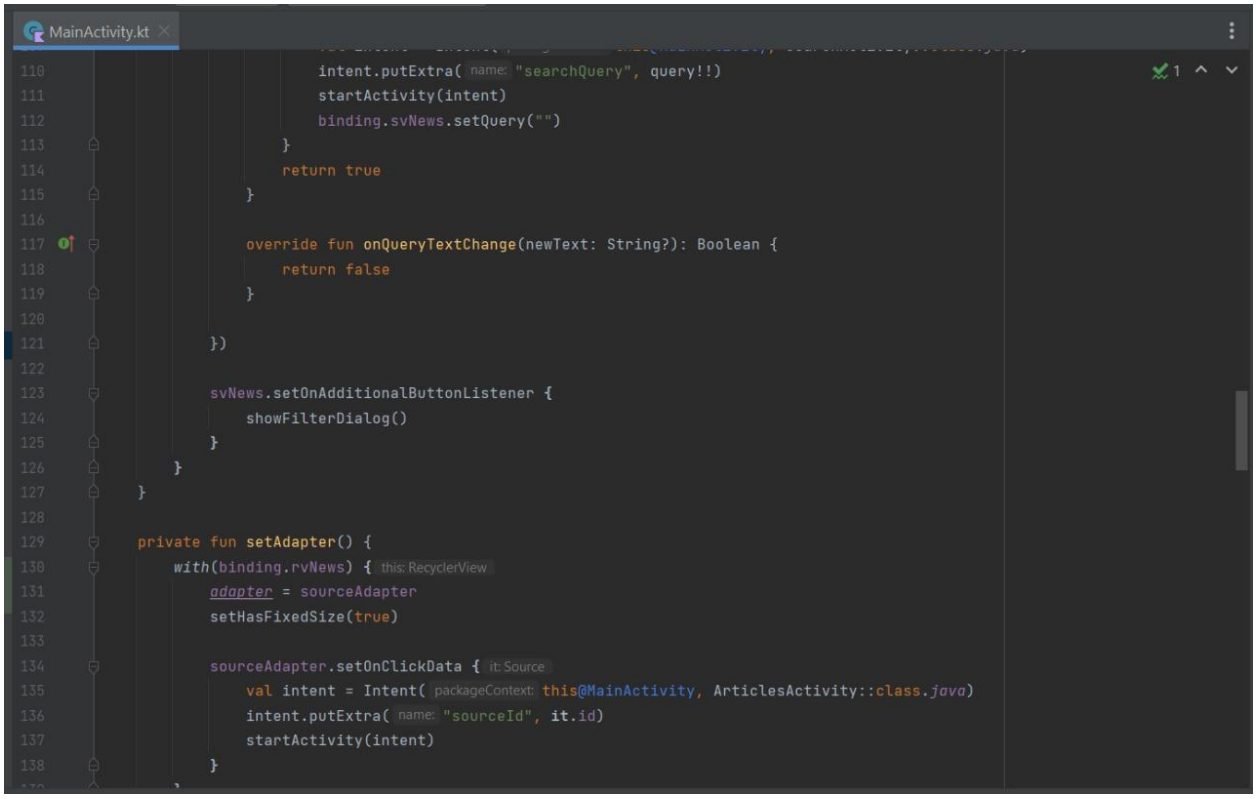
```kotlin
    @Inject
    lateinit var mainViewModel: MainViewModel

    @Inject
    lateinit var encryptedPreferences: EncryptedPreferences

    private val sourceAdapter: SourceAdapter by lazy { SourceAdapter() }

    private var paginator: RecyclerViewPaginator? = null

    lateinit var currCategory: String

    private var onTabSelectedListener: TabLayout.OnTabSelectedListener? = null

    private lateinit var categories : List<String>

    override val bindingInflater: (LayoutInflater) -> ActivityMainBinding
        get() = { ActivityMainBinding.inflate(layoutInflater) }

    override fun setupView() {
        initiateData()
        initViewModel()
        setAction()
        setAdapter()
        setPagination()
        setTabAction()
        setTabItems()
    }
```



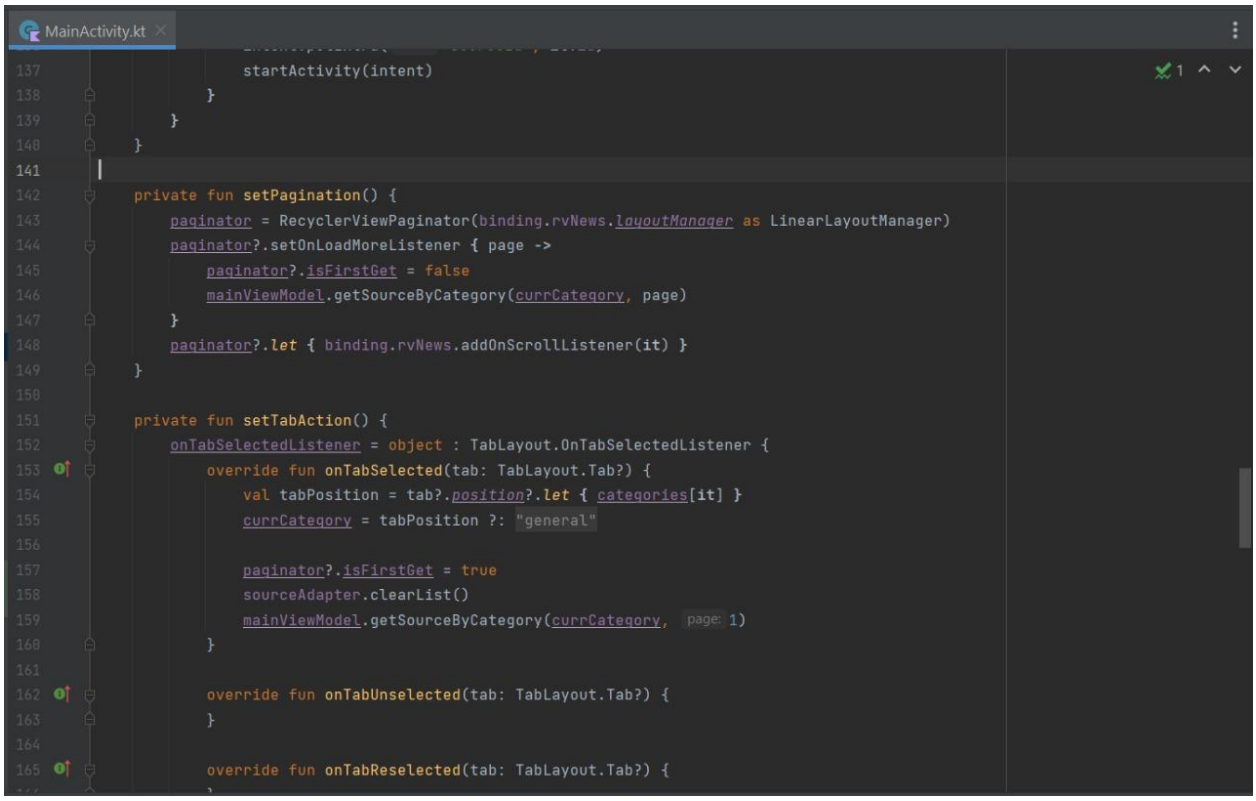
```kotlin
    }

    override fun onChanged(state: MainViewModel.MainUiState?) {
        when (state) {
            is MainViewModel.MainUiState.SourcesLoaded -> {
                stopLoading()
                if (state.data.isEmpty() && paginator?.isFirstGet == true) {
                    showEmptyData()
                }else {
                    hideEmptyData()
                    sourceAdapter.appendList(state.data)
                }
            }
            is MainViewModel.MainUiState.InitialLoading -> {
                initialLoading()
            }
            is MainViewModel.MainUiState.PagingLoading -> {
                pagingLoading()
            }
            is MainViewModel.MainUiState.FailedLoaded -> {
                stopLoading()
                handleFailure(state.failure)
            }

            else -> {}
        }
    }

    private fun initiateData() {
```



```kotlin
    private fun initiateData() {
        currCategory = "general"

        categories= listOf(
            "general",
            "business",
            "entertainment",
            "health",
            "science",
            "sports",
            "technology"
        )
    }

    private fun initViewModel() {
        mainViewModel.uiState.observe( owner: this,  observer: this)
        mainViewModel.getSourceByCategory(currCategory,  page: 1)
    }

    private fun setAction() {
        with(binding) { this: ActivityMainBinding
            svNews.setOnQueryChangeListener(object : SearchView.OnQueryTextListener {
                override fun onQueryTextSubmit(query: String?): Boolean {
                    hideKeyboard(svNews)
                    if (query.toString().isNotEmpty()) {
                        val intent = Intent( packageContext: this@MainActivity, SearchActivity::class.java)
                        intent.putExtra( name: "searchQuery", query!!)
                        startActivity(intent)
```



```
MainActivity.kt
                    intent.putExtra( name: "searchQuery", query!!)
                    startActivity(intent)
                    binding.svNews.setQuery("")
                }
                return true
            }

            override fun onQueryTextChange(newText: String?): Boolean {
                return false
            }

        })

        svNews.setOnAdditionalButtonListener {
            showFilterDialog()
        }
    }
}

private fun setAdapter() {
    with(binding.rvNews) { this: RecyclerView
        adapter = sourceAdapter
        setHasFixedSize(true)

        sourceAdapter.setOnClickData { it: Source
            val intent = Intent( packageContext: this@MainActivity, ArticlesActivity::class.java)
            intent.putExtra( name: "sourceId", it.id)
            startActivity(intent)
        }
```



```
MainActivity.kt
            startActivity(intent)
        }
    }
}

private fun setPagination() {
    paginator = RecyclerViewPaginator(binding.rvNews.layoutManager as LinearLayoutManager)
    paginator?.setOnLoadMoreListener { page ->
        paginator?.isFirstGet = false
        mainViewModel.getSourceByCategory(currCategory, page)
    }
    paginator?.let { binding.rvNews.addOnScrollListener(it) }
}

private fun setTabAction() {
    onTabSelectedListener = object : TabLayout.OnTabSelectedListener {
        override fun onTabSelected(tab: TabLayout.Tab?) {
            val tabPosition = tab?.position?.let { categories[it] }
            currCategory = tabPosition ?: "general"

            paginator?.isFirstGet = true
            sourceAdapter.clearList()
            mainViewModel.getSourceByCategory(currCategory,  page: 1)
        }

        override fun onTabUnselected(tab: TabLayout.Tab?) {
        }

        override fun onTabReselected(tab: TabLayout.Tab?) {
```

MainActivity.kt

```
164
165         override fun onTabReselected(tab: TabLayout.Tab?) {
166         }
167     }
168
169     onTabSelectedListener?.let { it: TabLayout.OnTabSelectedListener
170         binding.tabLayoutCategory.addOnTabSelectedListener(it)
171     }
172 }
173
174 private fun setTabItems() {
175     with(binding.tabLayoutCategory) { this: CustomTabLayout
176         val tabList = mutableListOf<String>()
177         categories.forEach { it: String
178             tabList.add(it)
179         }
180         // Add tabs
181         addTitleOnlyTabs(tabList)
182
183         // Set margins
184         setTabsMargin(6.dp, 6.dp, 8.dp, 6.dp)
185     }
186 }
187
188 private fun initialLoading() {
189     binding.progressCircular.visibility = View.VISIBLE
190     binding.rvNews.visibility = View.GONE
191 }
192
```

MainActivity.kt

```
188 private fun initialLoading() {
189     binding.progressCircular.visibility = View.VISIBLE
190     binding.rvNews.visibility = View.GONE
191 }
192
193 private fun pagingLoading() {
194     binding.progressCircular.visibility = View.VISIBLE
195 }
196
197 private fun stopLoading() {
198     binding.rvNews.visibility = View.VISIBLE
199     binding.progressCircular.visibility = View.GONE
200 }
201
202 private fun showEmptyData() {
203     binding.emptyDataView.parent.visibility = View.VISIBLE
204 }
205
206 private fun hideEmptyData() {
207     binding.emptyDataView.parent.visibility = View.GONE
208 }
209
210 private fun handleFailure(failure: Failure) {
211     Toast.makeText( context: this, failure.throwable.message, Toast.LENGTH_SHORT).show()
212 }
213
214 private fun showFilterDialog() {
215     val filterOptions = arrayOf("Search for Source", "Search for Article")
216     var filter = encryptedPreferences.filter
```

MainActivity.kt

```
        val filterOptions = arrayOf("Search for Source", "Search for Article")
        var filter = encryptedPreferences.filter
        val builder = AlertDialog.Builder( context: this)
        builder.setTitle("Search For")
            .setSingleChoiceItems(filterOptions, if (filter == SEARCH_SOURCE) 0 else 1) { _, which ->
                when (which) {
                    0 -> {
                        filter = SEARCH_SOURCE
                    }
                    1 -> {
                        filter = SEARCH_ARTICLE
                    }
                }
            }
            .setNegativeButton("Cancel") { dialog, _ ->
                dialog.dismiss()
            }
            .setPositiveButton("Apply") { dialog, _ ->
                // Apply button clicked
                encryptedPreferences.filter = filter
                dialog.dismiss()
            }
        val dialog = builder.create()
        dialog.show()
    }
    companion object {
        const val SEARCH_SOURCE = 0
        const val SEARCH_ARTICLE = 1
    }
}
```

CODE 7: MainViewModel.kt

MainViewModel.kt

```
package com.aarush.newsapp.ui.main

import androidx.lifecycle.viewModelScope
import com.aarush.core.abstraction.BaseViewModel
import com.aarush.core.dispatcher.DispatcherProvider
import com.aarush.core.model.model.Source
import com.aarush.core.model.usecase.GetSourcesByCategoryUseCase
import com.aarush.core.util.exception.Failure
import com.aarush.core.util.extension.onError
import com.aarush.core.util.extension.onSuccess
import kotlinx.coroutines.flow.collectLatest
import kotlinx.coroutines.flow.onStart
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import javax.inject.Inject

class MainViewModel @Inject constructor(
    private val dispatcherProvider: DispatcherProvider,
    private val getSourcesByCategory: GetSourcesByCategoryUseCase,
) : BaseViewModel<MainViewModel.MainUiState>() {

    sealed class MainUiState {
        object InitialLoading: MainUiState()
        object PagingLoading: MainUiState()
        data class SourcesLoaded(val data: List<Source>) : MainUiState()
        data class FailedLoaded(val failure: Failure) : MainUiState()
    }

    fun getSourceByCategory(category: String, page: Long){
        _uiState.value = if (page == 1L) {
```

MainViewModel.kt

```
    sealed class MainUiState {
        object InitialLoading: MainUiState()
        object PagingLoading: MainUiState()
        data class SourcesLoaded(val data: List<Source>) : MainUiState()
        data class FailedLoaded(val failure: Failure) : MainUiState()
    }

    fun getSourceByCategory(category: String, page: Long){
        _uiState.value = if (page == 1L) {
            MainUiState.InitialLoading
        }else{
            MainUiState.PagingLoading
        }
        viewModelScope.launch(dispatcherProvider.io) { this: CoroutineScope
            getSourcesByCategory.run(GetSourcesByCategoryUseCase.Params(category, page.toInt()))
                .onSuccess { it: List<Source>
                    withContext(dispatcherProvider.main) { this: CoroutineScope
                        _uiState.value = MainUiState.SourcesLoaded(it)
                    }
                }
                .onError { it: Failure
                    withContext(dispatcherProvider.main) { this: CoroutineScope
                        _uiState.value = MainUiState.FailedLoaded(it)
                    }
                }
        }
    }

}
```

CODE 8: SearchAdapter.kt

SearchAdapter.kt

```
package com.aarush.newsapp.ui.search.adapter

import android.view.LayoutInflater
import android.view.ViewGroup
import androidx.recyclerview.widget.RecyclerView
import com.aarush.core.abstraction.PagingRecyclerViewAdapter
import com.aarush.core.model.model.Article
import com.aarush.core.model.model.Source
import com.aarush.core.util.extension.glide
import com.aarush.newsapp.databinding.ItemArticlesBinding
import com.aarush.newsapp.databinding.ItemSourceBinding
import com.aarush.newsapp.ui.main.MainActivity.Companion.SEARCH_ARTICLE
import com.aarush.newsapp.ui.main.MainActivity.Companion.SEARCH_SOURCE

class SearchAdapter : PagingRecyclerViewAdapter<RecyclerView.ViewHolder, Any>() {

    private var onClick: ((Any, Int) -> Unit)? = null
    private var viewType = SEARCH_SOURCE

    fun setTypeAdapter(type: Int) {
        viewType = type
        notifyDataSetChanged()
    }

    override val holderInflater: (LayoutInflater, ViewGroup, Boolean) -> RecyclerView.ViewHolder
        get() = { inflater, viewGroup, boolean ->
            when (viewType) {
                SEARCH_SOURCE -> {
                    SourceViewHolder(ItemSourceBinding.inflate(inflater, viewGroup, boolean)) ^lambda
                }
```

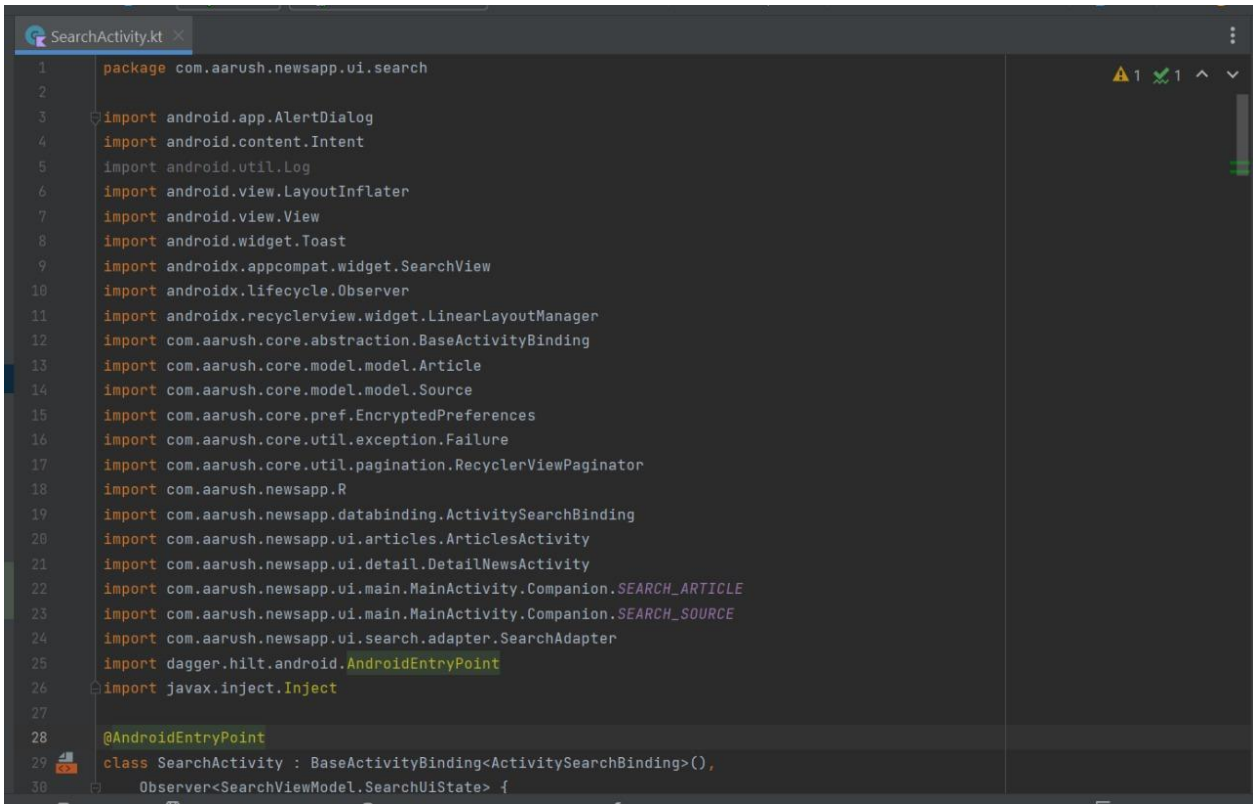
SearchAdapter.kt

```
                SEARCH_SOURCE -> {
                    SourceViewHolder(ItemSourceBinding.inflate(inflater, viewGroup, boolean)) ^lambda
                }
                SEARCH_ARTICLE -> {
                    ArticleViewHolder(ItemArticlesBinding.inflate(inflater, viewGroup, boolean)) ^lambda
                }
                else -> throw IllegalArgumentException("Invalid view type")
            }
        }

    inner class SourceViewHolder(private val binding: ItemSourceBinding) : RecyclerView.ViewHolder(binding.root) {
        fun bind(data: Source) {
            with(binding) { this: ItemSourceBinding
                tvSource.text = data.name
                tvDesc.text = data.description

                root.setOnClickListener { it: View!
                    onClick?.invoke(data, viewType)
                }
            }
        }
    }

    inner class ArticleViewHolder(private val binding: ItemArticlesBinding) : RecyclerView.ViewHolder(binding.root) {
        fun bind(data: Article) {
            with(binding) { this: ItemArticlesBinding
                tvTitle.text = data.title
                tvDesc.text = data.description
                imgArticle.glide(itemView, data.image)
```

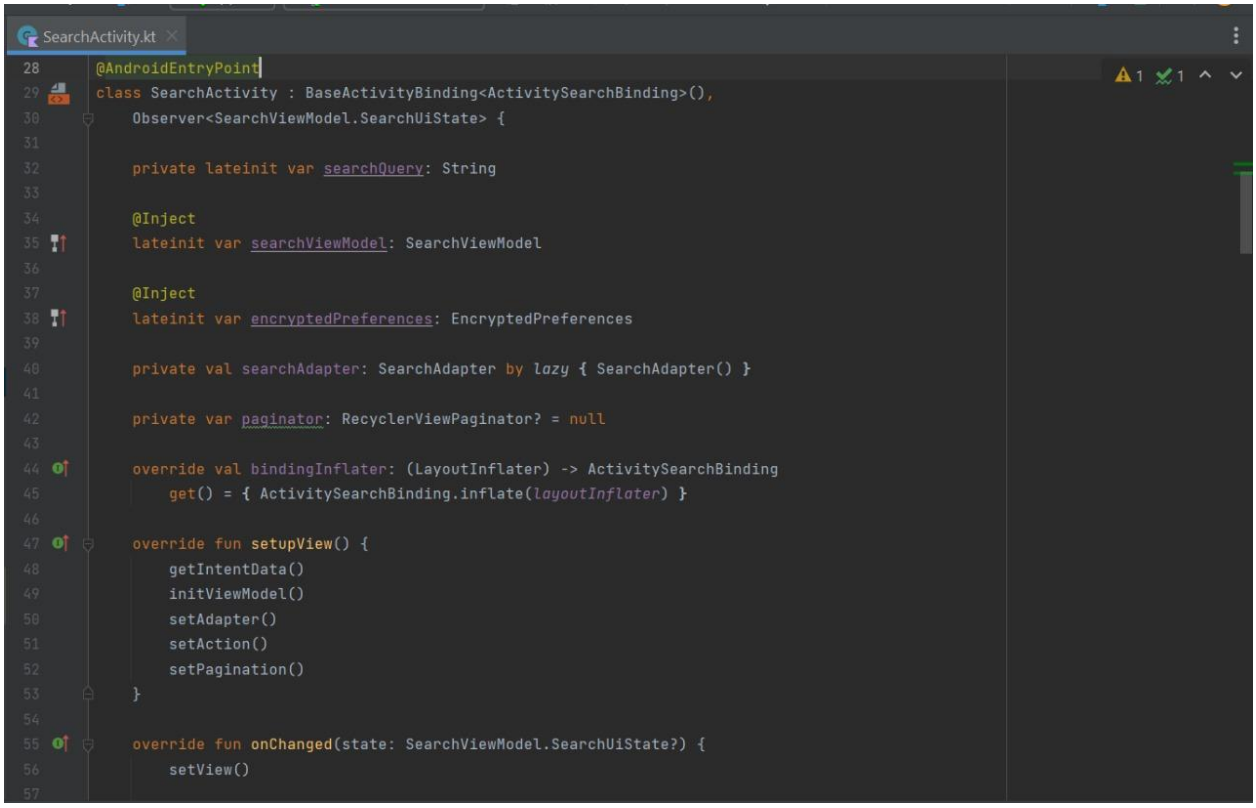
SearchAdapter.kt

```
    }

    inner class ArticleViewHolder(private val binding: ItemArticlesBinding) : RecyclerView.ViewHolder(binding.root) {
        fun bind(data: Article) {
            with(binding) { this: ItemArticlesBinding
                tvTitle.text = data.title
                tvDesc.text = data.description
                imgArticle.glide(itemView, data.image)

                root.setOnClickListener { it: View!
                    onClick?.invoke(data, viewType)
                }
            }
        }
    }

    override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
        when (holder) {
            is SourceViewHolder -> holder.bind(listData[position] as Source)
            is ArticleViewHolder -> holder.bind(listData[position] as Article)
        }
    }

    fun setOnClickData(onClick: (Any, Int) -> Unit) {
        this.onClick = onClick
    }

}
```

CODE 9: SearchActivity.kt



```kotlin
package com.aarush.newsapp.ui.search

import android.app.AlertDialog
import android.content.Intent
import android.util.Log
import android.view.LayoutInflater
import android.view.View
import android.widget.Toast
import androidx.appcompat.widget.SearchView
import androidx.lifecycle.Observer
import androidx.recyclerview.widget.LinearLayoutManager
import com.aarush.core.abstraction.BaseActivityBinding
import com.aarush.core.model.model.Article
import com.aarush.core.model.model.Source
import com.aarush.core.pref.EncryptedPreferences
import com.aarush.core.util.exception.Failure
import com.aarush.core.util.pagination.RecyclerViewPaginator
import com.aarush.newsapp.R
import com.aarush.newsapp.databinding.ActivitySearchBinding
import com.aarush.newsapp.ui.articles.ArticlesActivity
import com.aarush.newsapp.ui.detail.DetailNewsActivity
import com.aarush.newsapp.ui.main.MainActivity.Companion.SEARCH_ARTICLE
import com.aarush.newsapp.ui.main.MainActivity.Companion.SEARCH_SOURCE
import com.aarush.newsapp.ui.search.adapter.SearchAdapter
import dagger.hilt.android.AndroidEntryPoint
import javax.inject.Inject

@AndroidEntryPoint
class SearchActivity : BaseActivityBinding<ActivitySearchBinding>(),
    Observer<SearchViewModel.SearchUiState> {
```



```kotlin
@AndroidEntryPoint
class SearchActivity : BaseActivityBinding<ActivitySearchBinding>(),
    Observer<SearchViewModel.SearchUiState> {

    private lateinit var searchQuery: String

    @Inject
    lateinit var searchViewModel: SearchViewModel

    @Inject
    lateinit var encryptedPreferences: EncryptedPreferences

    private val searchAdapter: SearchAdapter by lazy { SearchAdapter() }

    private var paginator: RecyclerViewPaginator? = null

    override val bindingInflater: (LayoutInflater) -> ActivitySearchBinding
        get() = { ActivitySearchBinding.inflate(layoutInflater) }

    override fun setupView() {
        getIntentData()
        initViewModel()
        setAdapter()
        setAction()
        setPagination()
    }

    override fun onChanged(state: SearchViewModel.SearchUiState?) {
        setView()
```

SearchActivity.kt

```
57
58          when (state) {
59              is SearchViewModel.SearchUiState.SourcesLoaded -> {
60                  stopLoading()
61                  if (state.data.isEmpty() && paginator?.isFirstGet == true) {
62                      showEmptyData()
63                  }else {
64                      hideEmptyData()
65                      searchAdapter.appendList(state.data)
66                  }
67              }
68
69              is SearchViewModel.SearchUiState.ArticleLoaded -> {
70                  stopLoading()
71                  if (state.data.isEmpty() && paginator?.isFirstGet == true) {
72                      showEmptyData()
73                  }else {
74                      hideEmptyData()
75                      searchAdapter.appendList(state.data)
76                  }
77              }
78
79              is SearchViewModel.SearchUiState.InitialLoading -> {
80                  initialLoading()
81              }
82
83              is SearchViewModel.SearchUiState.PagingLoading -> {
84                  pagingLoading()
85              }
86
```

SearchActivity.kt

```
87              is SearchViewModel.SearchUiState.FailedLoaded -> {
88                  stopLoading()
89                  handleFailure(state.failure)
90              }
91
92              else -> {
93
94              }
95          }
96      }
97
98      private fun getIntentData() {
99          searchQuery = intent.getStringExtra( name: "searchQuery") ?: ""
100     }
101
102     private fun initViewModel() {
103         searchViewModel.uiState.observe( owner: this,  observer: this)
104         if (encryptedPreferences.filter == SEARCH_SOURCE) {
105             searchViewModel.getSourceByName(searchQuery,  page: 1)
106         } else {
107             searchViewModel.getArticlesByQuery(searchQuery,  page: 1)
108         }
109     }
110
111     private fun setView() {
112         binding.tvResult.text =  "Result for {searchQuery}"
113     }
114
115     private fun setAction() {
116         with(binding) { this: ActivitySearchBinding
```

SearchActivity.kt

```
114
115     private fun setAction() {
116         with(binding) { this: ActivitySearchBinding
117             svNews.setQuery(searchQuery)
118             svNews.setOnQueryChangeListener(object : SearchView.OnQueryTextListener {
119                 override fun onQueryTextSubmit(query: String?): Boolean {
120                     hideKeyboard(svNews)
121                     if (query.toString().isNotEmpty()) {
122                         searchQuery = query!!
123                         if (encryptedPreferences.filter == SEARCH_SOURCE) {
124                             searchViewModel.getSourceByName(searchQuery, page: 1)
125                             searchAdapter.setTypeAdapter(SEARCH_SOURCE)
126                             searchAdapter.clearList()
127                             binding.rvNews.adapter = searchAdapter
128                         } else {
129                             searchViewModel.getArticlesByQuery(searchQuery, page: 1)
130                             searchAdapter.setTypeAdapter(SEARCH_ARTICLE)
131                             searchAdapter.clearList()
132                             binding.rvNews.adapter = searchAdapter
133                         }
134                     }
135                     return true
136                 }
137
138                 override fun onQueryTextChange(newText: String?): Boolean {
139
140                     return false
141                 }
142
143             })
```

SearchActivity.kt

```
141                 }
142
143             })
144
145             svNews.setOnAdditionalButtonListener {
146                 showFilterDialog()
147             }
148
149             btnBack.setOnClickListener { onBackPressedDispatcher.onBackPressed() }
150
151         }
152     }
153
154     private fun setAdapter() {
155         with(binding.rvNews) { this: RecyclerView
156             searchAdapter.setTypeAdapter(encryptedPreferences.filter)
157             adapter = searchAdapter
158             setHasFixedSize(true)
159
160             searchAdapter.setOnClickData { it, filter ->
161                 if (filter == SEARCH_SOURCE) {
162                     val data = it as Source
163                     val intent = Intent( packageContext: this@SearchActivity, ArticlesActivity::class.java)
164                     intent.putExtra( name: "sourceId", data.id)
165                     startActivity(intent)
166                 } else {
167                     val data = it as Article
168                     val intent = Intent( packageContext: this@SearchActivity, DetailNewsActivity::class.java)
169                     intent.putExtra( name: "sourceNews", it.source)
170                     intent.putExtra( name: "urlNews", data.url)
```

SearchActivity.kt

```
                    startActivity(intent)
                }
            }
        }
    }

    private fun setPagination() {
        paginator = RecyclerViewPaginator(binding.rvNews.layoutManager as LinearLayoutManager)
        paginator?.setOnLoadMoreListener { page ->
            paginator?.isFirstGet = false
            if (encryptedPreferences.filter == SEARCH_SOURCE) {
                searchViewModel.getSourceByName(searchQuery, page)
            } else {
                searchViewModel.getArticlesByQuery(searchQuery, page)
            }
        }
        paginator?.let { binding.rvNews.addOnScrollListener(it) }
    }

    private fun initialLoading() {
        binding.progressCircular.visibility = View.VISIBLE
        binding.rvNews.visibility = View.GONE
    }

    private fun pagingLoading() {
        binding.progressCircular.visibility = View.VISIBLE
    }

    private fun stopLoading() {
        binding.rvNews.visibility = View.VISIBLE
```

SearchActivity.kt

```

    private fun stopLoading() {
        binding.rvNews.visibility = View.VISIBLE
        binding.progressCircular.visibility = View.GONE
    }

    private fun showEmptyData() {
        binding.emptyDataView.parent.visibility = View.VISIBLE
    }

    private fun hideEmptyData() {
        binding.emptyDataView.parent.visibility = View.GONE
    }

    private fun handleFailure(failure: Failure) {
        Toast.makeText( context: this, failure.throwable.message, Toast.LENGTH_SHORT).show()
    }

    private fun showFilterDialog() {
        val filterOptions = arrayOf("Search for Source", "Search for Article")
        var filter = encryptedPreferences.filter
        val builder = AlertDialog.Builder( context: this)
        builder.setTitle("Search For")
            .setSingleChoiceItems(filterOptions, if (filter == SEARCH_SOURCE) 0 else 1) { _, which ->
                when (which) {
                    0 -> {
                        filter = SEARCH_SOURCE
                    }
                    1 -> {
                        filter = SEARCH_ARTICLE
```

```kotlin
SearchActivity.kt
            val filterOptions = arrayOf("Search for Source", "Search for Article")
            var filter = encryptedPreferences.filter
            val builder = AlertDialog.Builder( context: this)
            builder.setTitle("Search For")
                .setSingleChoiceItems(filterOptions, if (filter == SEARCH_SOURCE) 0 else 1) { _, which ->
                    when (which) {
                        0 -> {
                            filter = SEARCH_SOURCE
                        }
                        1 -> {
                            filter = SEARCH_ARTICLE
                        }
                    }
                }
                .setNegativeButton("Cancel") { dialog, _ ->
                    dialog.dismiss()
                }
                .setPositiveButton("Apply") { dialog, _ ->
                    // Apply button clicked
                    encryptedPreferences.filter = filter
                    dialog.dismiss()
                }

            val dialog = builder.create()
            dialog.show()
        }

}
```

CODE 10: Search View Model

```kotlin
SearchViewModel.kt
package com.aarush.newsapp.ui.search

import android.util.Log
import androidx.lifecycle.viewModelScope
import com.aarush.core.abstraction.BaseViewModel
import com.aarush.core.dispatcher.DispatcherProvider
import com.aarush.core.model.model.Article
import com.aarush.core.model.model.Source
import com.aarush.core.model.usecase.GetArticlesByQueryUseCase
import com.aarush.core.model.usecase.GetSourceByNameUseCase
import com.aarush.core.util.exception.Failure
import com.aarush.core.util.extension.onError
import com.aarush.core.util.extension.onSuccess
import kotlinx.coroutines.launch
import kotlinx.coroutines.withContext
import javax.inject.Inject

class SearchViewModel @Inject constructor(
    private val dispatcherProvider: DispatcherProvider,
    private val getSourceByNameUseCase: GetSourceByNameUseCase,
    private val getArticlesByQueryUseCase: GetArticlesByQueryUseCase,
) : BaseViewModel<SearchViewModel.SearchUiState>() {

    sealed class SearchUiState {
        object InitialLoading : SearchUiState()
        object PagingLoading : SearchUiState()
        data class SourcesLoaded(val data: List<Source>) : SearchUiState()
        data class ArticleLoaded(val data: List<Article>) : SearchUiState()
        data class FailedLoaded(val failure: Failure) : SearchUiState()
    }
```

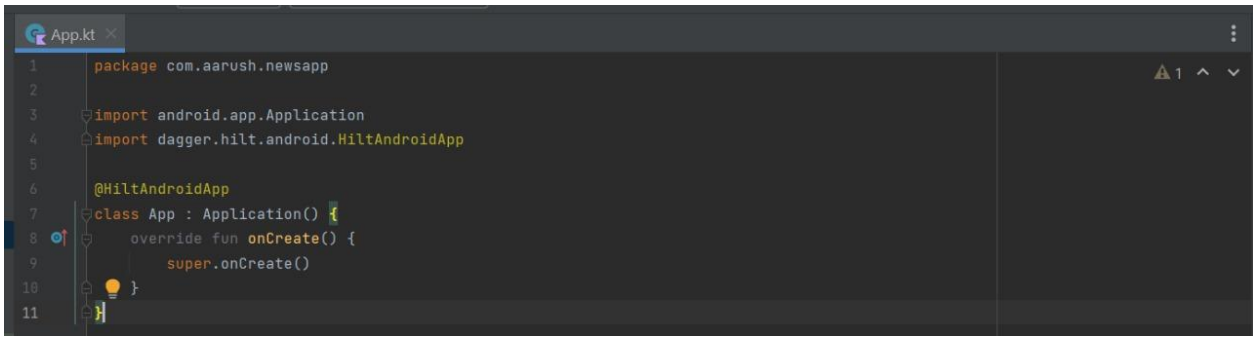
SearchViewModel.kt

```
        data class ArticleLoaded(val data: List<Article>) : SearchUiState()
        data class FailedLoaded(val failure: Failure) : SearchUiState()
    }

    fun getSourceByName(name: String, page: Long) {
        _uiState.value = if (page == 1L) {
            SearchUiState.InitialLoading
        } else {
            SearchUiState.PagingLoading
        }
        viewModelScope.launch(dispatcherProvider.io) { this: CoroutineScope
            val data = getSourceByNameUseCase.getSourceByName(name, page.toInt())
            val dataString = data.joinToString(separator = ", ")
            Log.d( tag: "datalistv", dataString)
            withContext(dispatcherProvider.main) { this: CoroutineScope
                _uiState.value = SearchUiState.SourcesLoaded(data)
            }
        }
    }

    fun getArticlesByQuery(source: String, page: Long) {
        _uiState.value = if (page == 1L) {
            SearchUiState.InitialLoading
        } else {
            SearchUiState.PagingLoading
        }
        viewModelScope.launch(dispatcherProvider.io) { this: CoroutineScope
            getArticlesByQueryUseCase.run(GetArticlesByQueryUseCase.Params(source, page.toString()))
                .onSuccess { it: List<Article>
                    withContext(dispatcherProvider.main) { this: CoroutineScope
```

SearchViewModel.kt

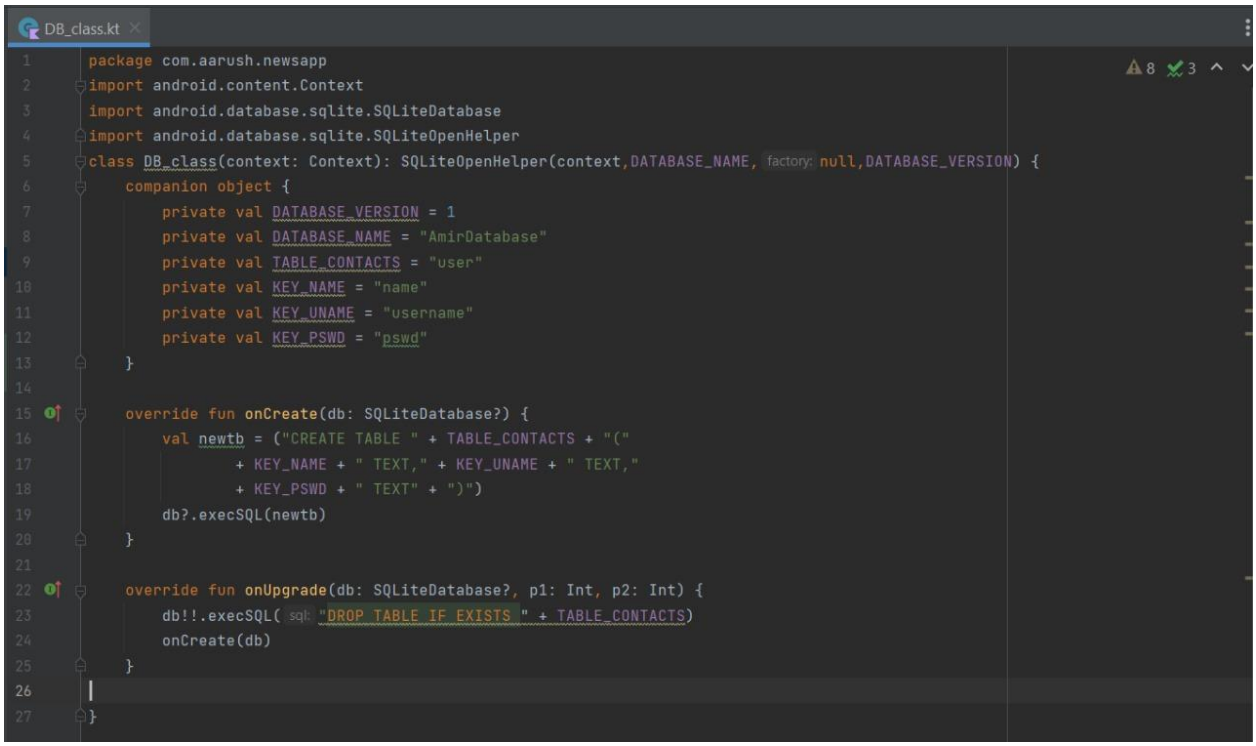
```
                withContext(dispatcherProvider.main) { this: CoroutineScope
                    _uiState.value = SearchUiState.SourcesLoaded(data)
                }
            }
        }

        fun getArticlesByQuery(source: String, page: Long) {
            _uiState.value = if (page == 1L) {
                SearchUiState.InitialLoading
            } else {
                SearchUiState.PagingLoading
            }
            viewModelScope.launch(dispatcherProvider.io) { this: CoroutineScope
                getArticlesByQueryUseCase.run(GetArticlesByQueryUseCase.Params(source, page.toString()))
                    .onSuccess { it: List<Article>
                        withContext(dispatcherProvider.main) { this: CoroutineScope
                            _uiState.value = SearchUiState.ArticleLoaded(it)
                        }
                    }
                    .onError { it: Failure
                        withContext(dispatcherProvider.main) { this: CoroutineScope
                            _uiState.value = SearchUiState.FailedLoaded(it)
                        }
                    }
            }
        }

}
```

CODE 11: App.kt



```kotlin
package com.aarush.newsapp

import android.app.Application
import dagger.hilt.android.HiltAndroidApp

@HiltAndroidApp
class App : Application() {
    override fun onCreate() {
        super.onCreate()
    }
}
```

CODE 12: DB_class.kt



```kotlin
package com.aarush.newsapp
import android.content.Context
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper
class DB_class(context: Context): SQLiteOpenHelper(context,DATABASE_NAME, factory: null,DATABASE_VERSION) {
    companion object {
        private val DATABASE_VERSION = 1
        private val DATABASE_NAME = "AmirDatabase"
        private val TABLE_CONTACTS = "user"
        private val KEY_NAME = "name"
        private val KEY_UNAME = "username"
        private val KEY_PSWD = "pswd"
    }

    override fun onCreate(db: SQLiteDatabase?) {
        val newtb = ("CREATE TABLE " + TABLE_CONTACTS + "("
                + KEY_NAME + " TEXT," + KEY_UNAME + " TEXT,"
                + KEY_PSWD + " TEXT" + ")")
        db?.execSQL(newtb)
    }

    override fun onUpgrade(db: SQLiteDatabase?, p1: Int, p2: Int) {
        db!!.execSQL( sql: "DROP TABLE IF EXISTS " + TABLE_CONTACTS)
        onCreate(db)
    }

}
```

CODE 13: login_form.kt

login_form.kt

```kotlin
package com.aarush.newsapp

import android.content.Intent
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import android.widget.Toast
import androidx.appcompat.app.AlertDialog
import com.aarush.newsapp.databinding.ActivityLoginFormBinding
import com.aarush.newsapp.ui.main.MainActivity

class login_form : AppCompatActivity() {
    private lateinit var bind: ActivityLoginFormBinding
    private var isPasswordVisible = false

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        bind = ActivityLoginFormBinding.inflate(layoutInflater)
        setContentView(bind.root)
        val dbhelp = DB_class(applicationContext)
        val db = dbhelp.readableDatabase

        bind.btnlogin.setOnClickListener { it: View!
            val username = bind.logtxt.text.toString()
            val password = bind.ed3.text.toString()
            val query = "SELECT * FROM user WHERE username='$username' AND pswd='$password'"
            val rs = db.rawQuery(query,  selectionArgs: null)
            if (rs.moveToFirst()) {
                val name = rs.getString(rs.getColumnIndex("name"))
                rs.close()
```

login_form.kt

```kotlin
                val name = rs.getString(rs.getColumnIndex("name"))
                rs.close()
                startActivity(Intent( packageContext: this, MainActivity::class.java).putExtra( name: "name", name))
            } else {
                val ad = AlertDialog.Builder( context: this)
                ad.setTitle("Message")
                ad.setMessage("Username or password is incorrect!")
                ad.setPositiveButton( text: "Ok",  listener: null)
                ad.show()
            }
        }

        bind.regisLink.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, signupForm::class.java)
            startActivity(intent)
        }

        // Toggle password visibility
        bind.passwordVisibilityButton.setOnClickListener { it: View!
            isPasswordVisible = !isPasswordVisible
            if (isPasswordVisible) {
                bind.ed3.inputType = 129 // Show Password
            } else {
                bind.ed3.inputType = 128 // Hide Password
            }
            bind.ed3.setSelection(bind.ed3.text.length)
        }
    }
}
```

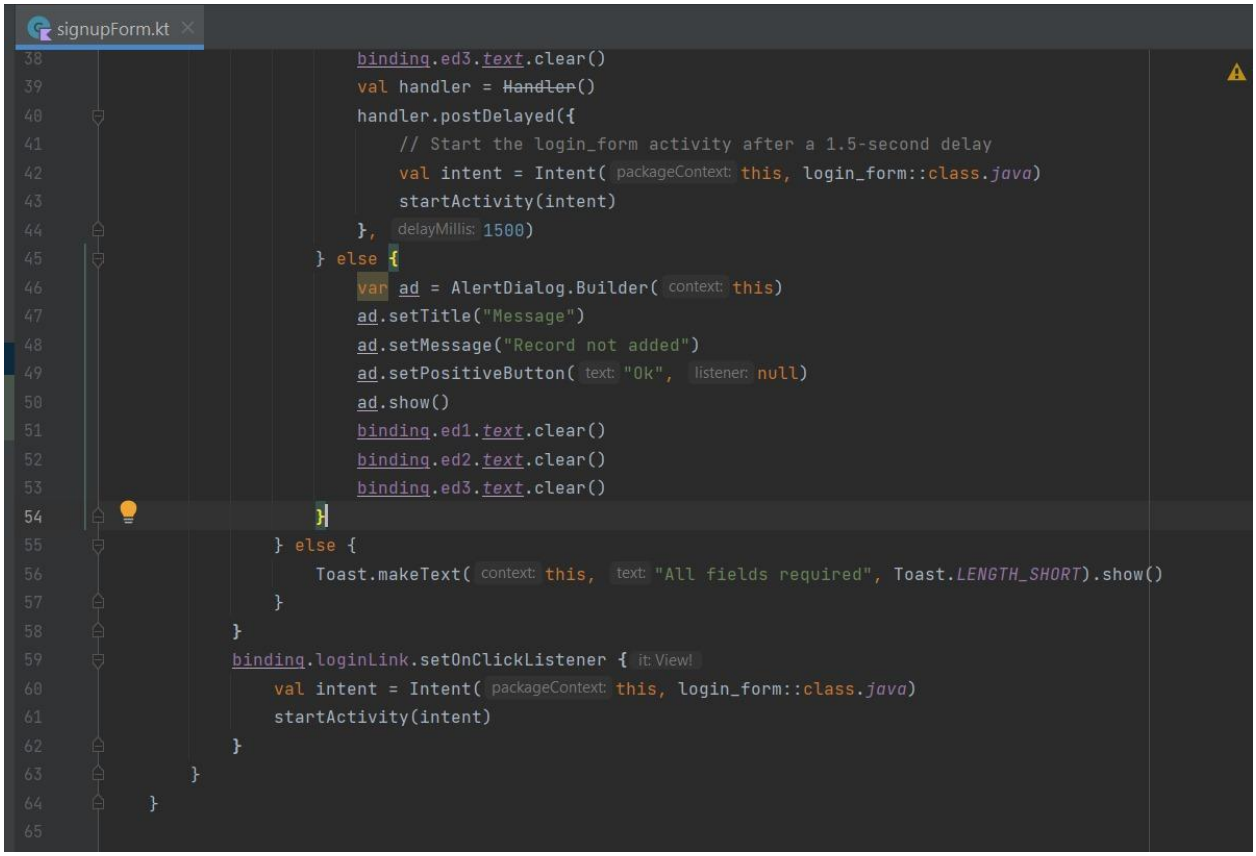
CODE 14: signupForm.kt

signupForm.kt

```kotlin
package com.aarush.newsapp

import android.app.AlertDialog;
import android.content.ContentValues;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Toast;
import android.os.Handler
import androidx.appcompat.app.AppCompatActivity;
import com.aarush.newsapp.databinding.ActivitySignupFormBinding;
class signupForm : AppCompatActivity() {
        private lateinit var binding: ActivitySignupFormBinding

        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)
            binding = ActivitySignupFormBinding.inflate(layoutInflater)
            setContentView(binding.root)
            var dbhelp = DB_class(applicationContext)
            var db = dbhelp.writableDatabase
            binding.btnrgs.setOnClickListener { it: View!
                var name = binding.ed1.text.toString()
                var username = binding.ed2.text.toString()
                var password = binding.ed3.text.toString()
                if (name.isNotEmpty() && username.isNotEmpty() && password.isNotEmpty()) {
                    var data = ContentValues()
                    data.put("name", binding.ed1.text.toString())
                    data.put("username", binding.ed2.text.toString())
                    data.put("pswd", binding.ed3.text.toString())
                    var rs: Long = db.insert( table: "user",  nullColumnHack: null, data)
                    if (!rs.equals(-1)) {
```

signupForm.kt

```kotlin
                        var ad = AlertDialog.Builder( context: this)
                        ad.setTitle("Message")
                        ad.setMessage("Account registered successfully")
                        ad.setPositiveButton( text: "Ok",  listener: null)
                        ad.show()
                        binding.ed1.text.clear()
                        binding.ed2.text.clear()
                        binding.ed3.text.clear()
                        val handler = Handler()
                        handler.postDelayed({
                            // Start the login_form activity after a 1.5-second delay
                            val intent = Intent( packageContext: this, login_form::class.java)
                            startActivity(intent)
                        },  delayMillis: 1500)
                    } else {
                        var ad = AlertDialog.Builder( context: this)
                        ad.setTitle("Message")
                        ad.setMessage("Record not added")
                        ad.setPositiveButton( text: "Ok",  listener: null)
                        ad.show()
                        binding.ed1.text.clear()
                        binding.ed2.text.clear()
                        binding.ed3.text.clear()
                    }
                } else {
                    Toast.makeText( context: this,  text: "All fields required", Toast.LENGTH_SHORT).show()
                }
            }
            binding.loginLink.setOnClickListener { it: View!
                val intent = Intent( packageContext: this, login_form::class.java)
```



CODE 15: CustomSearchBar.kt

CustomSearchBar.kt

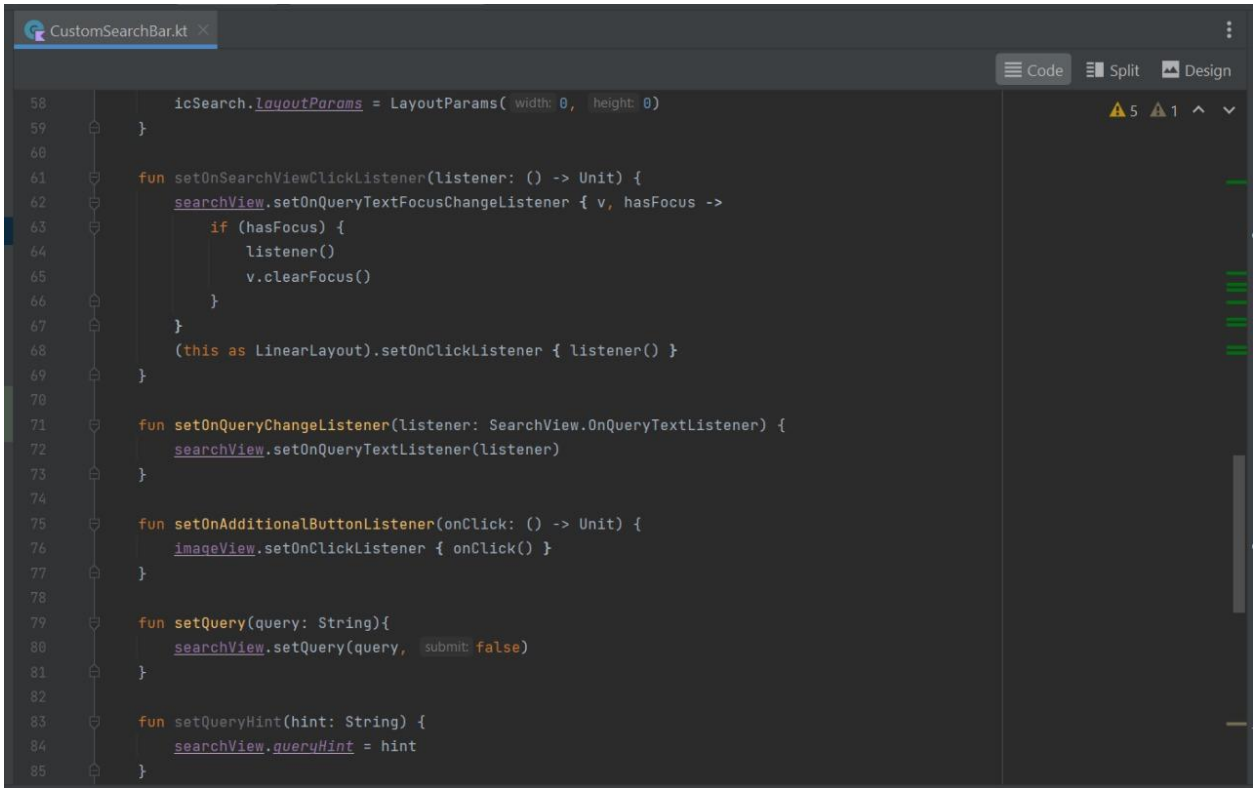
Code | Split | Design

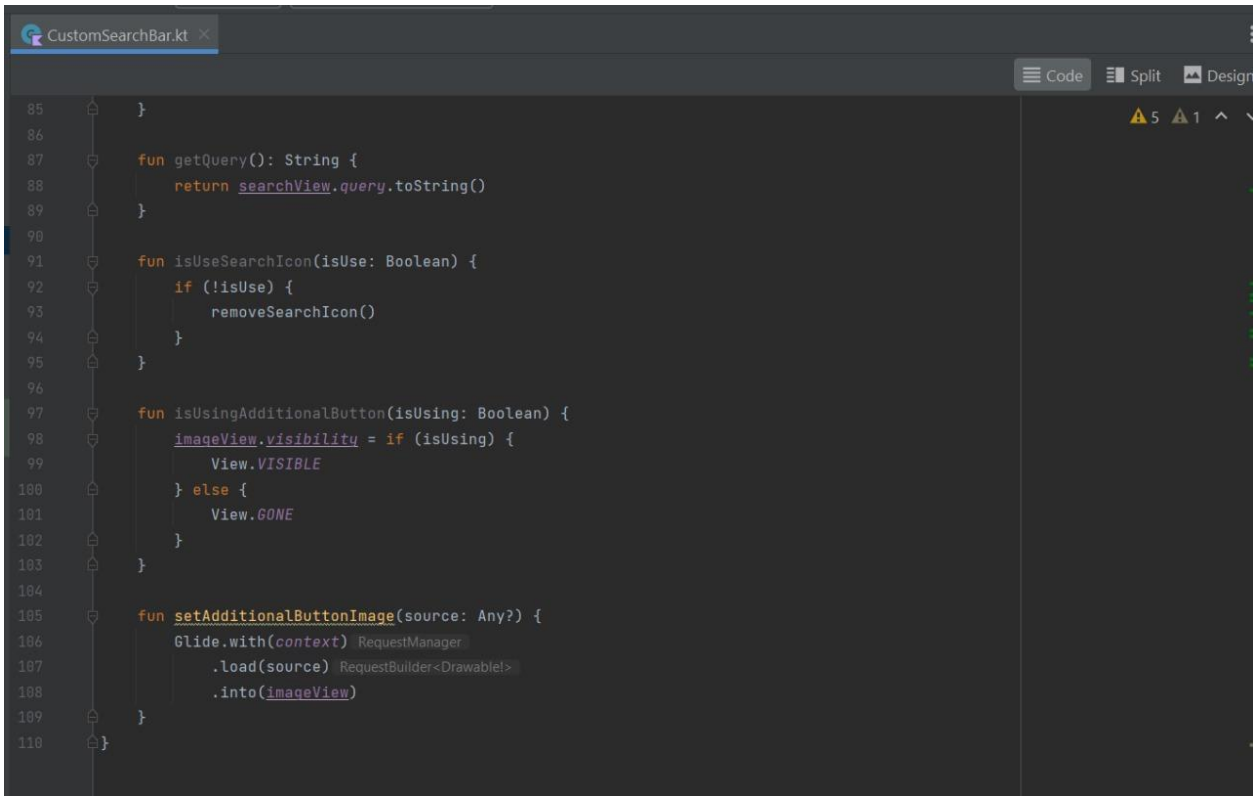
```kotlin
package com.aarush.newsapp.widget

import android.content.Context
import android.util.AttributeSet
import android.view.View
import android.widget.ImageView
import android.widget.LinearLayout
import androidx.appcompat.widget.SearchView
import com.bumptech.glide.Glide
import com.aarush.newsapp.R

class CustomSearchBar : LinearLayout {

    private lateinit var searchView: SearchView
    private lateinit var imageView: ImageView

    constructor(context: Context, attrs: AttributeSet?) : super(context, attrs) {
        initView(context, attrs)
    }
    constructor(context: Context, attrs: AttributeSet?, defStyleAttr: Int) :
            super(context, attrs, defStyleAttr) {
        initView(context, attrs)
    }

    private fun initView(context: Context, attrs: AttributeSet?) {
        inflate(context, R.layout.searchview_custom,  root: this)

        searchView = findViewById(R.id.searchBar)
```

CustomSearchBar.kt

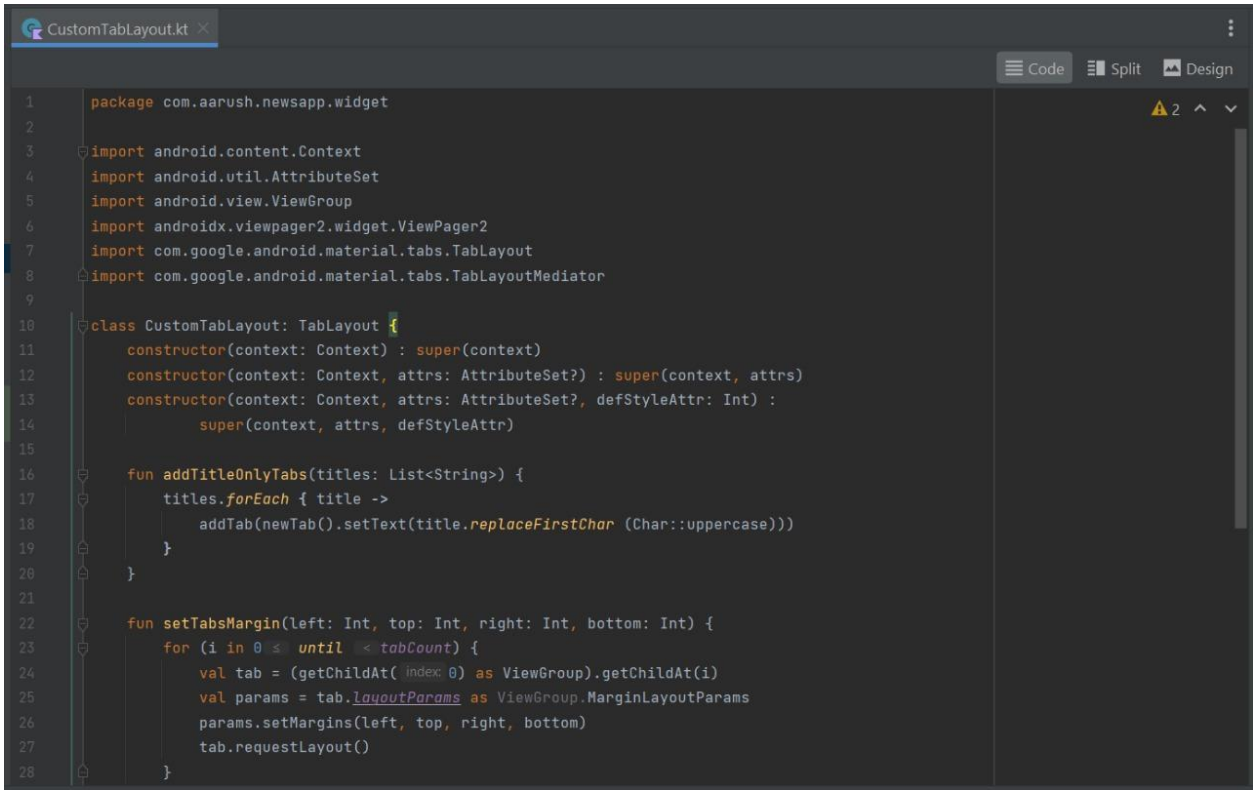
Code | Split | Design

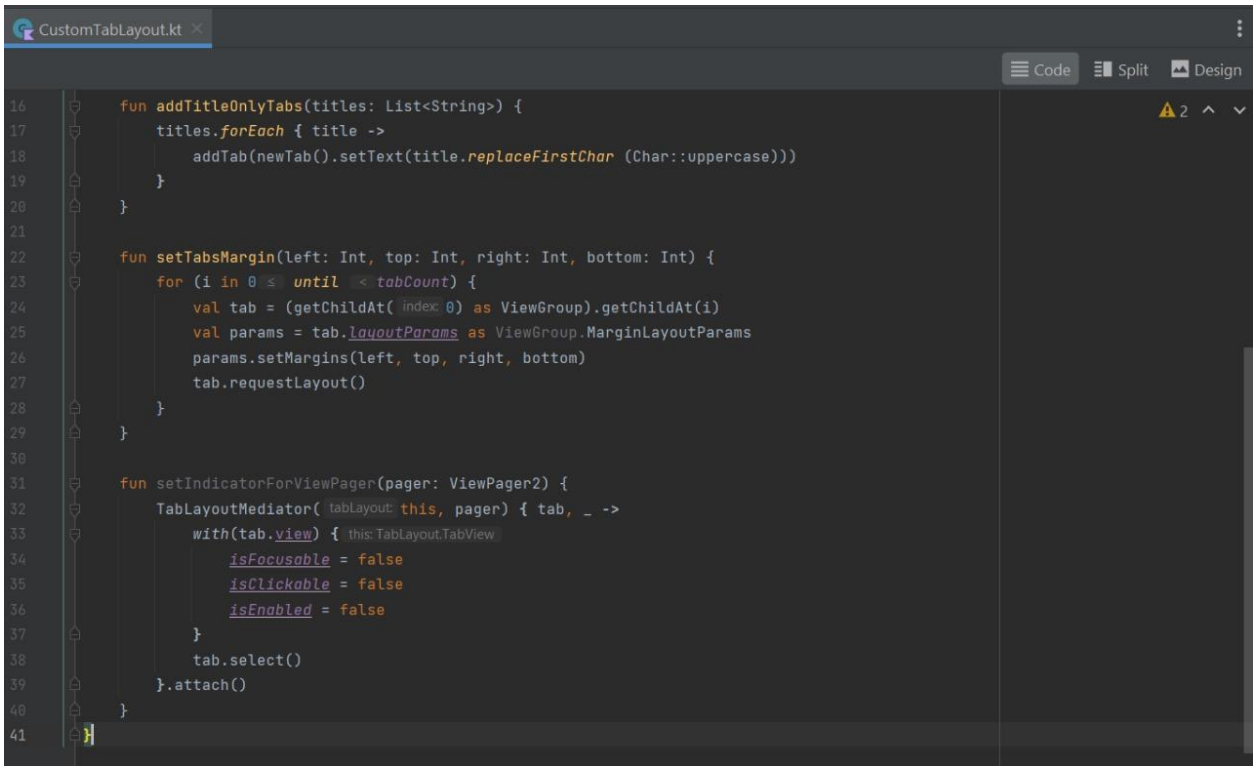
```kotlin
        val attributes = context.obtainStyledAttributes(attrs, R.styleable.CustomSearchBar)

        searchView.apply { this: SearchView
            queryHint = attributes.getString(R.styleable.CustomSearchBar_queryHint)
            queryHint = attributes.getString(R.styleable.CustomSearchBar_android_queryHint)
        }
        removeClearTextAnimation()

        val isUseAdditionalButton = attributes.getBoolean(R.styleable.CustomSearchBar_useAdditionalButton,  defValue: false)
        val btnSrc = attributes.getDrawable(R.styleable.CustomSearchBar_additionalButtonSrc)
        if (isUseAdditionalButton) {
            imageView.visibility = View.VISIBLE
            setAdditionalButtonImage(btnSrc)
        } else {
            imageView.visibility = View.GONE
        }

        attributes.recycle()
    }

    private fun removeClearTextAnimation() {
        val closeBtn = searchView.findViewById<ImageView>(androidx.appcompat.R.id.search_close_btn)
        closeBtn.background = null
    }

    private fun removeSearchIcon() {
        val icSearch = searchView.findViewById<ImageView>(androidx.appcompat.R.id.search_mag_icon)
        icSearch.layoutParams = LayoutParams( width: 0,  height: 0)
```





CODE 16: CustomTabLayout.kt





CODE 17: ExampleUnitTest.kt

ExampleUnitTest.kt | ExampleInstrumentedTest.kt

```kotlin
package com.aarush.newsapp
import org.junit.Test
import org.junit.Assert.*
/**
 * Example local unit test, which will execute on the development machine (host).
 *
 * See [testing documentation](http://d.android.com/tools/testing).
 */
class ExampleUnitTest {
    @Test
    fun addition_isCorrect() {
        assertEquals( expected: 4,  actual: 2 + 2)
    }
}
```

CODE 18: activity_articles.xml

activity_articles.xml

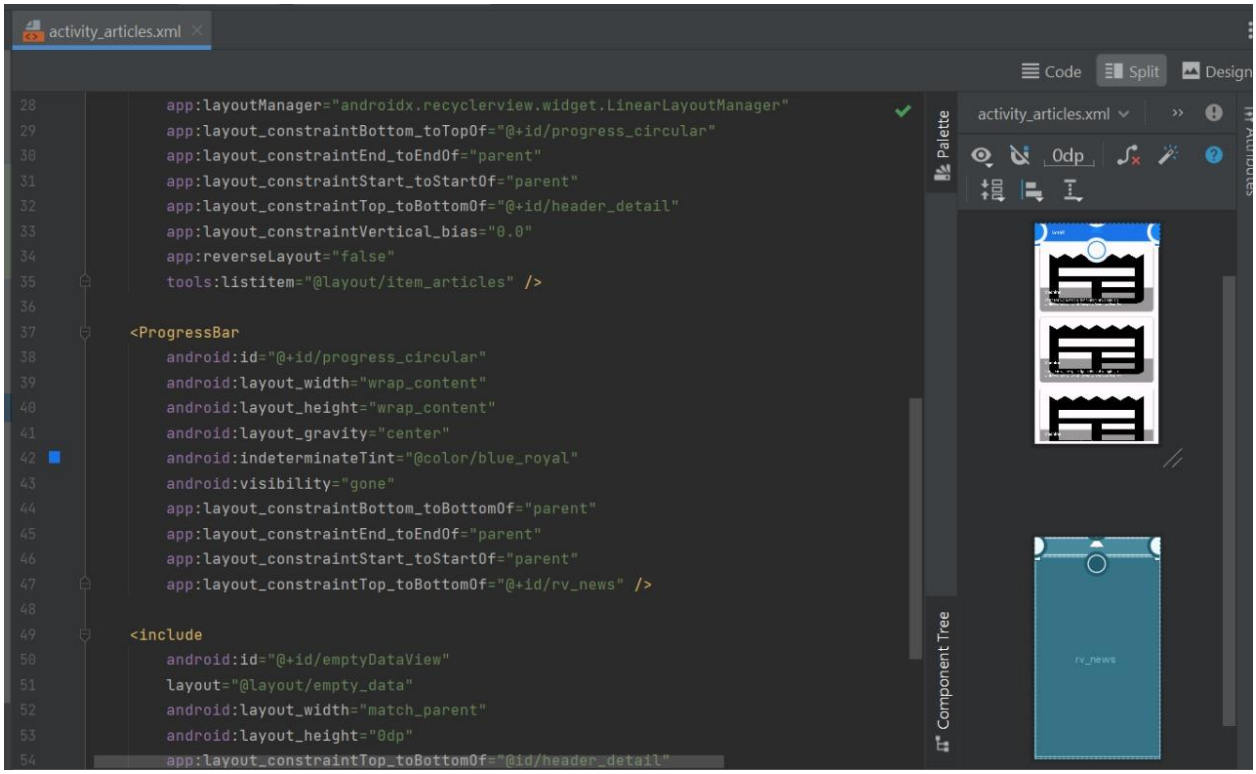
Code | Split | Desig

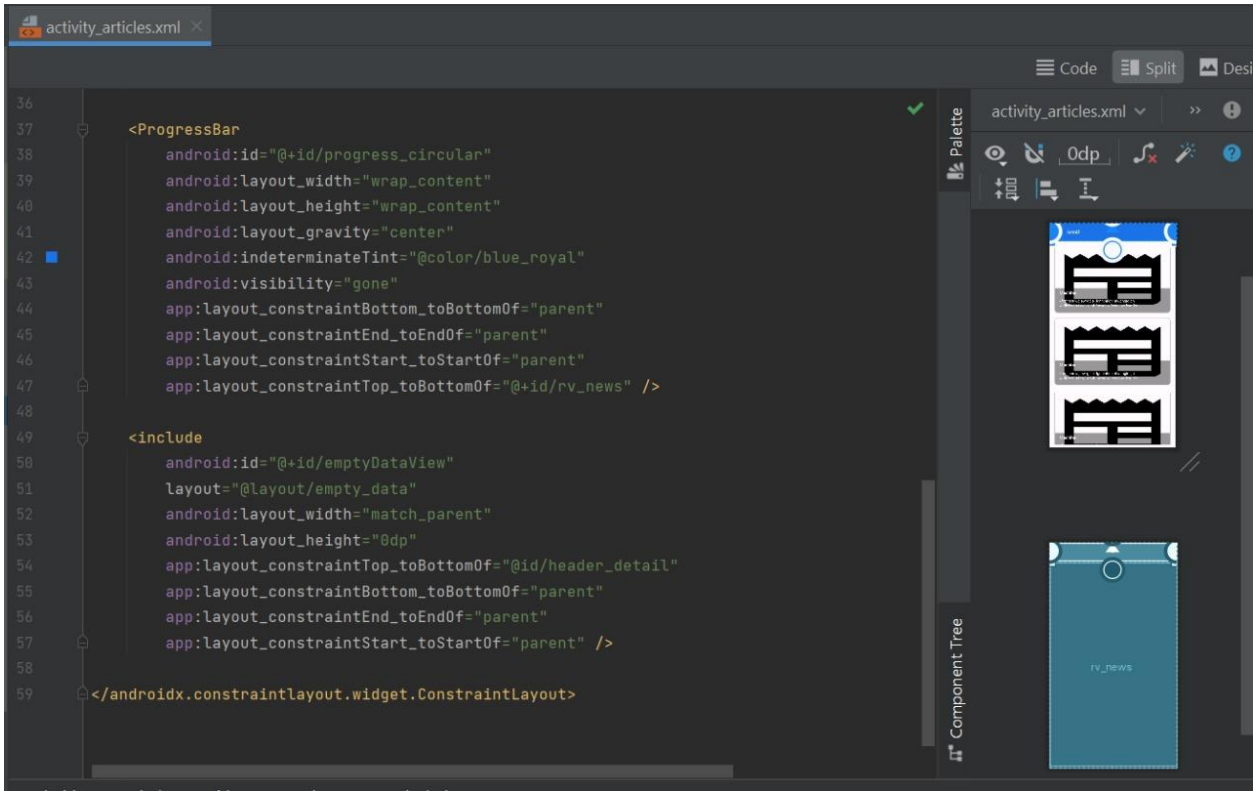
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.androiu.
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.articles.ArticlesActivity">

    <include
        android:id="@+id/header_detail"
        layout="@layout/header_app"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/rv_news"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:clipToPadding="false"
        android:orientation="vertical"
        android:visibility="visible"
```

Palette | Component Tree

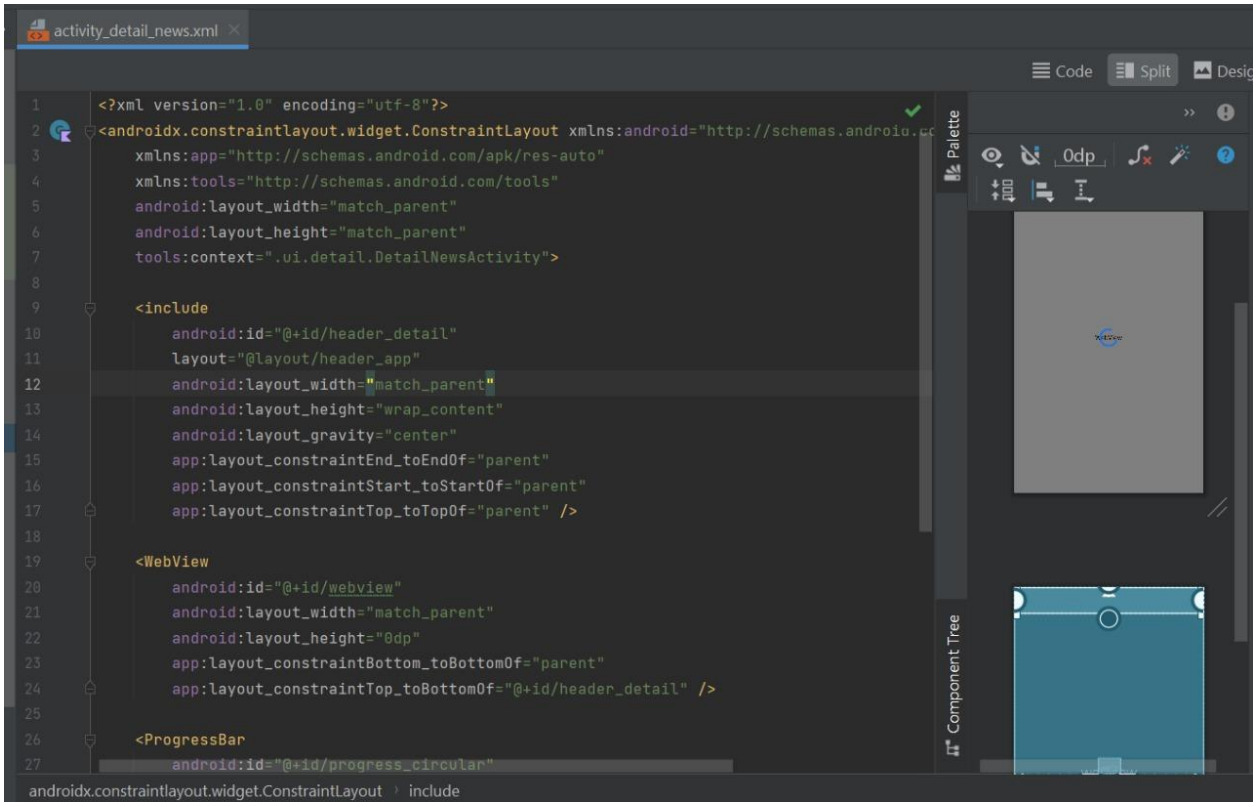
0dp

rv_news

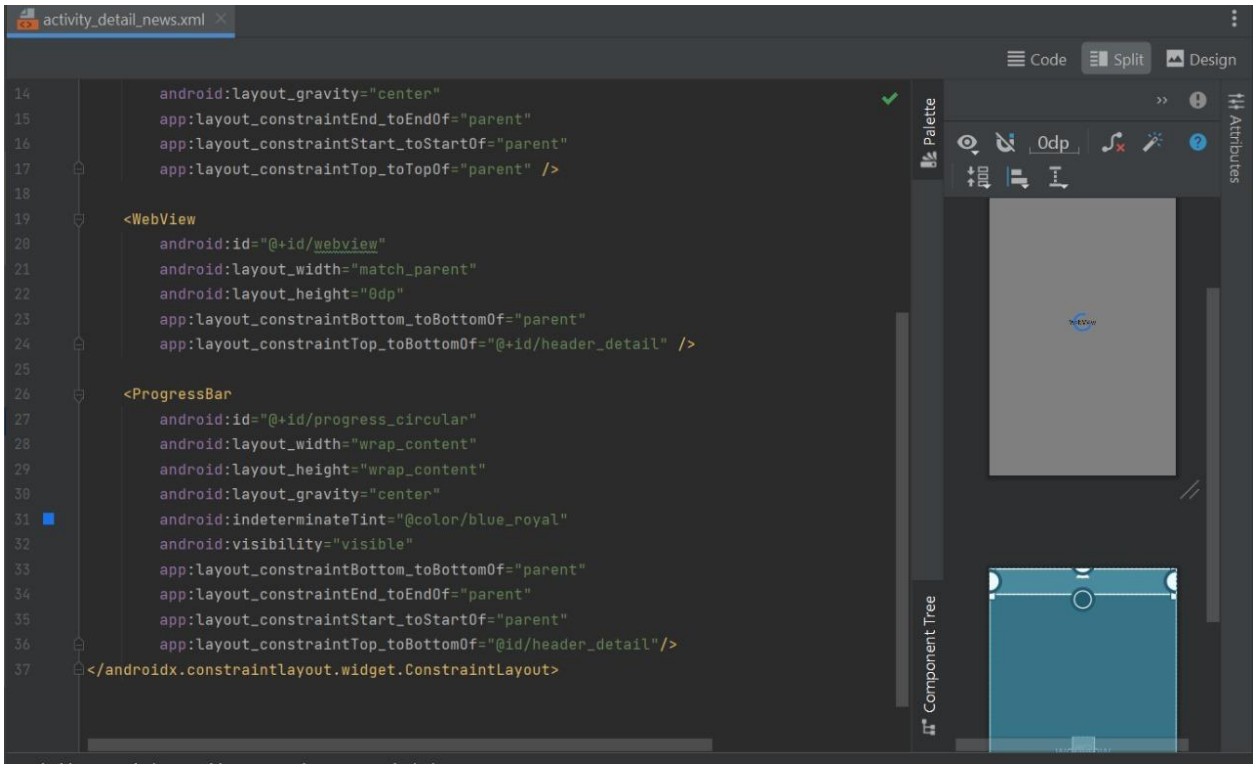




CODE 19: activity_detail_news.xml

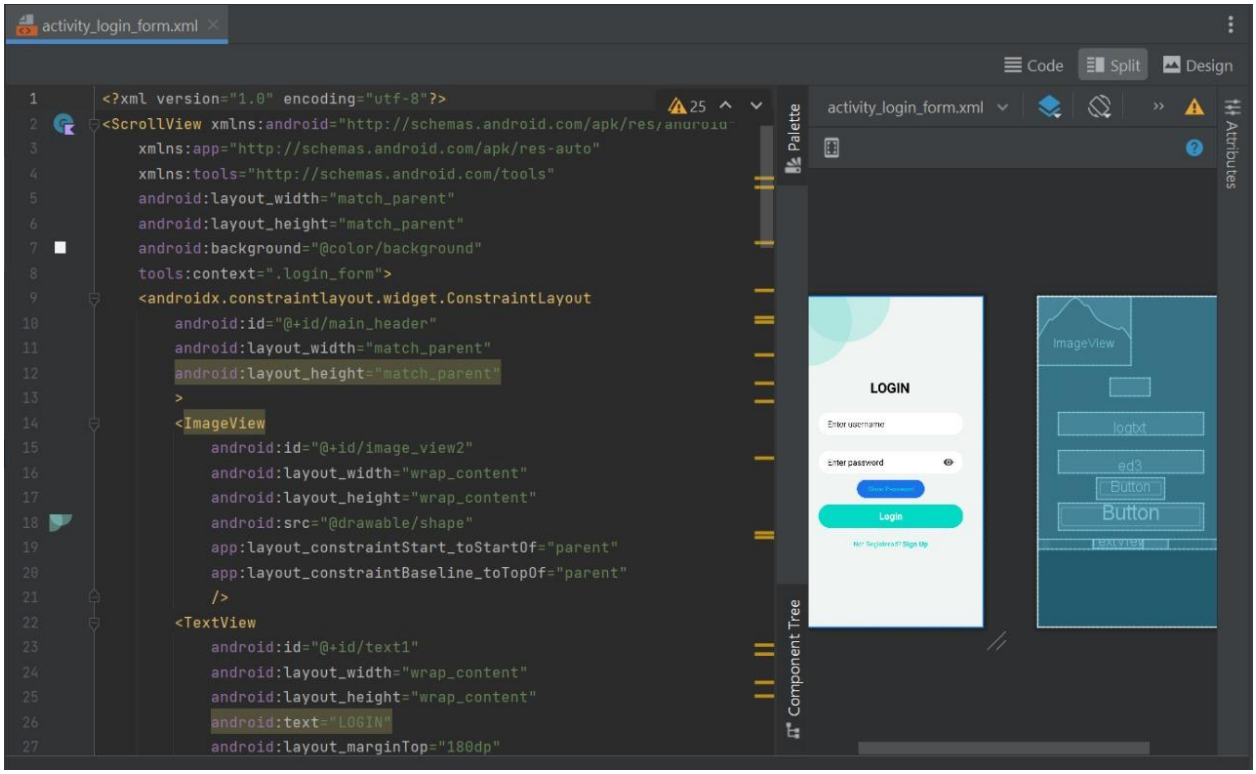


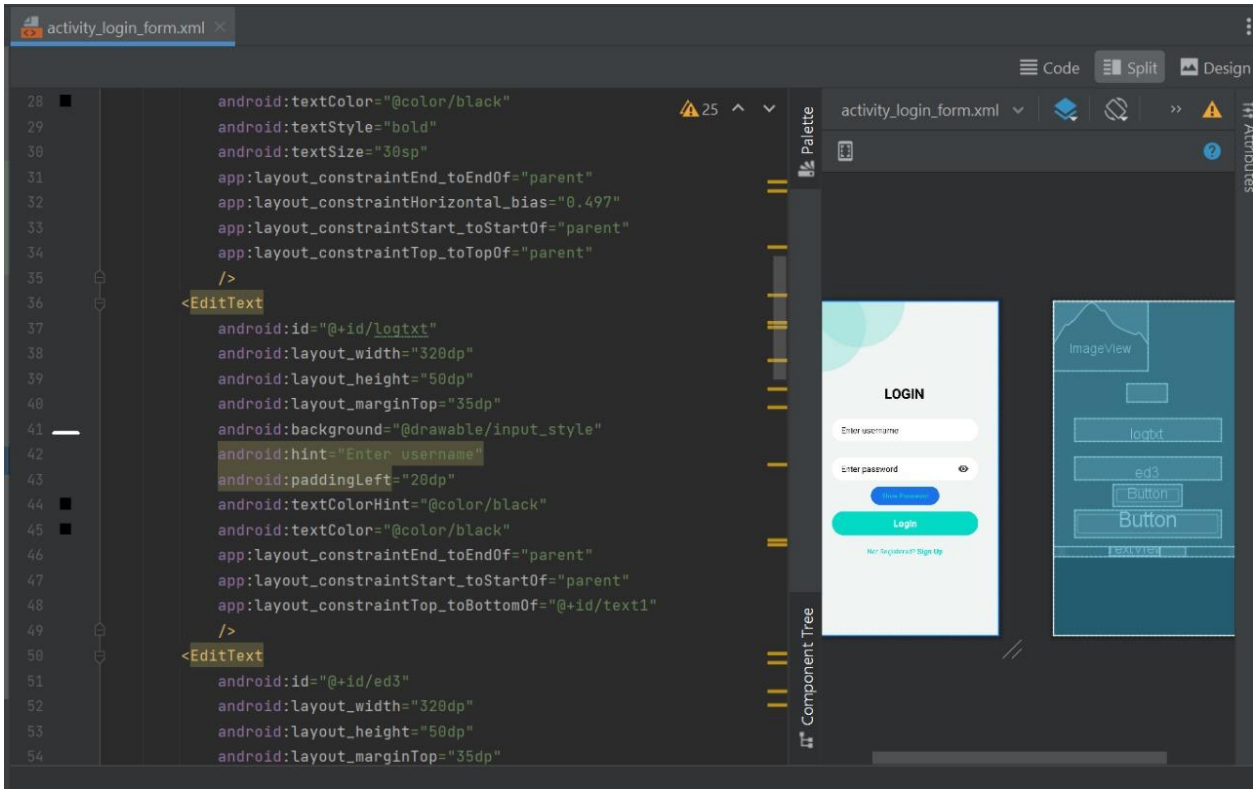

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.androi
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ui.detail.DetailNewsActivity">

    <include
        android:id="@+id/header_detail"
        layout="@layout/header_app"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <WebView
        android:id="@+id/webview"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/header_detail" />

    <ProgressBar
        android:id="@+id/progress_circular"
```

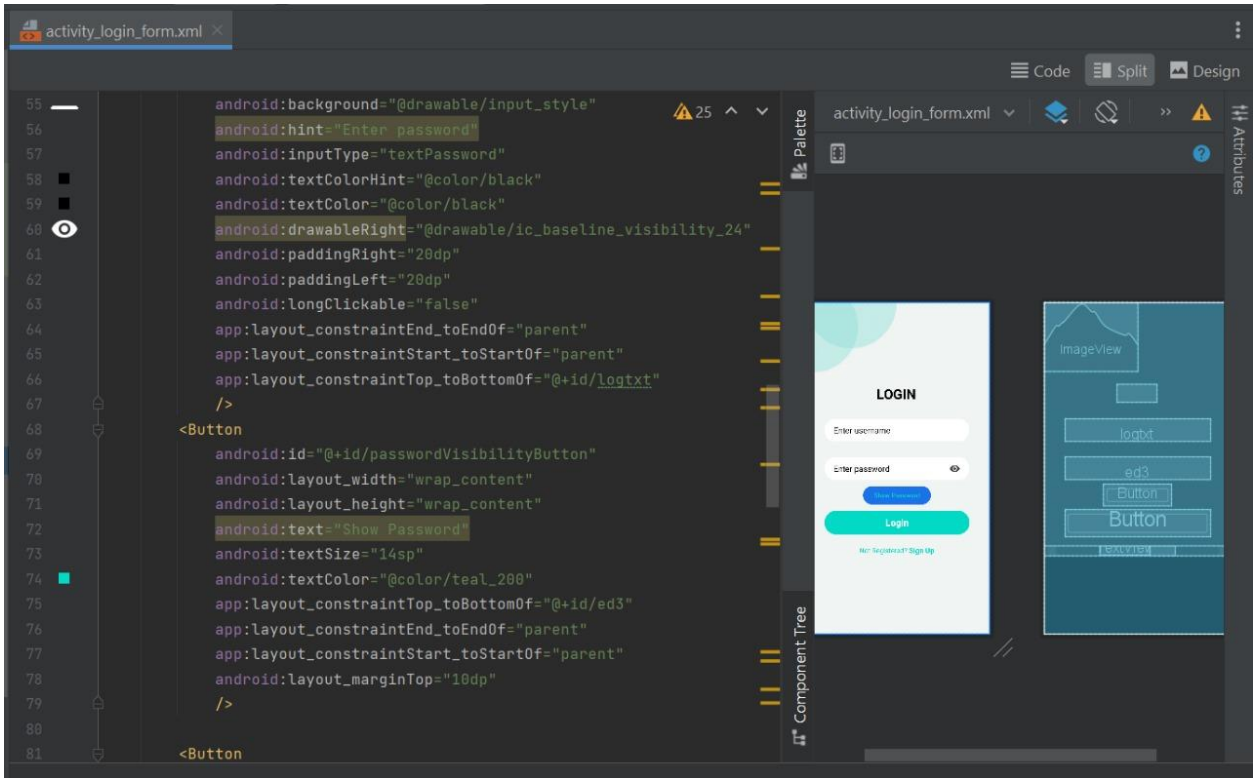


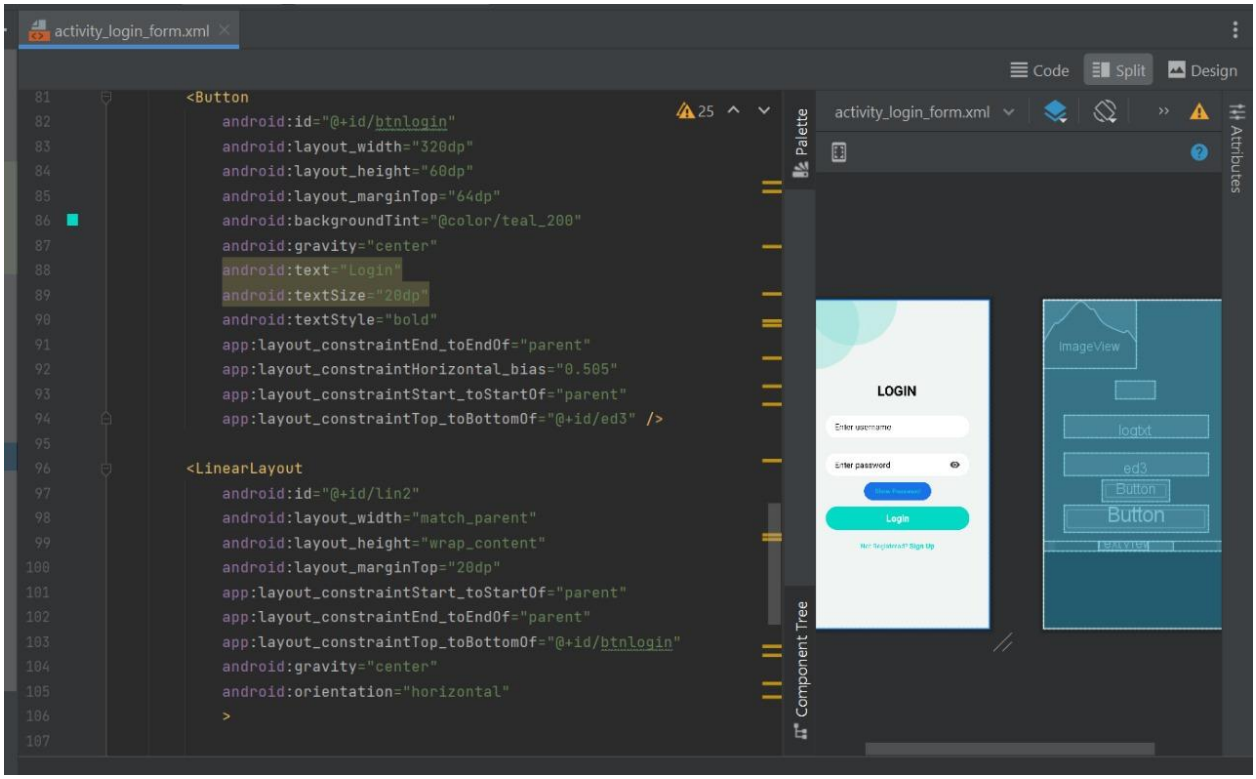

```
        android:layout_gravity="center"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <WebView
        android:id="@+id/webview"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/header_detail" />

    <ProgressBar
        android:id="@+id/progress_circular"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:indeterminateTint="@color/blue_royal"
        android:visibility="visible"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/header_detail"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

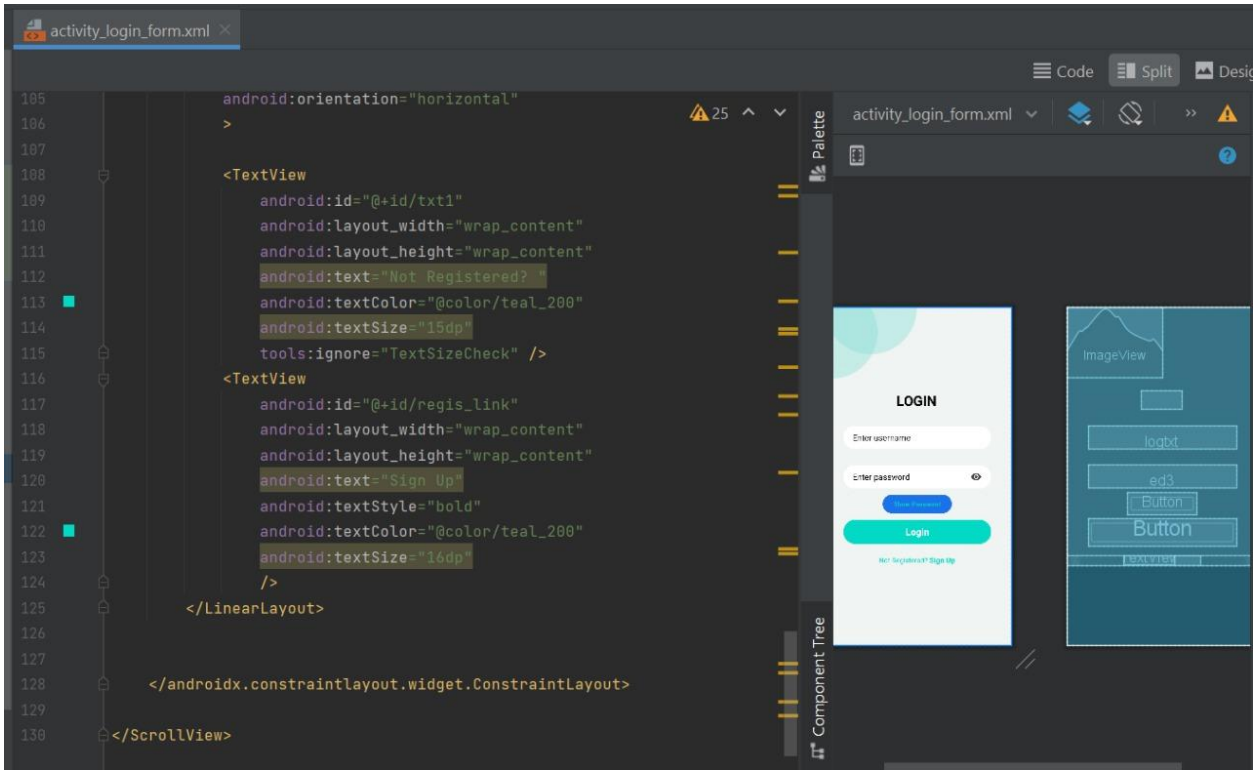
CODE 20: activity_login_form.xml


activity_login_form.xml
Code
Split
Design
<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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:background="@color/background"
tools:context=".login_form">
<androidx.constraintlayout.widget.ConstraintLayout
android:id="@+id/main_header"
android:layout_width="match_parent"
android:layout_height="match_parent"
>
<ImageView
android:id="@+id/image_view2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:src="@drawable/shape"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintBaseline_toTopOf="parent"
/>
<TextView
android:id="@+id/text1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="LOGIN"
android:layout_marginTop="180dp"
Palette
Component Tree
Attributes
LOGIN
Enter username
Enter password
Login
ImageView
logtxt
ed3
Button
Button

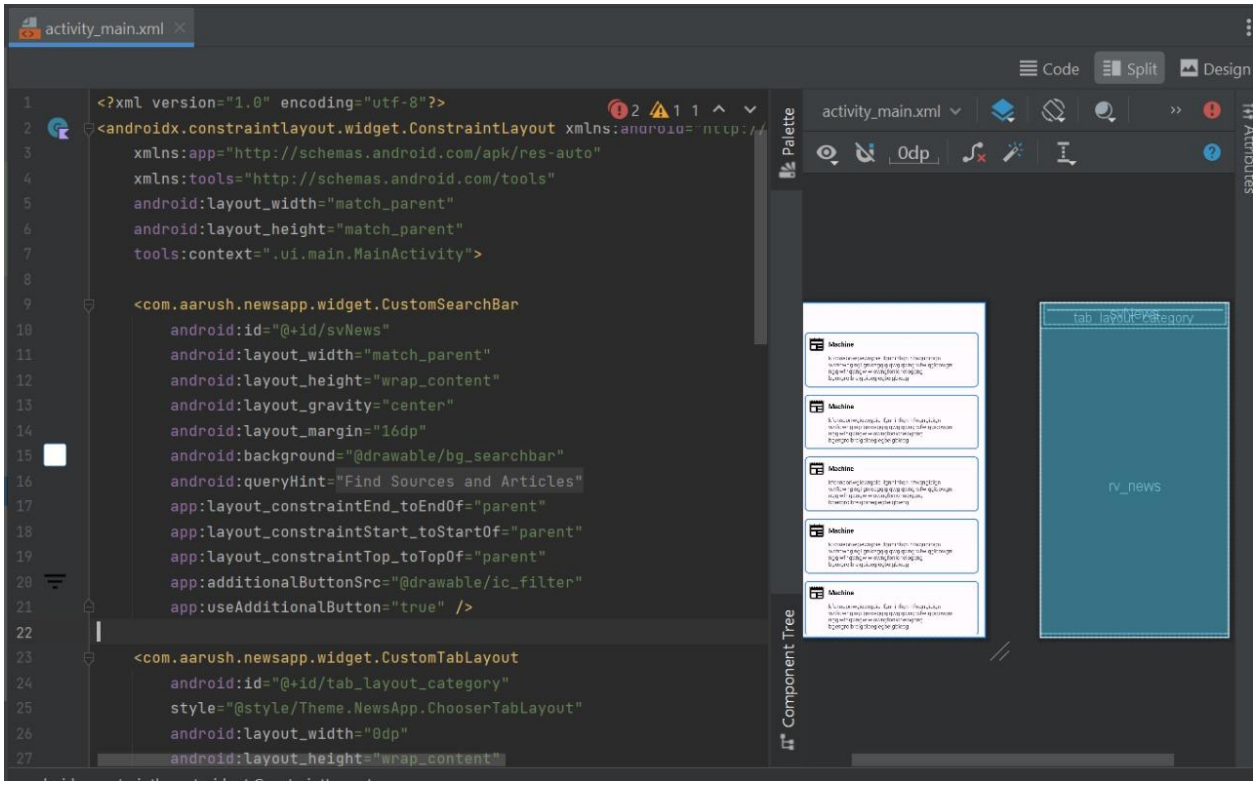

activity_login_form.xml
Code
Split
Design
android:textColor="@color/black"
android:textStyle="bold"
android:textSize="30sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.497"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
/>
<EditText
android:id="@+id/logtxt"
android:layout_width="320dp"
android:layout_height="50dp"
android:layout_marginTop="35dp"
android:background="@drawable/input_style"
android:hint="Enter username"
android:paddingLeft="20dp"
android:textColorHint="@color/black"
android:textColor="@color/black"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/text1"
/>
<EditText
android:id="@+id/ed3"
android:layout_width="320dp"
android:layout_height="50dp"
android:layout_marginTop="35dp"
Palette
Component Tree
Attributes
LOGIN
Enter username
Enter password
Login
ImageView
logtxt
ed3
Button
Button

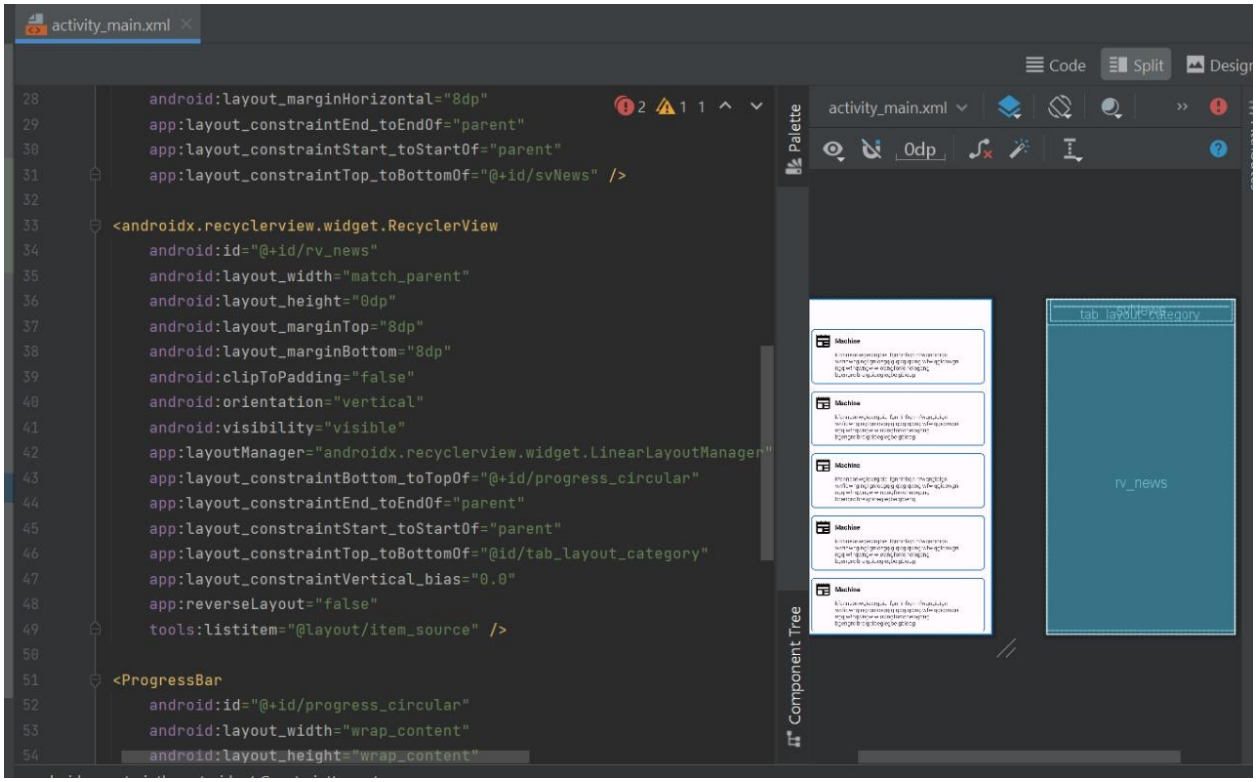

activity_login_form.xml
Code
Split
Design
android:background="@drawable/input_style"
android:hint="Enter password"
android:inputType="textPassword"
android:textColorHint="@color/black"
android:textColor="@color/black"
android:drawableRight="@drawable/ic_baseline_visibility_24"
android:paddingRight="20dp"
android:paddingLeft="20dp"
android:longClickable="false"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/logtxt"
/>
<Button
android:id="@+id/passwordVisibilityButton"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Show Password"
android:textSize="14sp"
android:textColor="@color/teal_200"
app:layout_constraintTop_toBottomOf="@+id/ed3"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
android:layout_marginTop="10dp"
/>
<Button
Palette
Component Tree
Attributes
LOGIN
Enter username
Enter password
Login
ImageView
logtxt
ed3
Button
Button

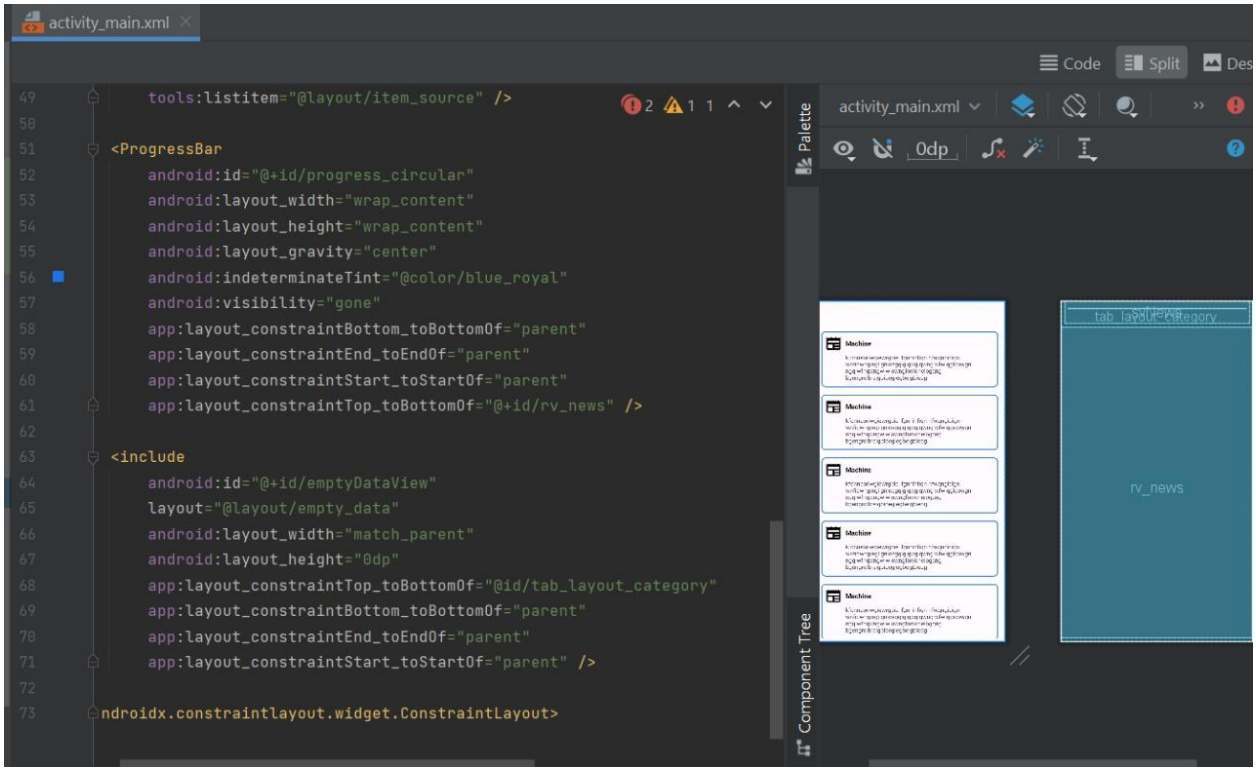

activity_login_form.xml
Code
Split
Design
<Button
android:id="@+id/btnlogin"
android:layout_width="320dp"
android:layout_height="60dp"
android:layout_marginTop="64dp"
android:backgroundTint="@color/teal_200"
android:gravity="center"
android:text="Login"
android:textSize="20dp"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.505"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/ed3" />
<LinearLayout
android:id="@+id/lin2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btnlogin"
android:gravity="center"
android:orientation="horizontal"
>
Palette
Component Tree
Attributes
LOGIN
Enter username
Enter password
Login
ImageView
logtxt
ed3
Button
Button



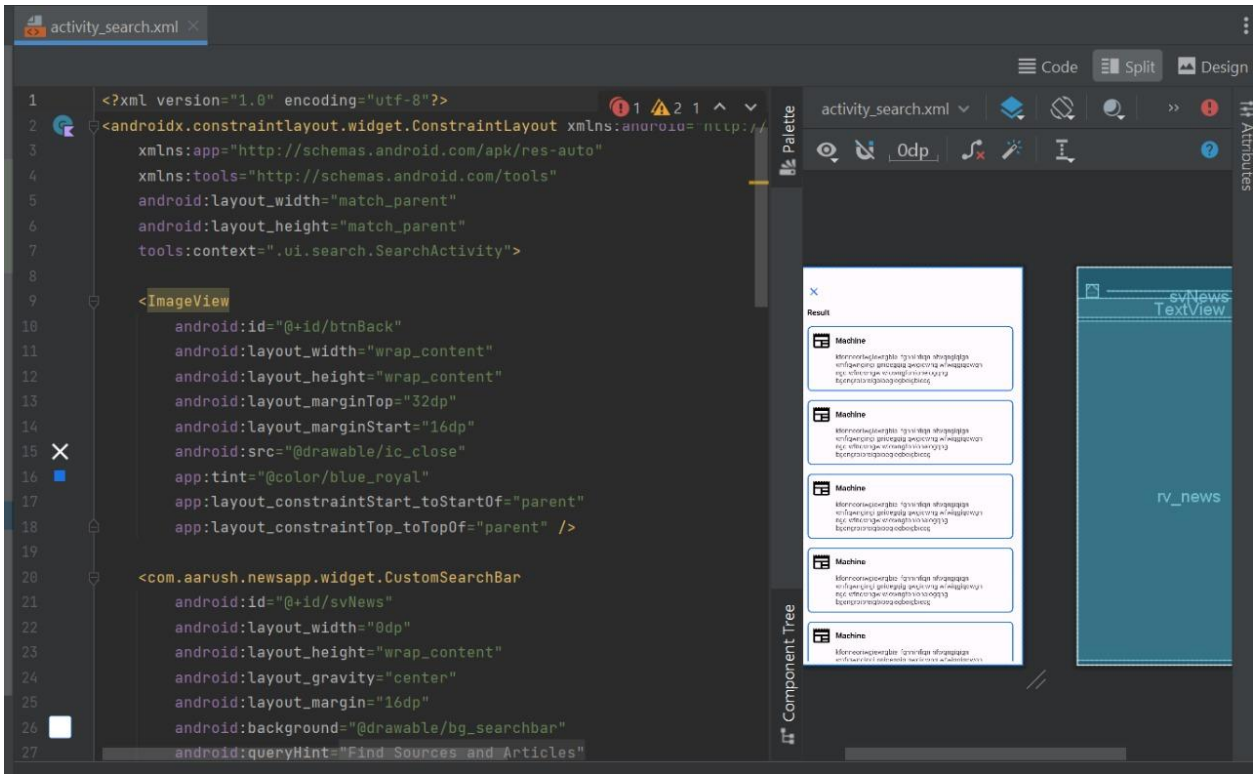
CODE 21: activity_main.xml

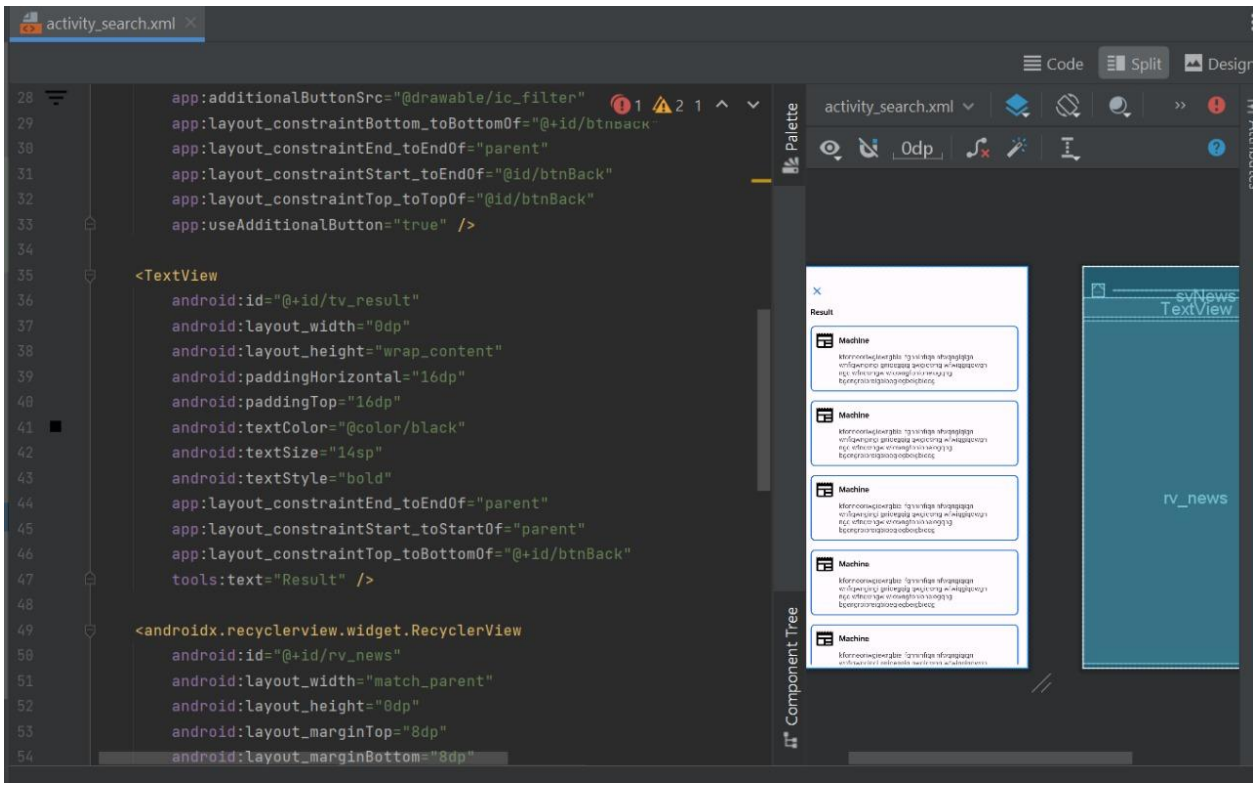


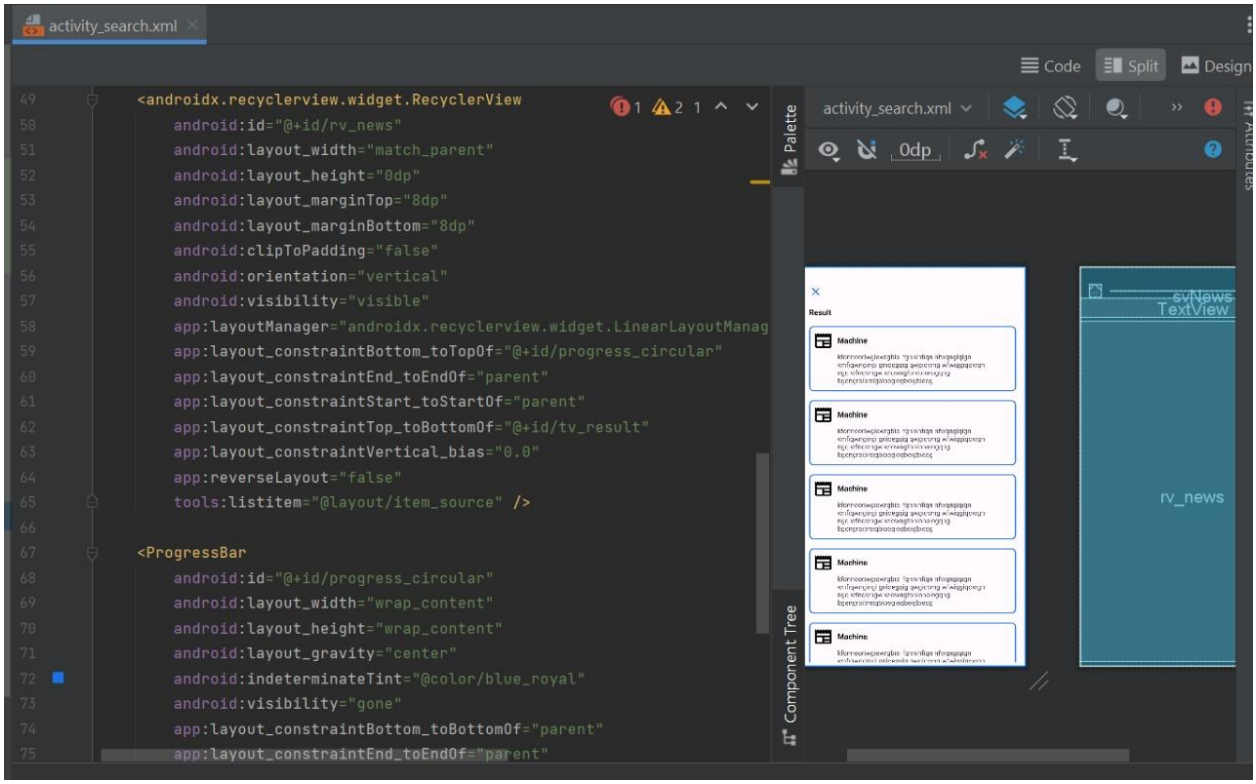


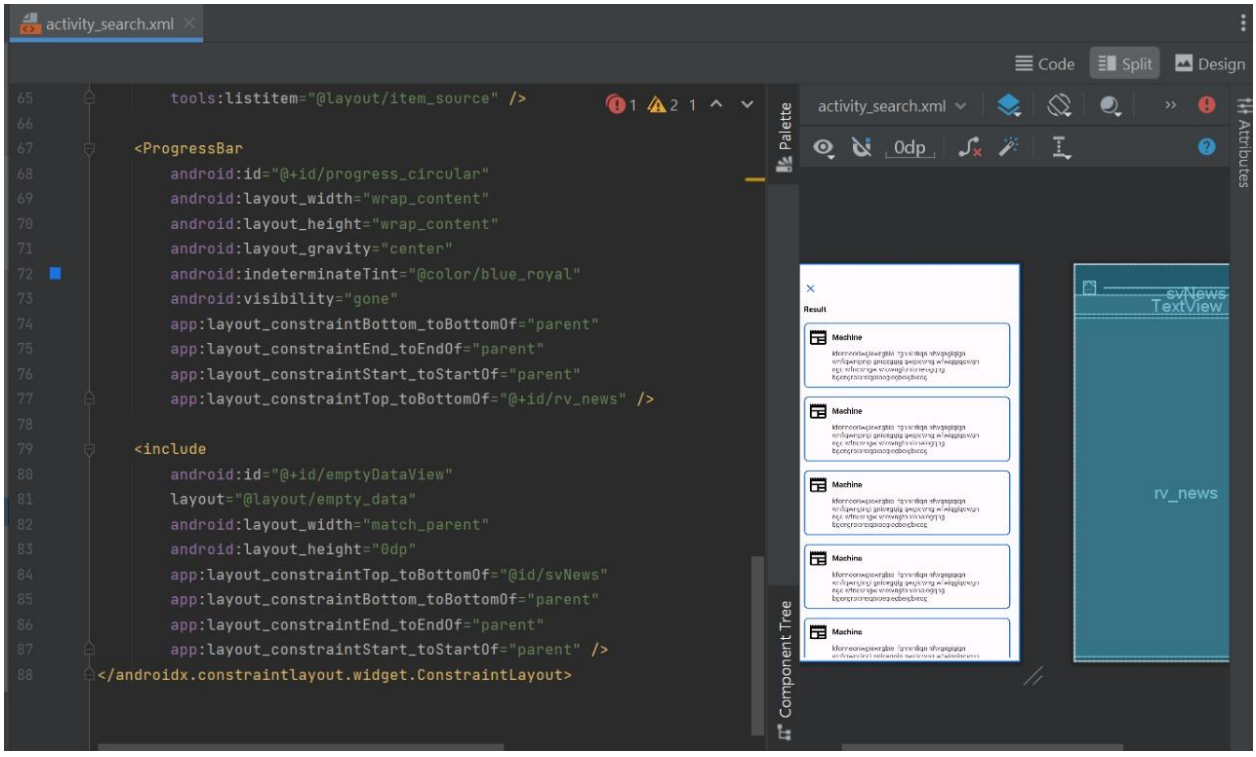


CODE 22: activity_search.xml

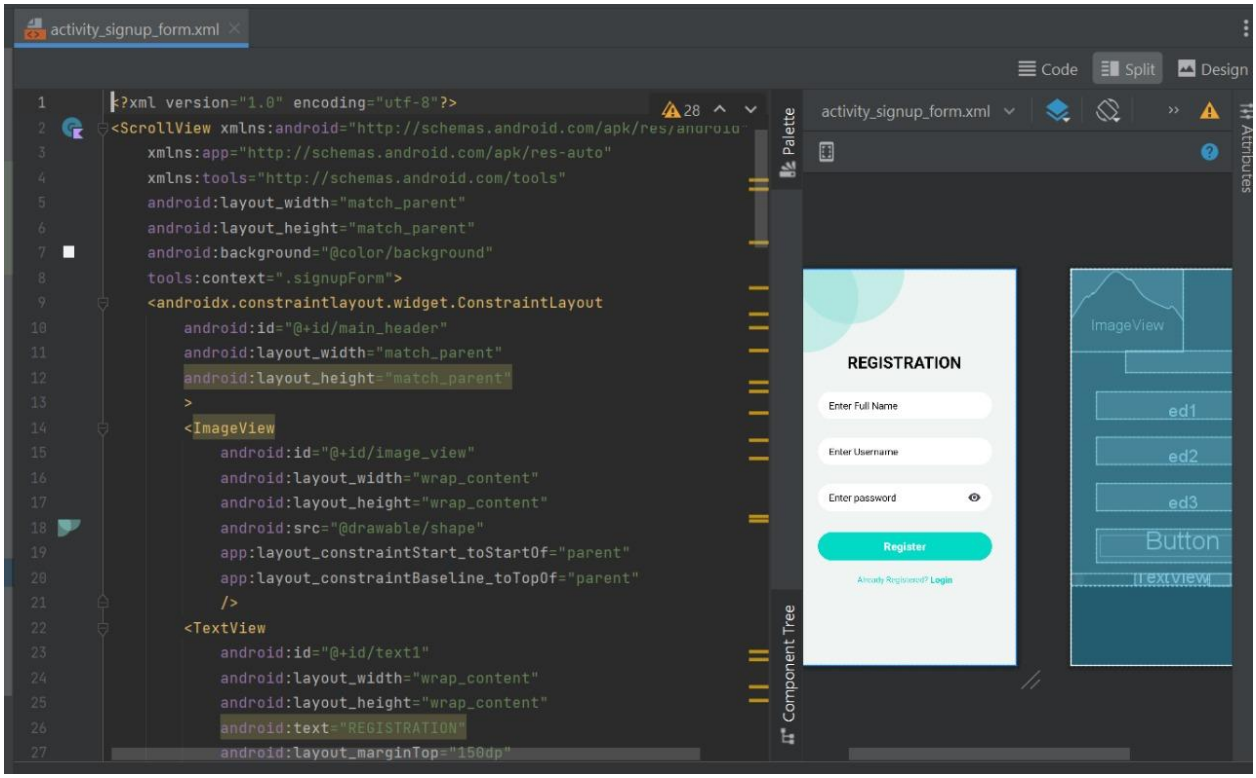

activity_search.xml
Code
Split
Design
Palette
Component Tree
Attributes
0dp
Result
Machine
svNews
TextView
rv_news
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".ui.search.SearchActivity">
<ImageView
android:id="@+id/btnBack"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="32dp"
android:layout_marginStart="16dp"
android:src="@drawable/ic_close"
app:tint="@color/blue_royal"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
<com.aarush.newsapp.widget.CustomSearchBar
android:id="@+id/svNews"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:layout_margin="16dp"
android:background="@drawable/bg_searchbar"
android:queryHint="Find Sources and Articles"


activity_search.xml
Code
Split
Palette
Component Tree
0dp
Result
Machine
svNews
TextView
rv_news
app:additionalButtonSrc="@drawable/ic_filter"
app:layout_constraintBottom_toBottomOf="@+id/btnBack"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toEndOf="@id/btnBack"
app:layout_constraintTop_toTopOf="@id/btnBack"
app:useAdditionalButton="true" />
<TextView
android:id="@+id/tv_result"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:paddingHorizontal="16dp"
android:paddingTop="16dp"
android:textColor="@color/black"
android:textSize="14sp"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btnBack"
tools:text="Result" />
<androidx.recyclerview.widget.RecyclerView
android:id="@+id/rv_news"
android:layout_width="match_parent"
android:layout_height="0dp"
android:layout_marginTop="8dp"
android:layout_marginBottom="8dp"

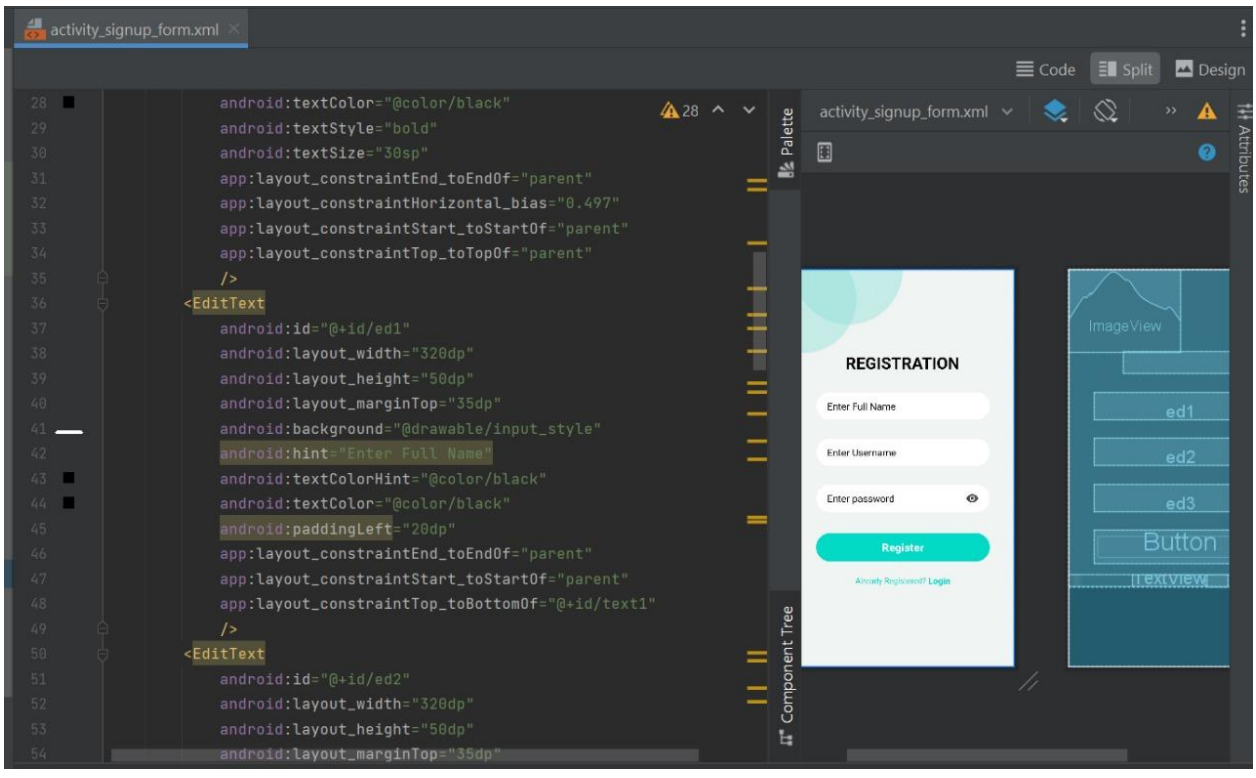




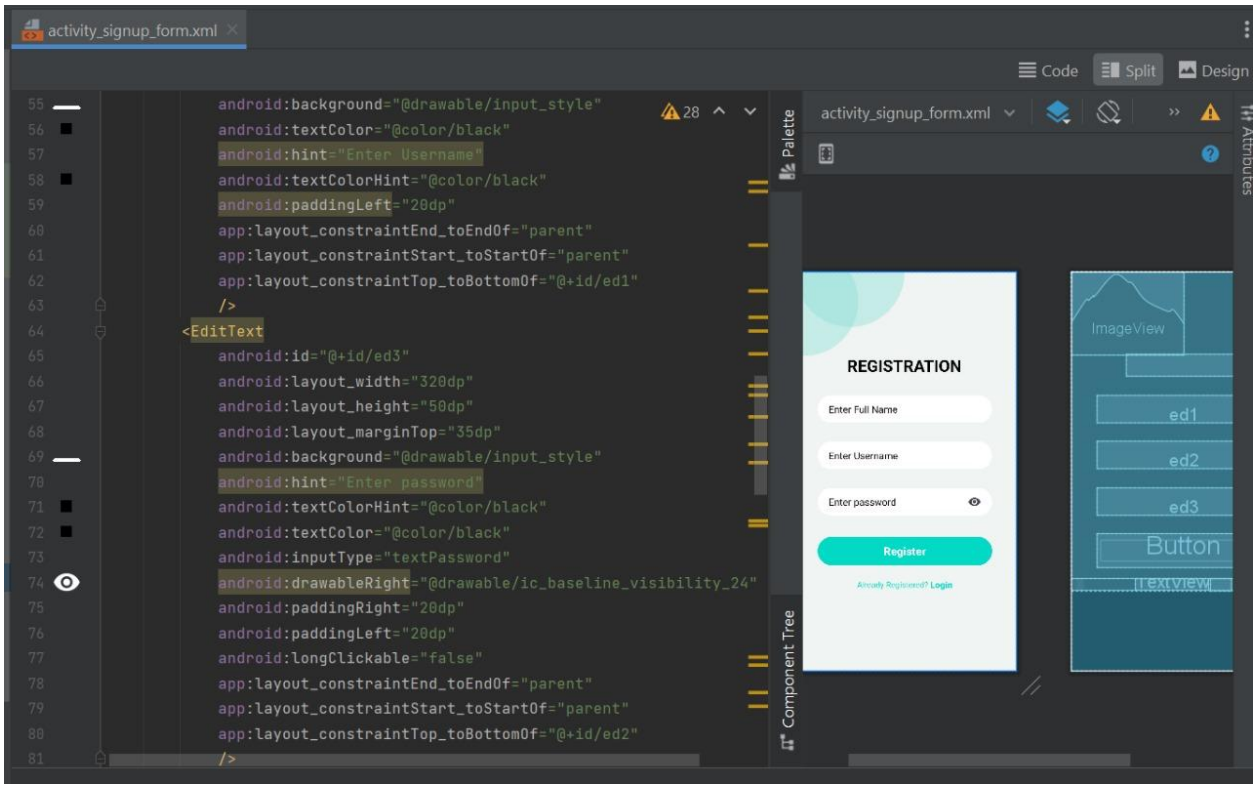
CODE 23: activity_signup_form.xml

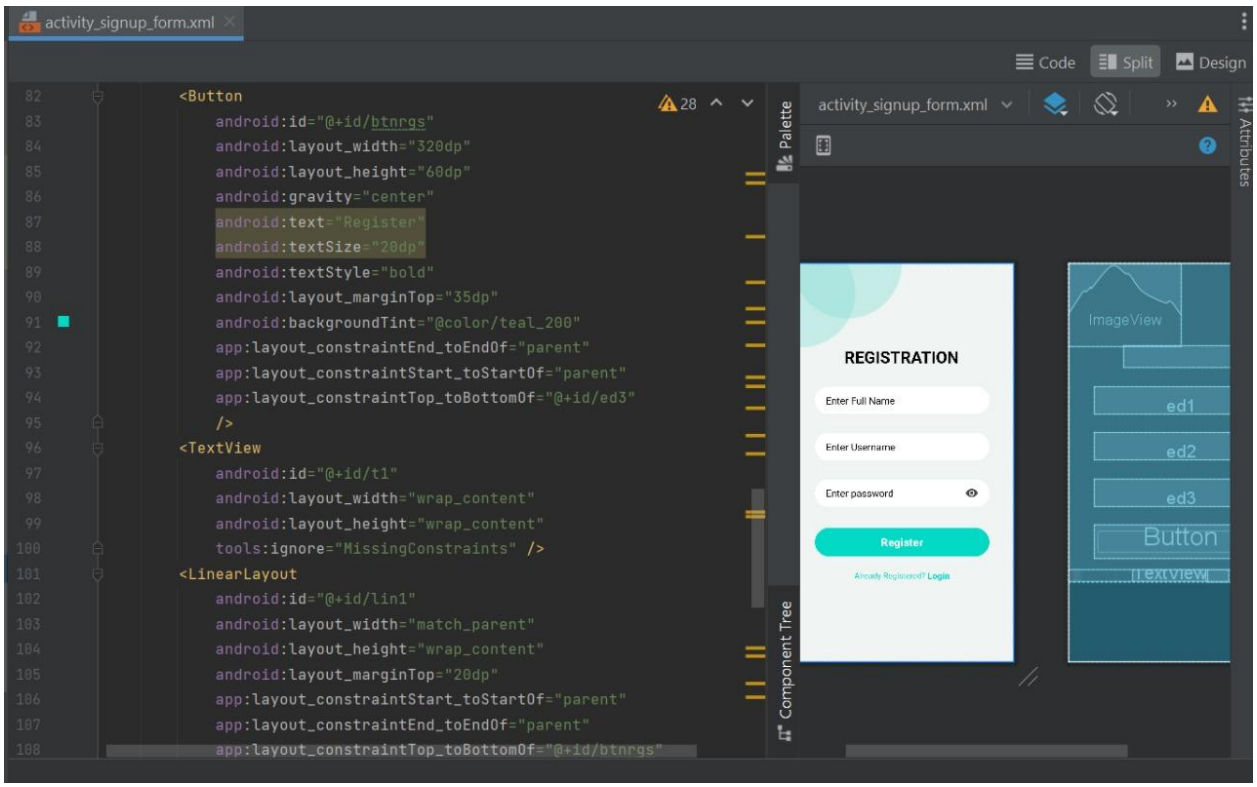


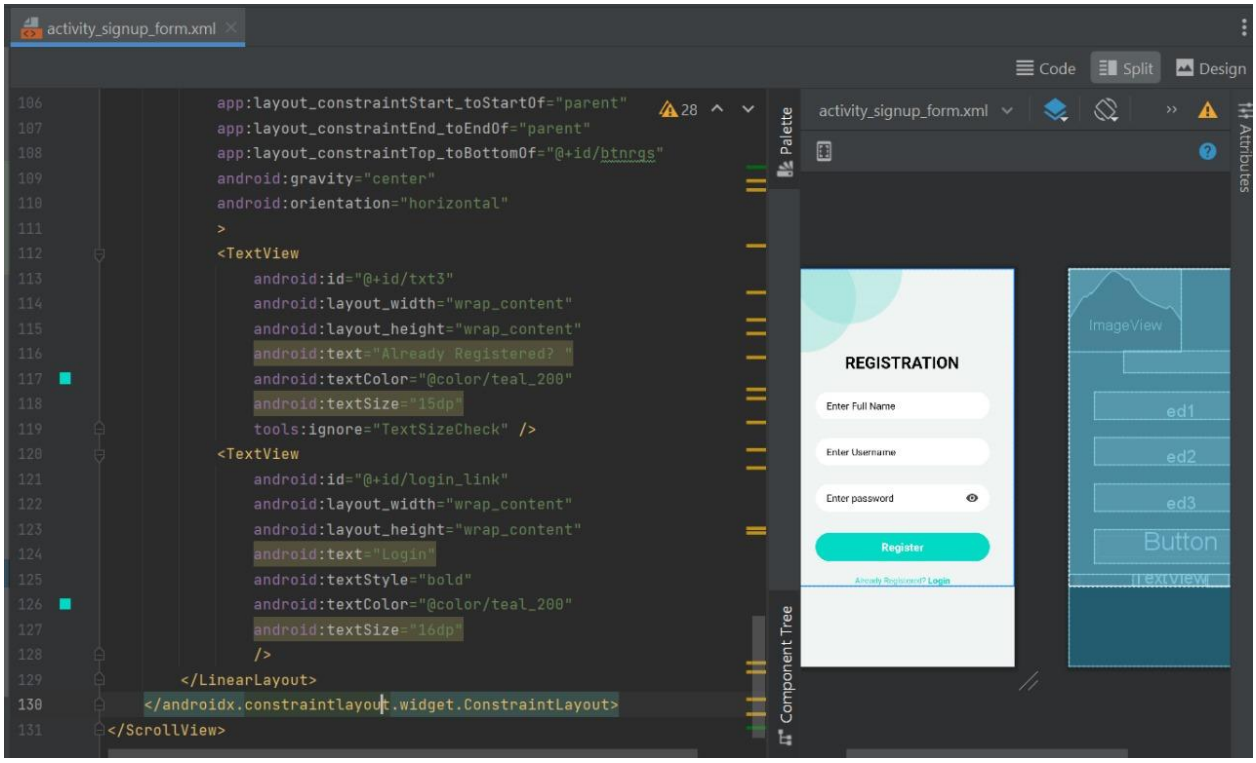

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/background"
    tools:context=".signupForm">
    <androidx.constraintlayout.widget.ConstraintLayout
        android:id="@+id/main_header"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        >
        <ImageView
            android:id="@+id/image_view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@drawable/shape"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintBaseline_toTopOf="parent"
            />
        <TextView
            android:id="@+id/text1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="REGISTRATION"
            android:layout_marginTop="150dp"
```



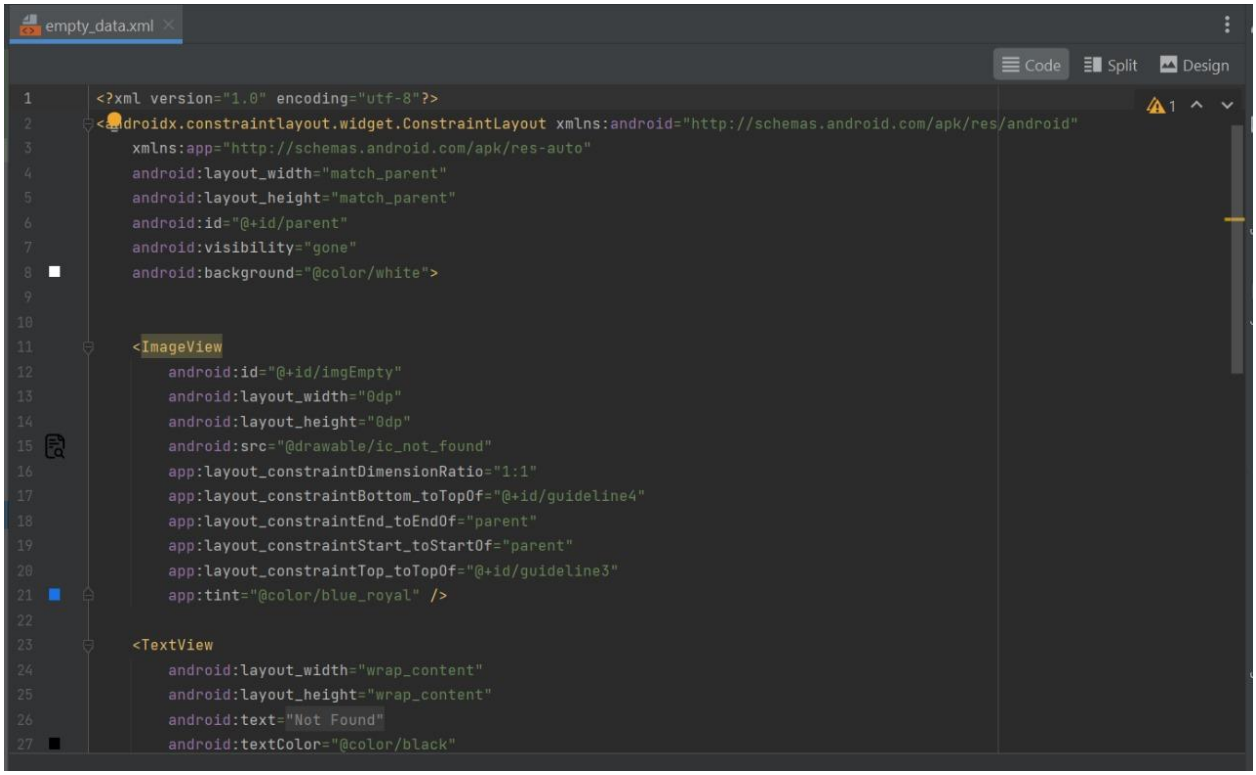

```
            android:textColor="@color/black"
            android:textStyle="bold"
            android:textSize="30sp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintHorizontal_bias="0.497"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent"
            />
        <EditText
            android:id="@+id/ed1"
            android:layout_width="320dp"
            android:layout_height="50dp"
            android:layout_marginTop="35dp"
            android:background="@drawable/input_style"
            android:hint="Enter Full Name"
            android:textColorHint="@color/black"
            android:textColor="@color/black"
            android:paddingLeft="20dp"
            app:layout_constraintEnd_toEndOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toBottomOf="@+id/text1"
            />
        <EditText
            android:id="@+id/ed2"
            android:layout_width="320dp"
            android:layout_height="50dp"
            android:layout_marginTop="35dp"
```


activity_signup_form.xml
Code
Split
Design
android:background="@drawable/input_style"
android:textColor="@color/black"
android:hint="Enter Username"
android:textColorHint="@color/black"
android:paddingLeft="20dp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/ed1"
/>
<EditText
android:id="@+id/ed3"
android:layout_width="320dp"
android:layout_height="50dp"
android:layout_marginTop="35dp"
android:background="@drawable/input_style"
android:hint="Enter password"
android:textColorHint="@color/black"
android:textColor="@color/black"
android:inputType="textPassword"
android:drawableRight="@drawable/ic_baseline_visibility_24"
android:paddingRight="20dp"
android:paddingLeft="20dp"
android:longClickable="false"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/ed2"
/>
Palette
Component Tree
Attributes
REGISTRATION
Enter Full Name
Enter Username
Enter password
Register
Already Registered? Login
ImageView
ed1
ed2
ed3
Button
TextView


activity_signup_form.xml
Code
Split
Design
<Button
android:id="@+id/btnrgs"
android:layout_width="320dp"
android:layout_height="60dp"
android:gravity="center"
android:text="Register"
android:textSize="20dp"
android:textStyle="bold"
android:layout_marginTop="35dp"
android:backgroundTint="@color/teal_200"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/ed3"
/>
<TextView
android:id="@+id/t1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
tools:ignore="MissingConstraints" />
<LinearLayout
android:id="@+id/lin1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btnrgs"
Palette
Component Tree
Attributes
REGISTRATION
Enter Full Name
Enter Username
Enter password
Register
Already Registered? Login
ImageView
ed1
ed2
ed3
Button
TextView



CODE 24: empty_data.xml



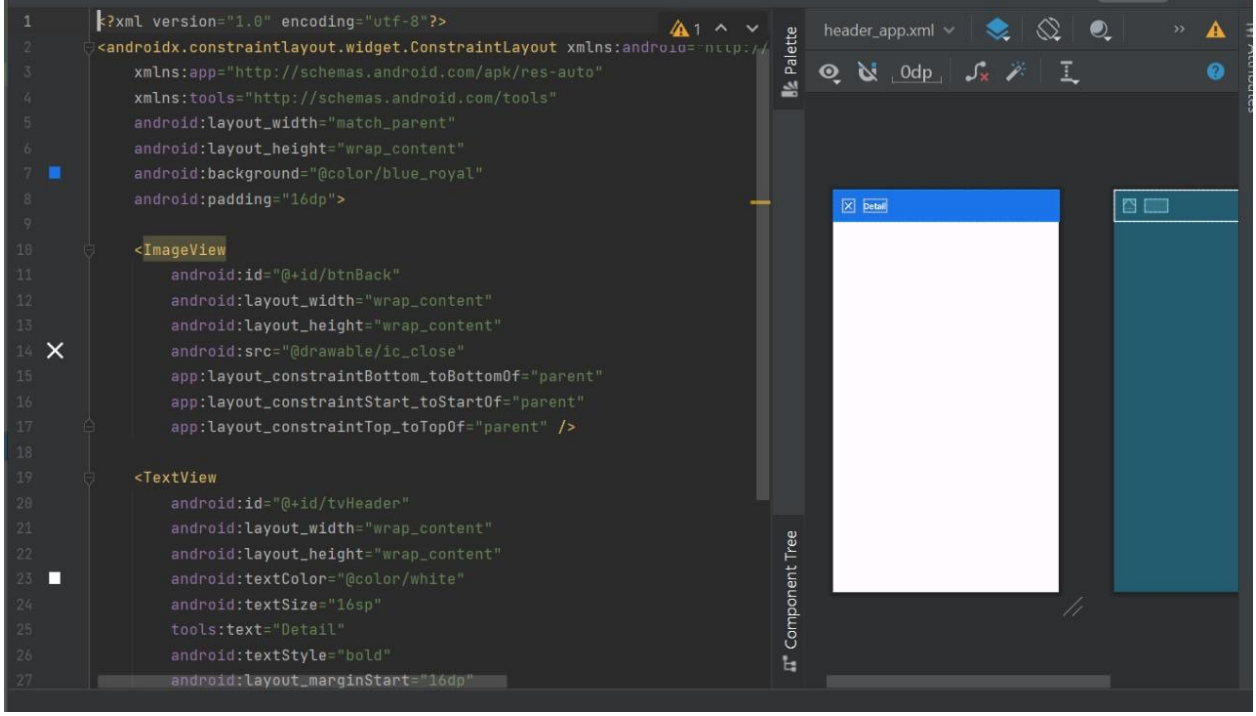
empty_data.xml

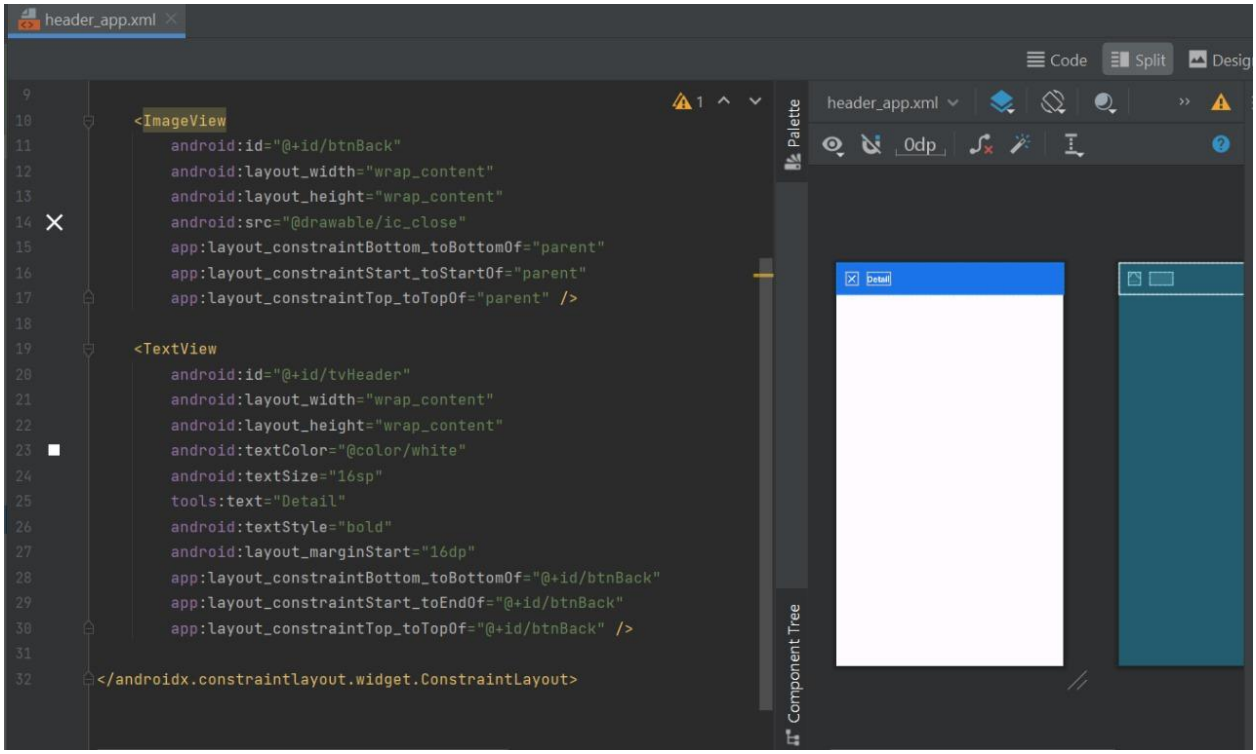
```
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.60" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_percent="0.4" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.35" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_percent="0.55" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

CODE 25: header_app.xml

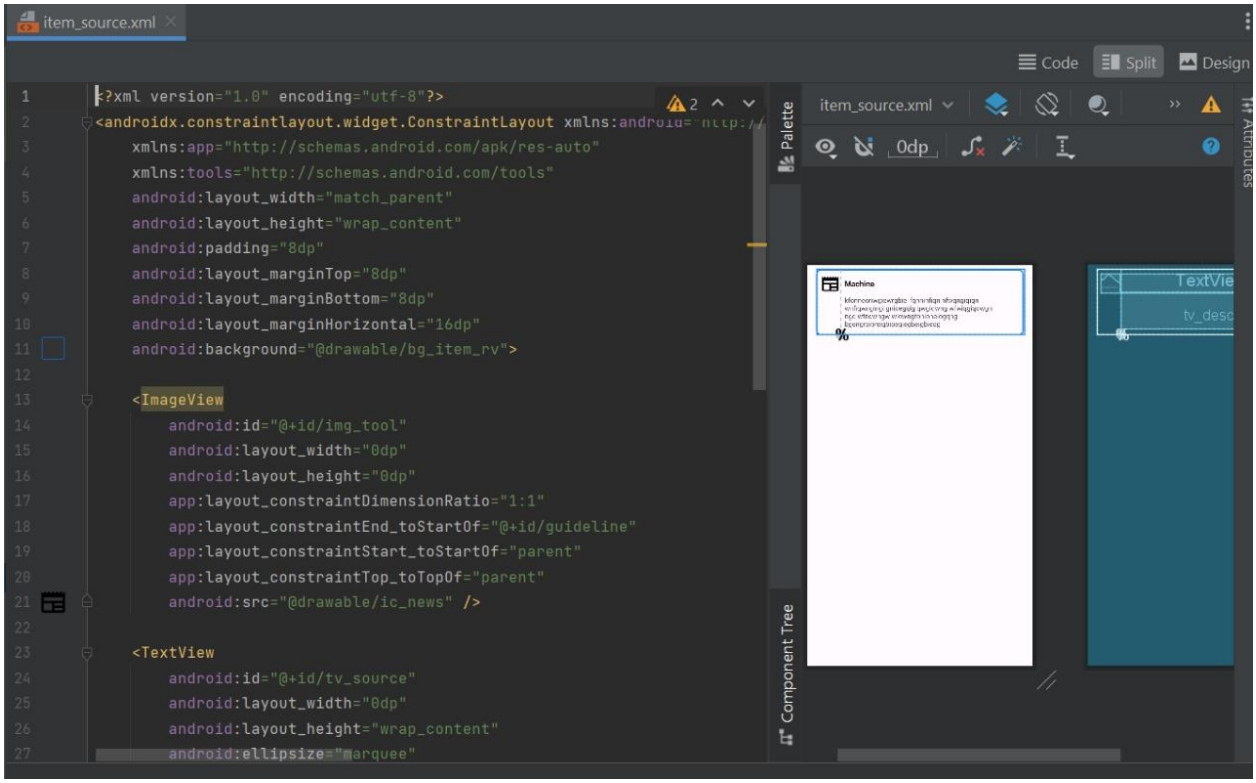
header_app.xml

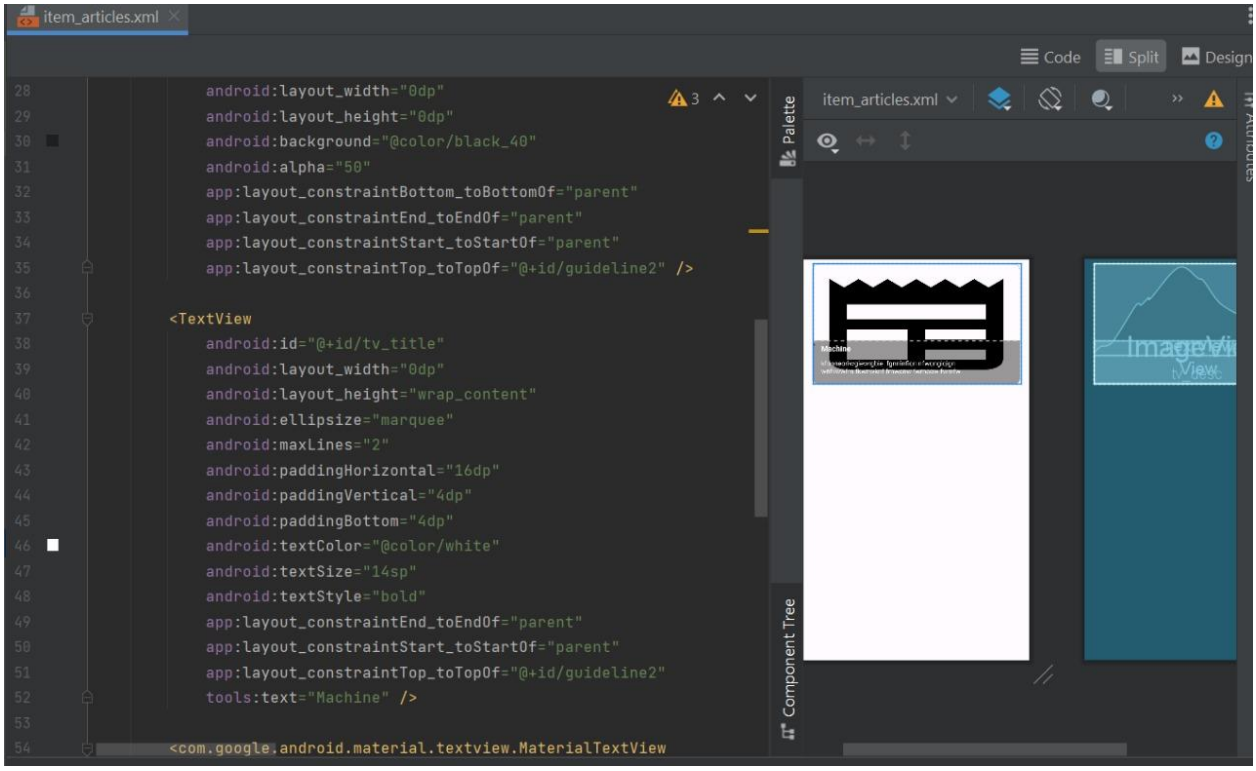
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/blue_royal"
    android:padding="16dp">

    <ImageView
        android:id="@+id/btnBack"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_close"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/tvHeader"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/white"
        android:textSize="16sp"
        tools:text="Detail"
        android:textStyle="bold"
        android:layout_marginStart="16dp"
```

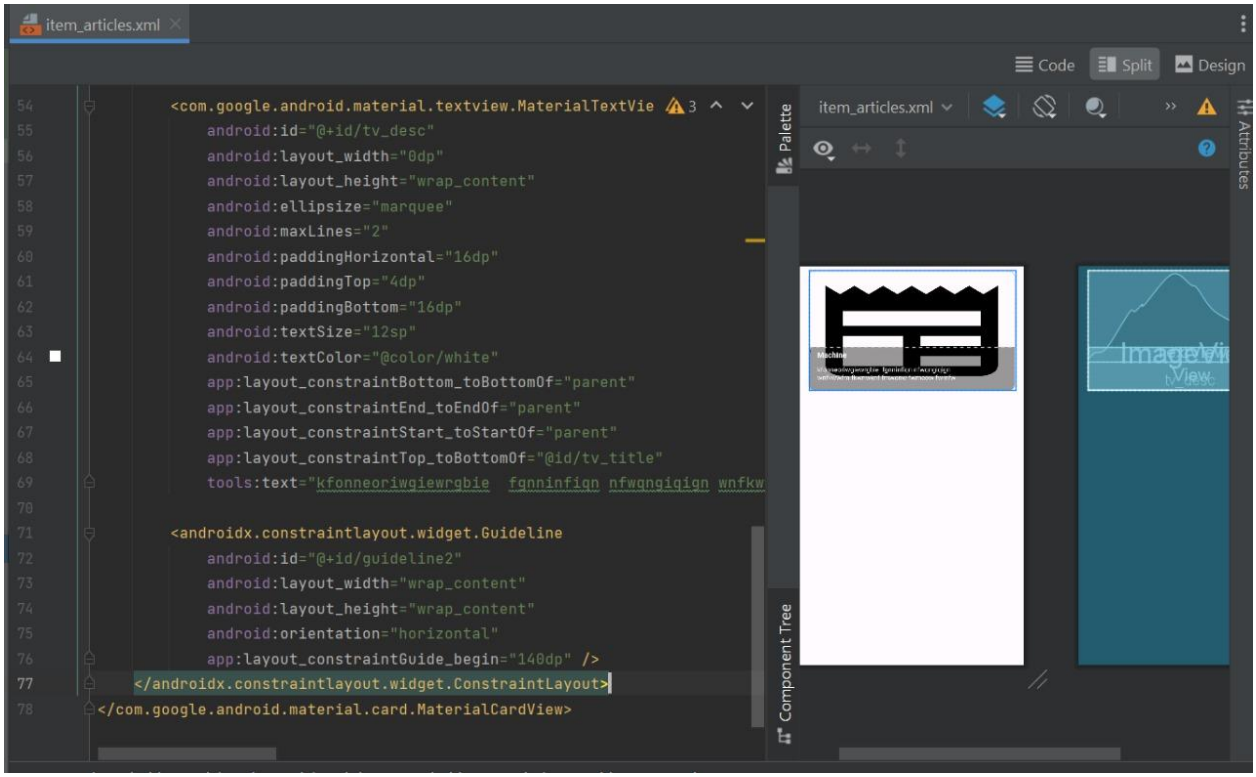




CODE 26: item.xml





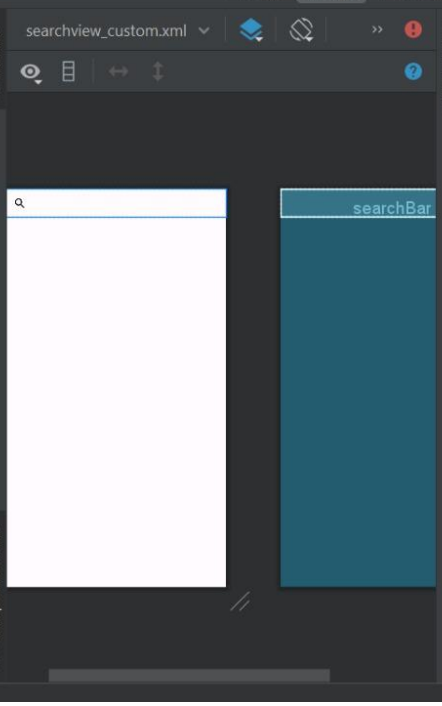


CODE 27: searchview_custom.xml

searchview_custom.xml

Code Split Design

```xml
<?xml version="1.0" encoding="utf-8"?>
<merge 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="wrap_content"
    android:orientation="horizontal"
    android:weightSum="16"
    android:paddingVertical="2dp"
    tools:parentTag="android.widget.LinearLayout">

    <androidx.appcompat.widget.SearchView
        android:id="@+id/searchBar"
        android:layout_weight="14"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:iconifiedByDefault="false"
        android:layout_gravity="center_vertical"
        android:theme="@style/SearchViewText"
        app:iconifiedByDefault="false"
        app:queryBackground="@android:color/transparent"
        app:searchIcon="@drawable/ic_search"/>

    <com.google.android.material.imageview.ShapeableImageView
        android:id="@+id/btnSearchUtil"
        android:layout_weight="2"
        android:layout_width="wrap_content"
```



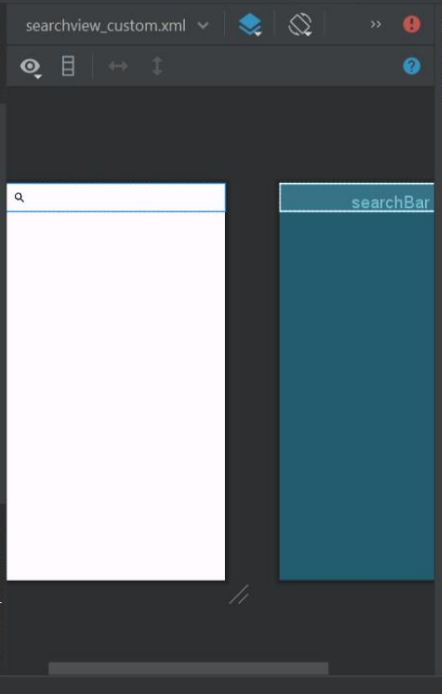
Palette Component Tree Attributes

merge

searchview_custom.xml

Code Split Design

```xml
    <androidx.appcompat.widget.SearchView
        android:id="@+id/searchBar"
        android:layout_weight="14"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:iconifiedByDefault="false"
        android:layout_gravity="center_vertical"
        android:theme="@style/SearchViewText"
        app:iconifiedByDefault="false"
        app:queryBackground="@android:color/transparent"
        app:searchIcon="@drawable/ic_search"/>

    <com.google.android.material.imageview.ShapeableImageView
        android:id="@+id/btnSearchUtil"
        android:layout_weight="2"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:scaleType="centerInside"
        android:visibility="gone"
        android:layout_marginEnd="16dp"
        android:layout_gravity="center_vertical"
        android:foreground="@android:color/transparent"
        android:background="@android:color/transparent"/>

</merge>
```

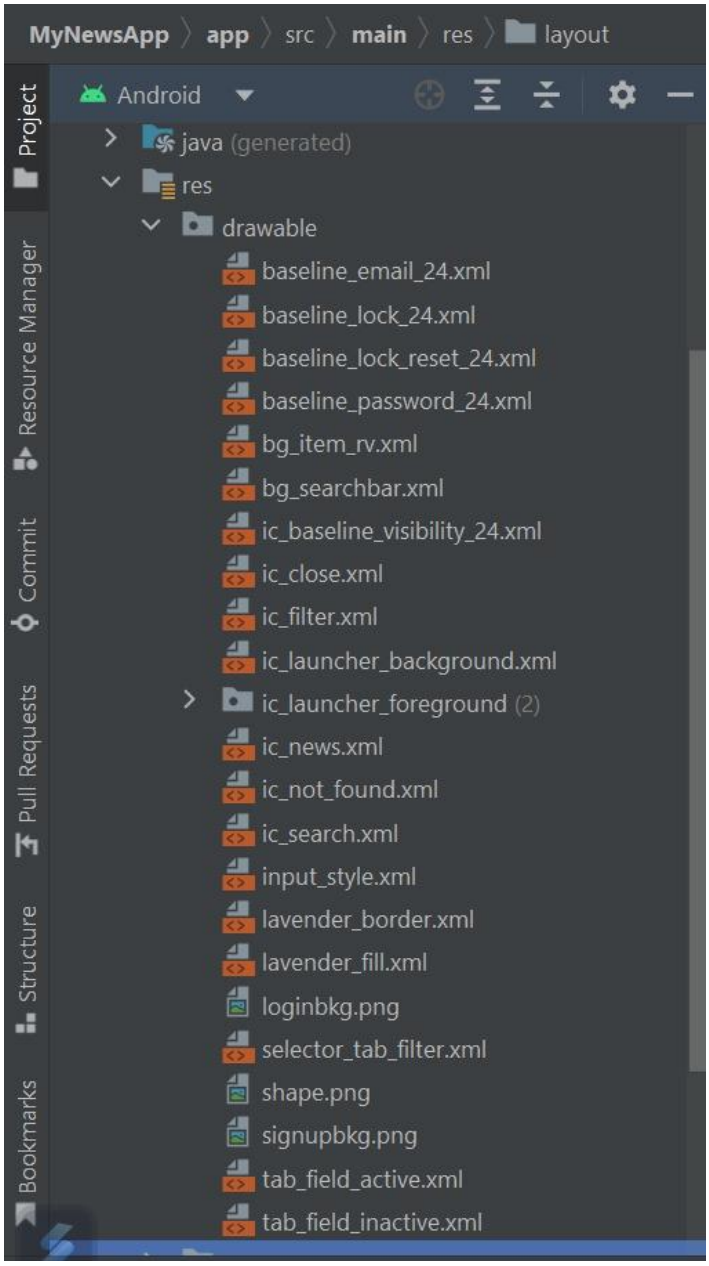


Palette Component Tree Attributes

merge

DRAWABLE RESOURCES

Take a screenshot of an image from your web browser, and drop it into the drawable folder. Refactor the image.

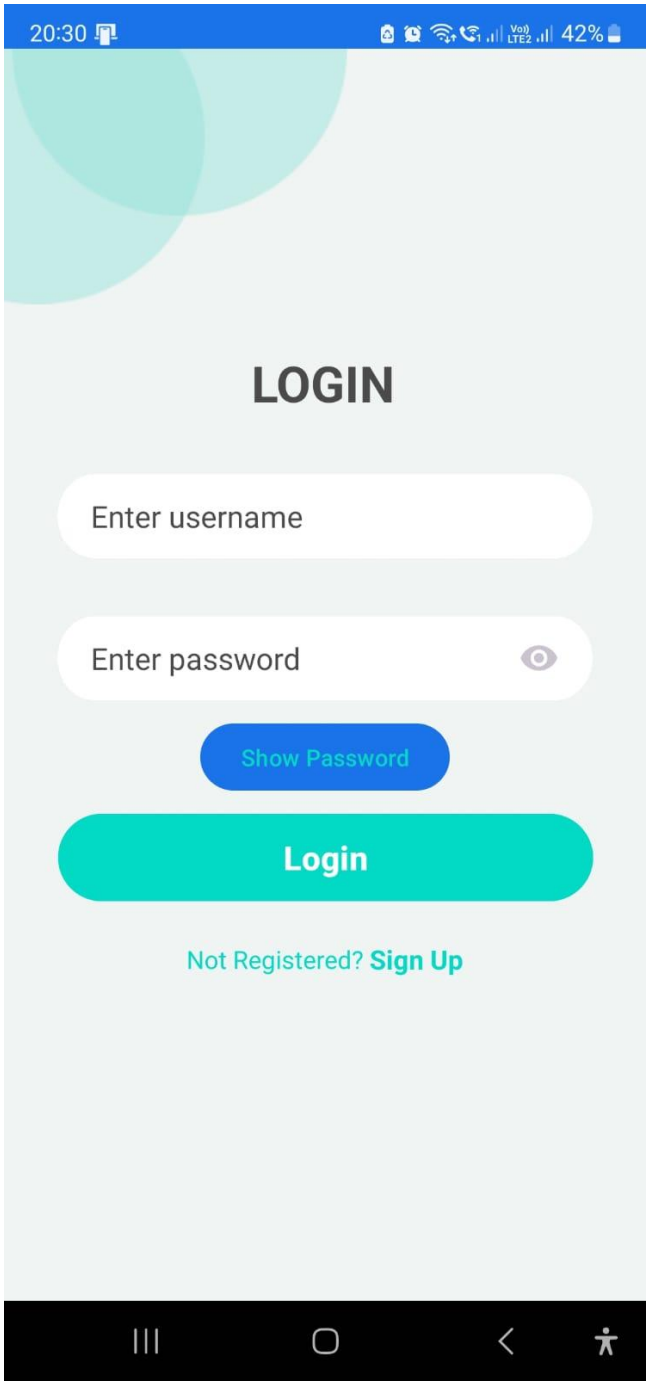


OUTPUT:

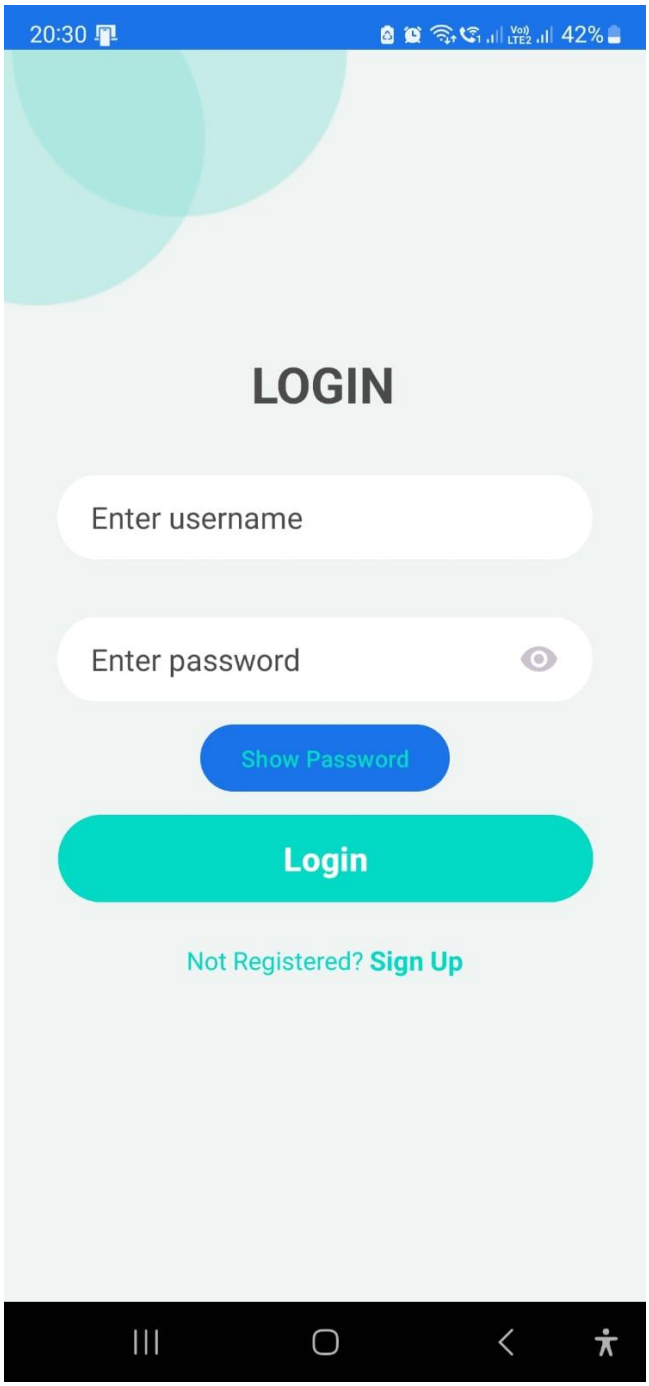
1. **Log-in Screen**
2. **Sign-up Screen**
3. **Home Page**
4. **News Page**
5. **Specific News Page**

OUTPUT 1: Log-in Screen

LIGHT MODE:

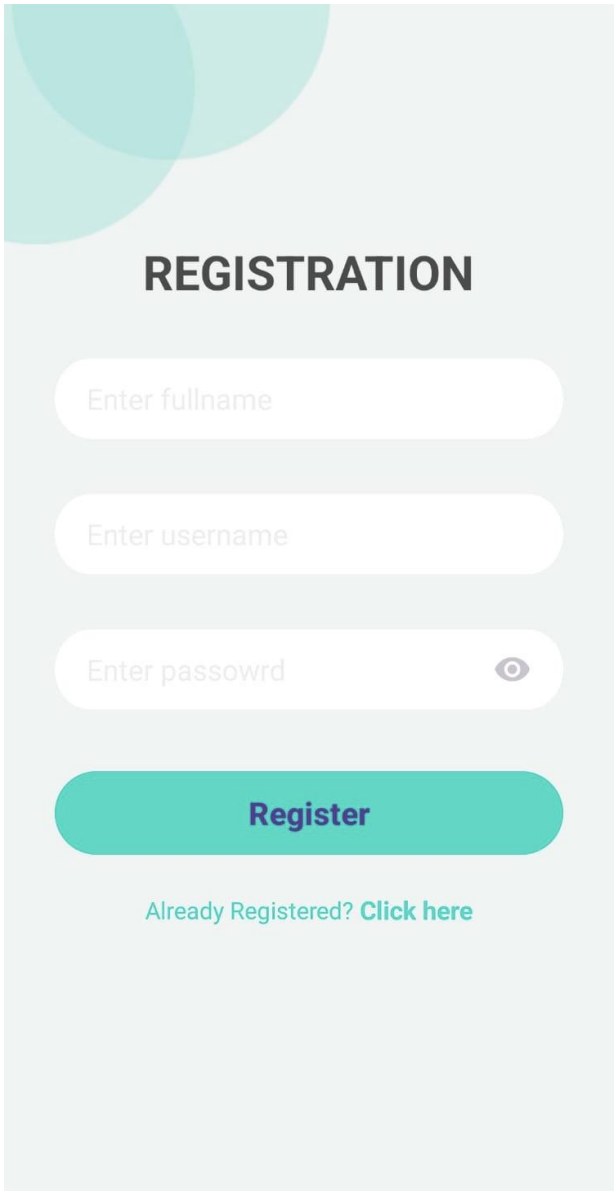


DARK MODE:

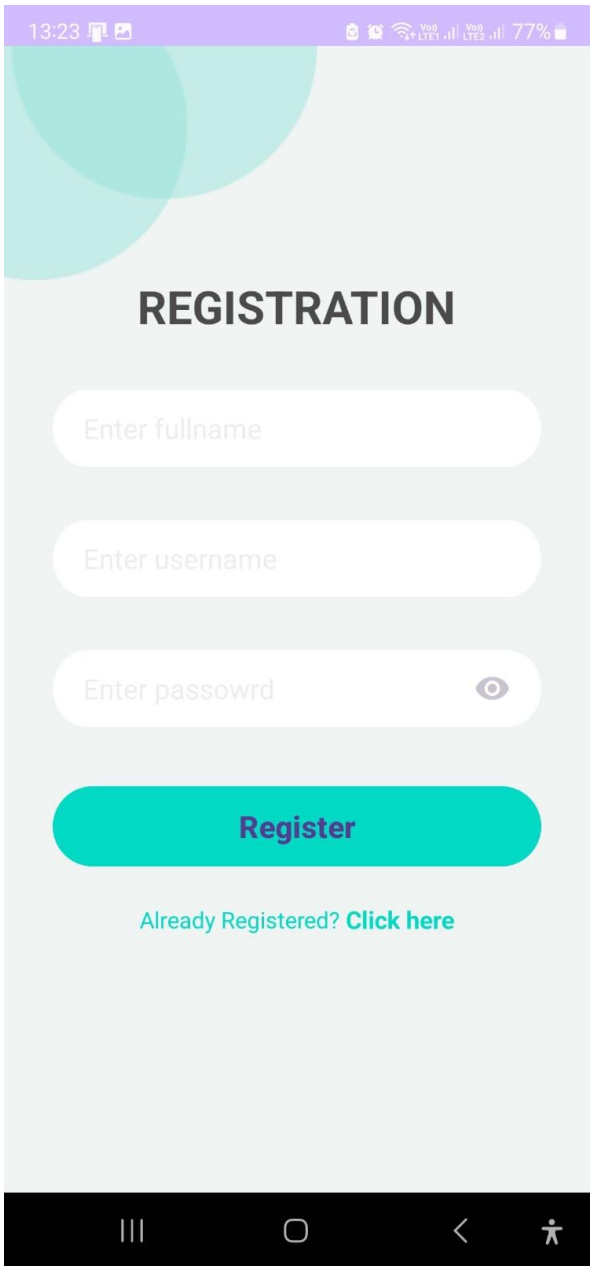

20:30
42%
LOGIN
Enter username
Enter password
Show Password
Login
Not Registered? Sign Up

OUTPUT 2: Sign-up Screen

LIGHT MODE:

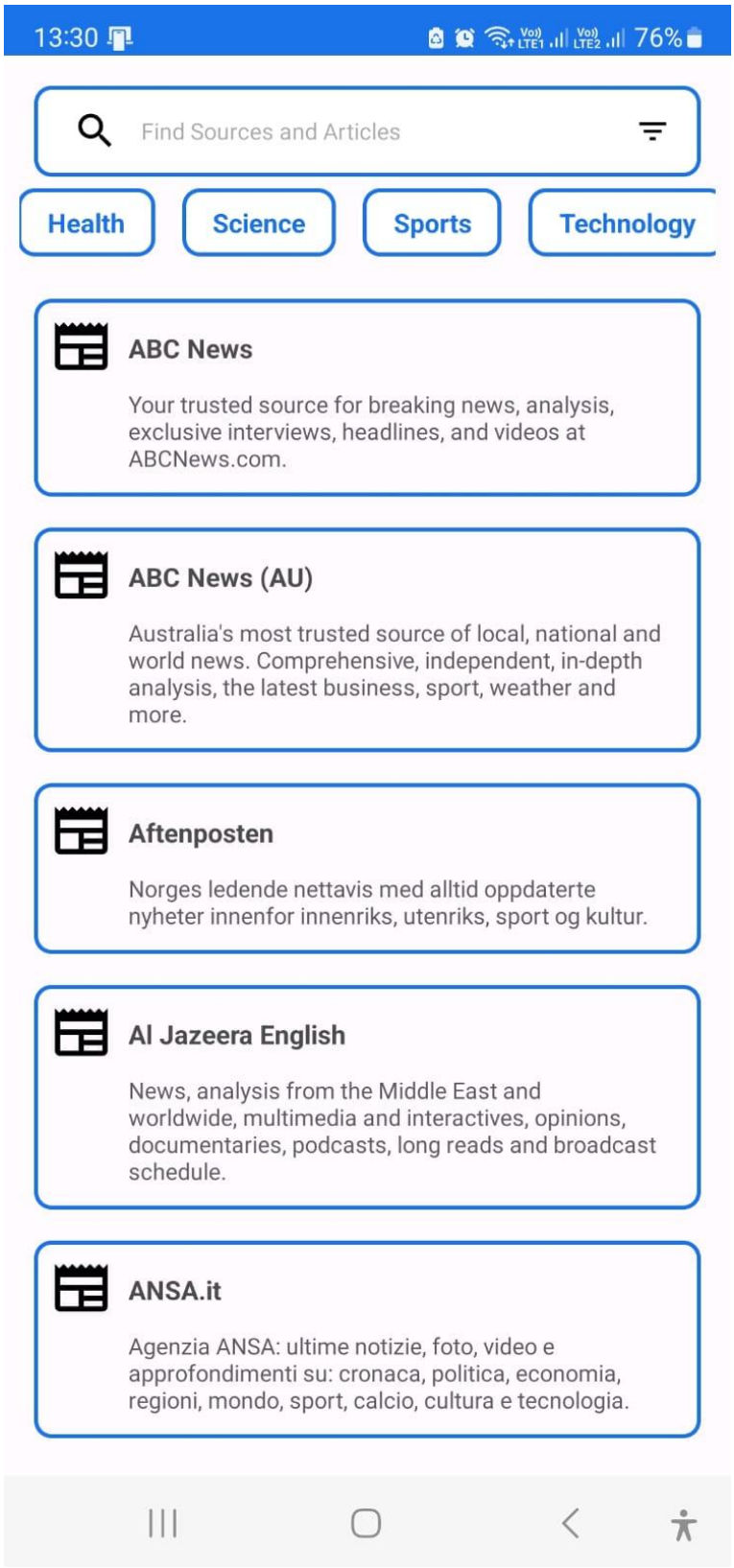


DARK MODE:

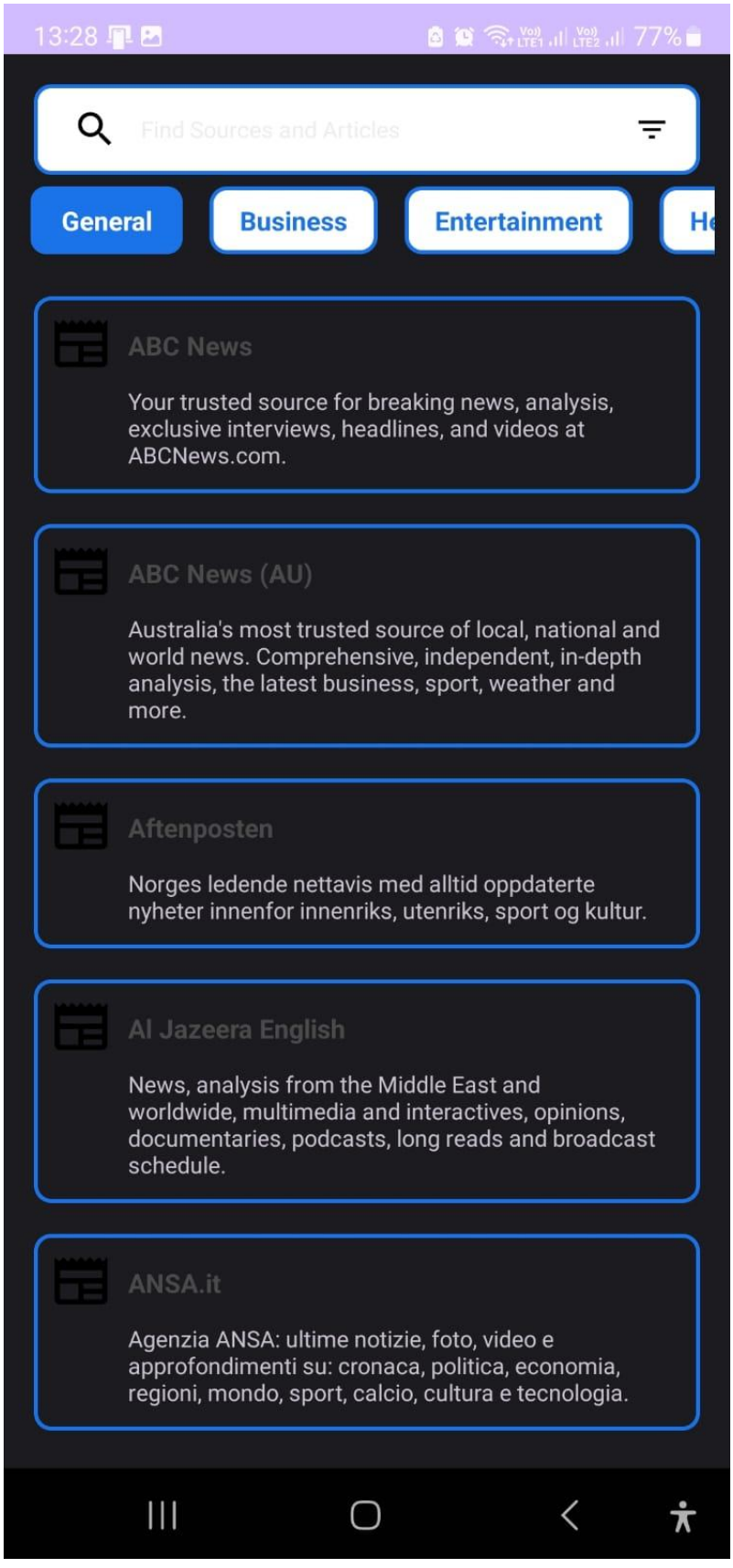

13:23
77%
REGISTRATION
Enter fullname
Enter username
Enter passowrd
Register
Already Registered? Click here

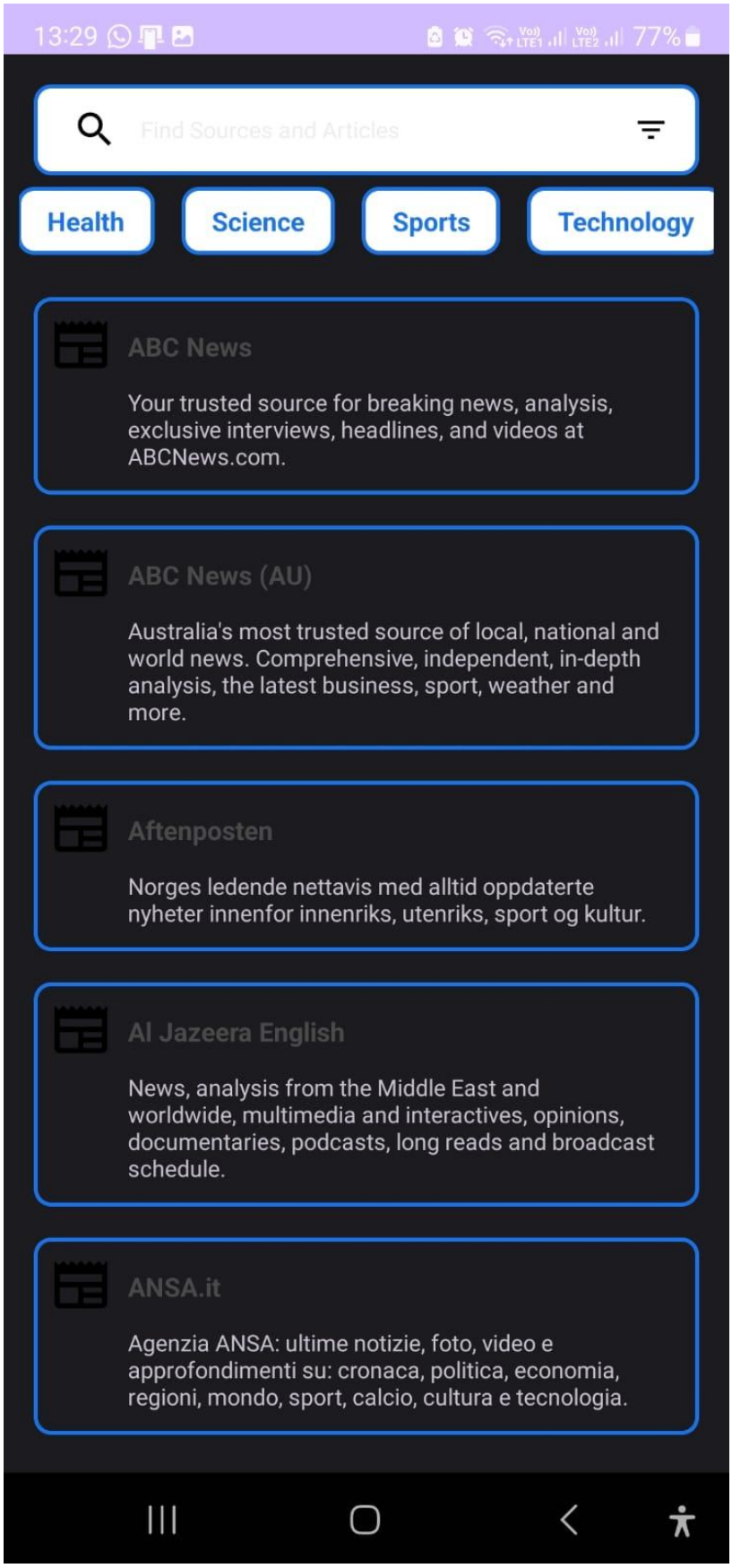
OUTPUT 3: Home Page

LIGHT MODE:



DARK MODE:


13:28
77%
Find Sources and Articles
General
Business
Entertainment
ABC News
Your trusted source for breaking news, analysis, exclusive interviews, headlines, and videos at ABCNews.com.
ABC News (AU)
Australia's most trusted source of local, national and world news. Comprehensive, independent, in-depth analysis, the latest business, sport, weather and more.
Aftenposten
Norges ledende nettavis med alltid oppdaterte nyheter innenfor innenriks, utenriks, sport og kultur.
Al Jazeera English
News, analysis from the Middle East and worldwide, multimedia and interactives, opinions, documentaries, podcasts, long reads and broadcast schedule.
ANSA.it
Agenzia ANSA: ultime notizie, foto, video e approfondimenti su: cronaca, politica, economia, regioni, mondo, sport, calcio, cultura e tecnologia.


13:29
77%
Find Sources and Articles
Health
Science
Sports
Technology
ABC News
Your trusted source for breaking news, analysis, exclusive interviews, headlines, and videos at ABCNews.com.
ABC News (AU)
Australia's most trusted source of local, national and world news. Comprehensive, independent, in-depth analysis, the latest business, sport, weather and more.
Aftenposten
Norges ledende nettavis med alltid oppdaterte nyheter innenfor innenriks, utenriks, sport og kultur.
Al Jazeera English
News, analysis from the Middle East and worldwide, multimedia and interactives, opinions, documentaries, podcasts, long reads and broadcast schedule.
ANSA.it
Agenzia ANSA: ultime notizie, foto, video e approfondimenti su: cronaca, politica, economia, regioni, mondo, sport, calcio, cultura e tecnologia.

OUTPUT 4: News Page

LIGHT MODE:

DARK MODE:


13:29
77%
Articles
China Premier Li seeks to bolster his country's economic outlook at the Shanghai export fair
Seeking to generate excitement around a largely lackluster economy, Chinese Premier Li Qiang is pledging to continue
abc NEWS
China Premier Li seeks to bolster his country's economic outlook at the Shanghai export fair
Seeking to generate excitement around a largely lackluster economy, Chinese Premier Li Qiang is pledging to continue
abc NEWS
Turkey's main opposition party elects Ozgur Ozel as new leader
Turkey's main opposition party has voted for fresh

OUTPUT 5: Specific News Page

Light:

DARK MODE:


13:29
77%
ABC News
abc NEWS
China Premier Li seeks to bolster his country's economic outlook at Shanghai trade fair
Seeking to generate excitement around a largely lackluster economy, Chinese Premier Li Qiang is pledging to continue deepening reforms, expand free trade zones and relax market access for foreign investment
By The Associated Press
November 5, 2023, 1:13 PM

GITHUB LINK: https://github.com/smartinternz02/SI-GuidedProject-587104-1696656485