NIT3213 Final Assignment – Android App

Developer

Name: Rezaul Karim
Student ID: *s8112591*
Unit: NIT3213 – Mobile Application Development

Project Overview

This Android application was developed for the NIT3213 final assignment to demonstrate proficiency in:

- **API integration**
- **Fragment-based navigation**
- **Dependency Injection using Hilt**
- **Clean architecture**
- **Unit testing with JUnit**

The app includes:

- **Login Screen**
- **Dashboard with RecyclerView**
- **Details Screen**

API Information

Base URL: `https://nit3213api.onrender.com/`
Endpoints:

1. **Login (POST)**
   `/footscray/auth` or `/sydney/auth` or `/ort/auth`
   Request body:

```
{
  "username": "YourFirstName",
  "password": "sYourStudentID"
}
```

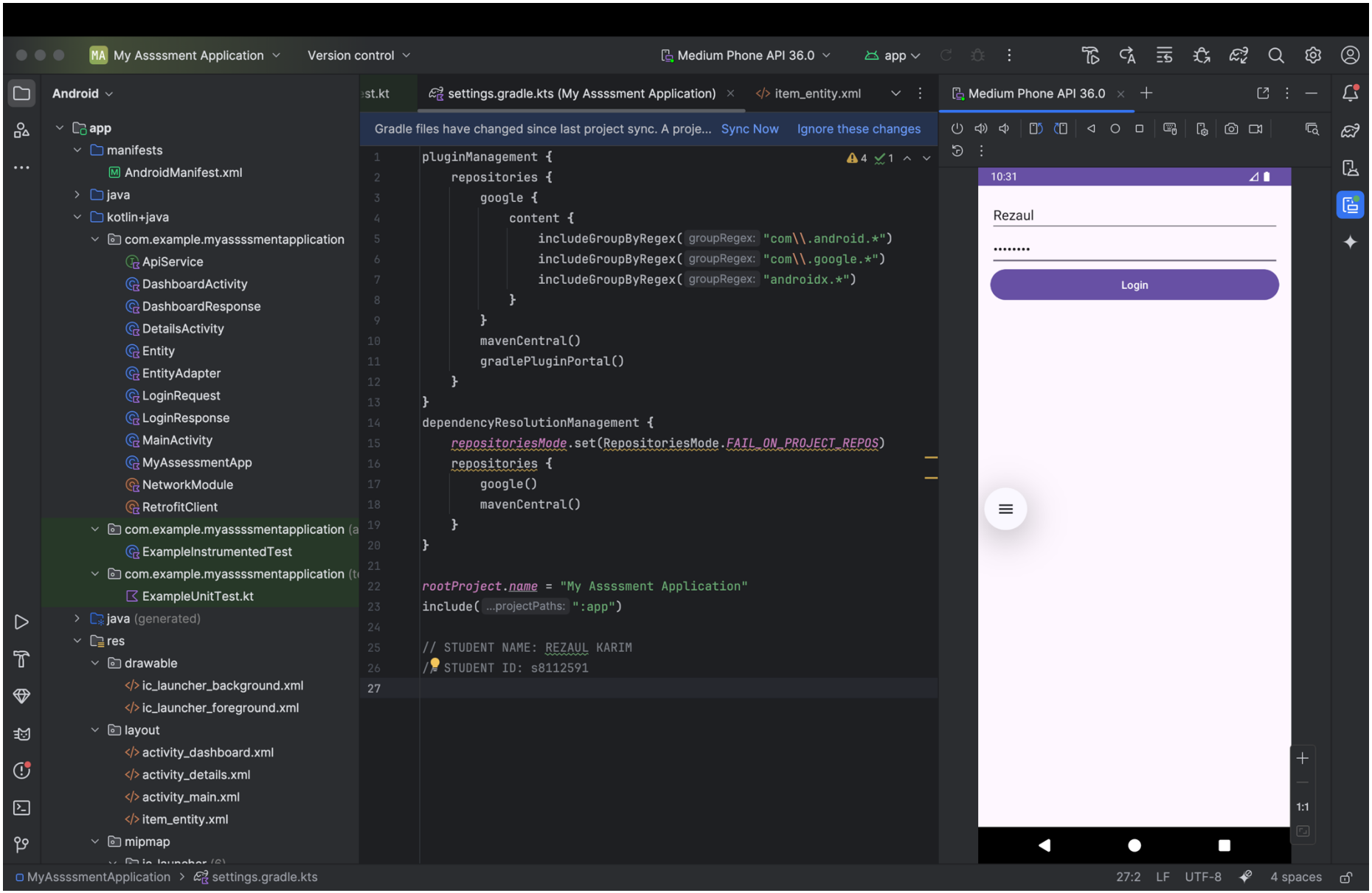
6. **Dashboard (GET)**
   `/dashboard/{keypass}`

Login Credentials Format

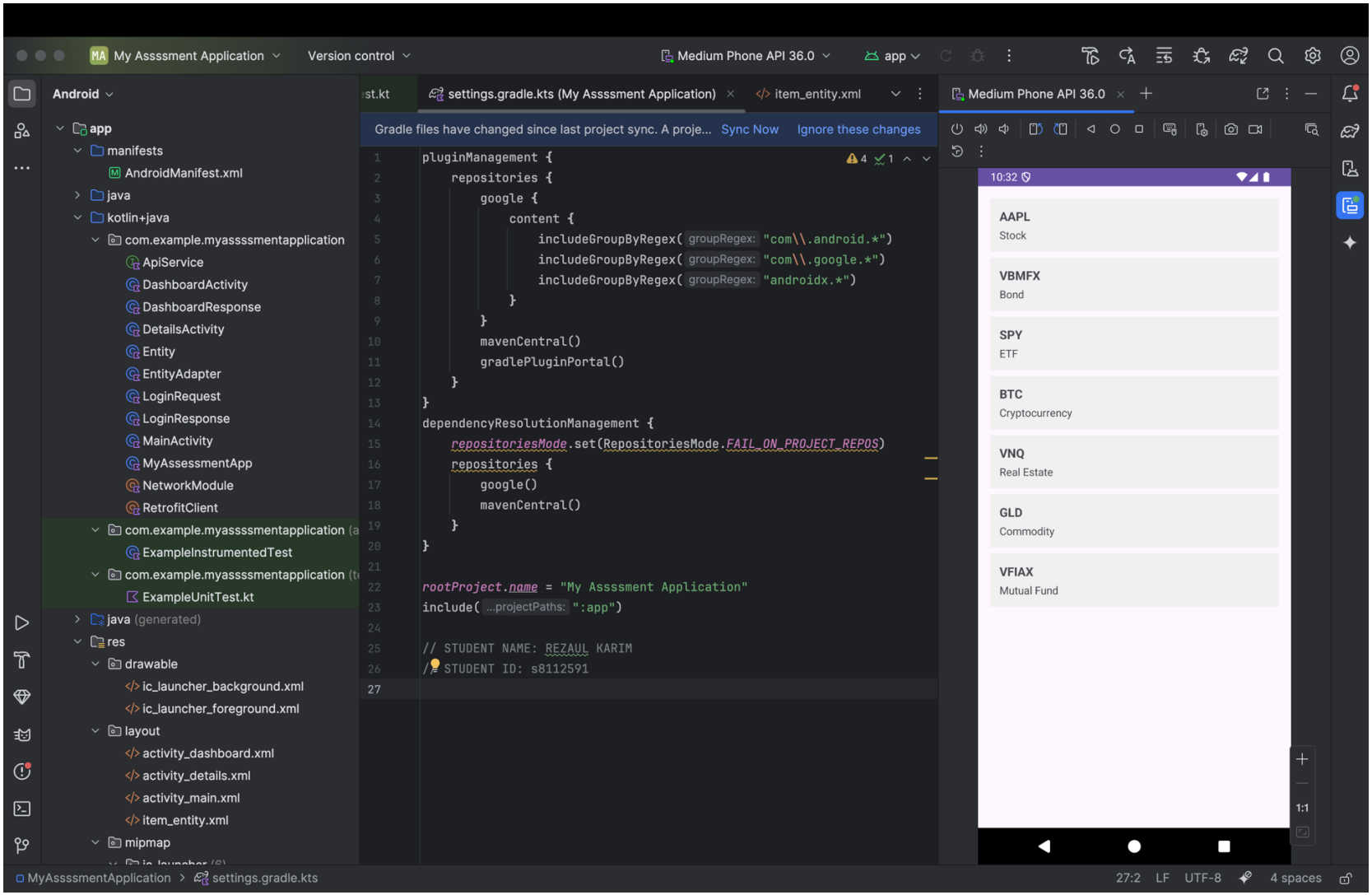
- **Username:** Your first name (e.g., `Rezaul`)
- **Password:** Your student ID (e.g., `s8112591`

Features

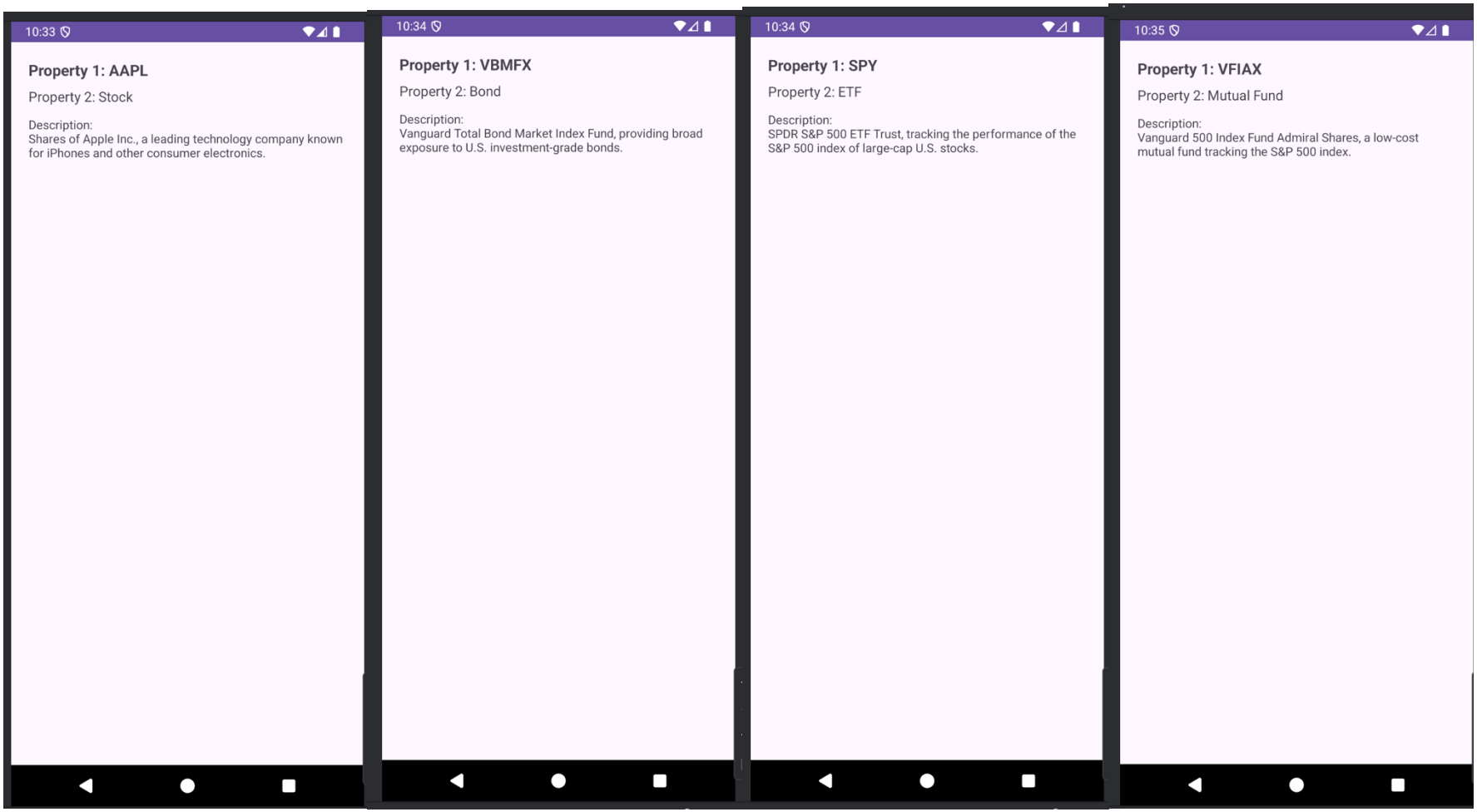
Login Screen: Authenticates users using provided credentials


My Asssment Application
Version control
Medium Phone API 36.0
app
Android
app
manifests
AndroidManifest.xml
java
kotlin+java
com.example.myassssmentapplication
ApiService
DashboardActivity
DashboardResponse
DetailsActivity
Entity
EntityAdapter
LoginRequest
LoginResponse
MainActivity
MyAssessmentApp
NetworkModule
RetrofitClient
com.example.myassssmentapplication (a
ExampleInstrumentedTest
com.example.myassssmentapplication (t
ExampleUnitTest.kt
java (generated)
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
activity_dashboard.xml
activity_details.xml
activity_main.xml
item_entity.xml
mipmap
st.kt
settings.gradle.kts (My Asssment Application)
item_entity.xml
Gradle files have changed since last project sync. A proje...
Sync Now
Ignore these changes
pluginManagement {
repositories {
google {
content {
includeGroupByRegex(groupRegex: "com\\.android.*")
includeGroupByRegex(groupRegex: "com\\.google.*")
includeGroupByRegex(groupRegex: "androidx.*")
}
}
mavenCentral()
gradlePluginPortal()
}
}
dependencyResolutionManagement {
repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
repositories {
google()
mavenCentral()
}
}
rootProject.name = "My Asssment Application"
include(...projectPaths: ":app")
// STUDENT NAME: REZAUL KARIM
// STUDENT ID: s8112591
Medium Phone API 36.0
10:31
Rezaul
Login
1:1
MyAssssmentApplication
settings.gradle.kts
27:2
LF
UTF-8
4 spaces

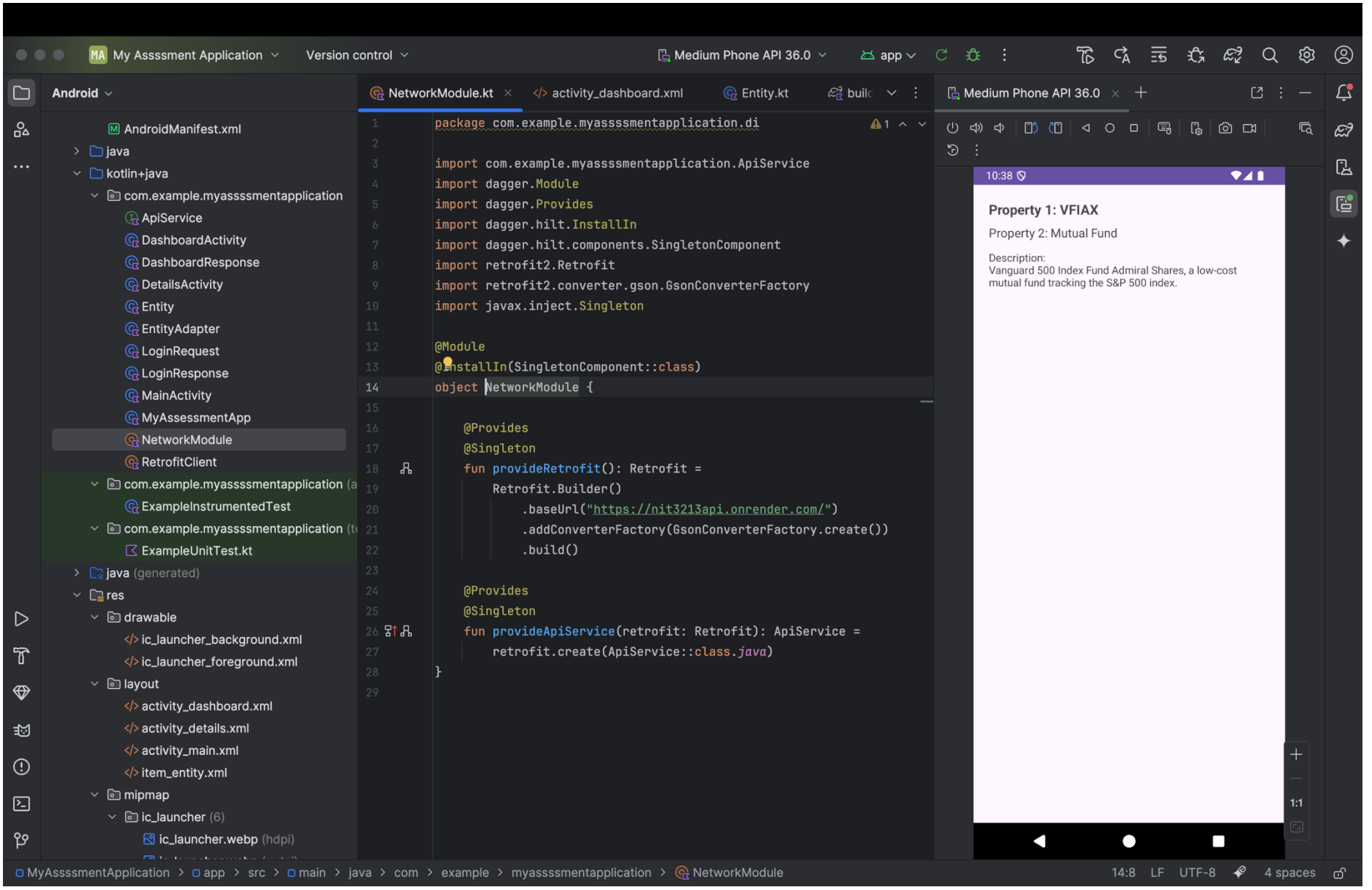
Dashboard: Displays list of entities using `RecyclerView`



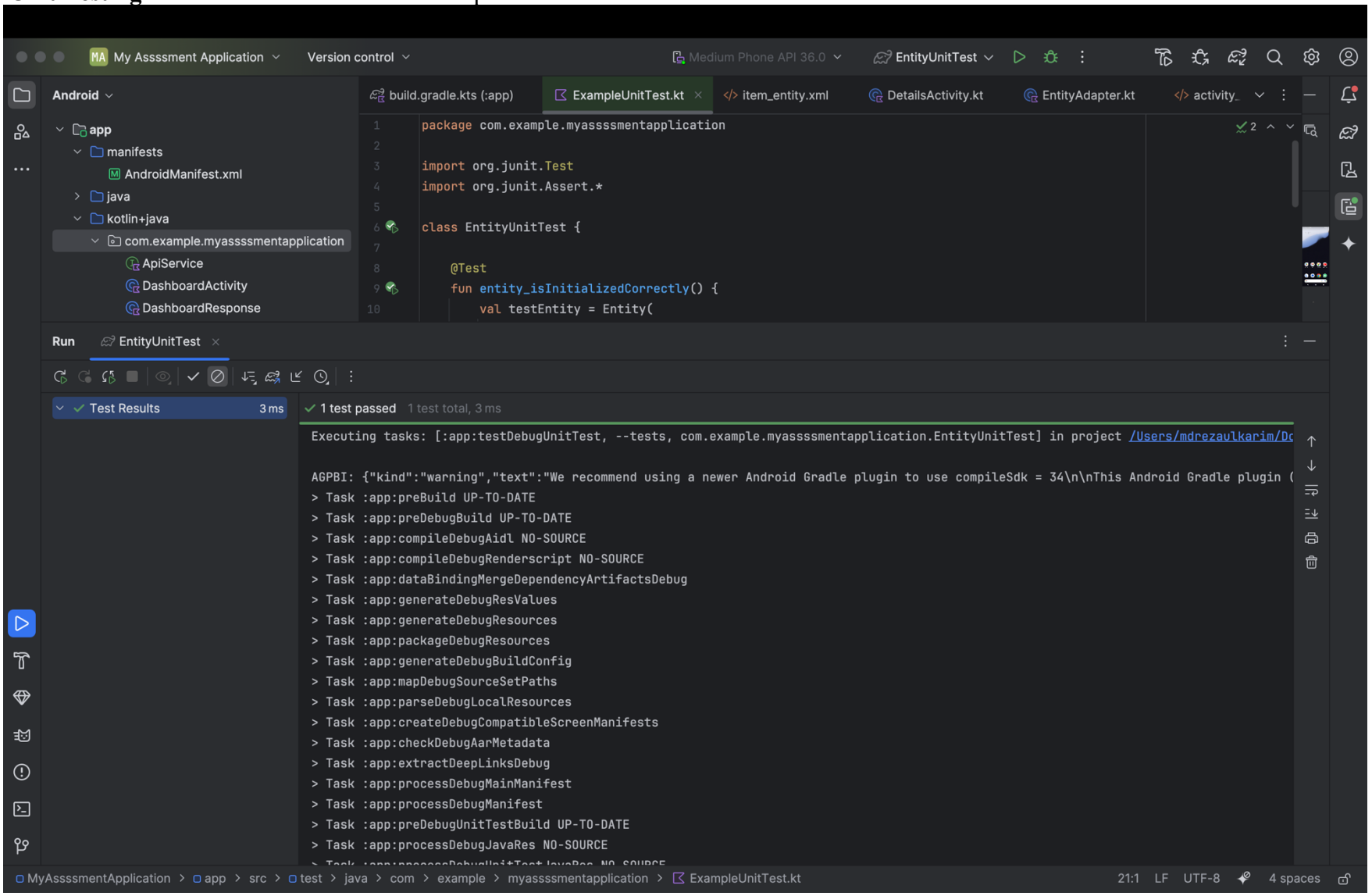
Details Screen: Shows detailed information from the API

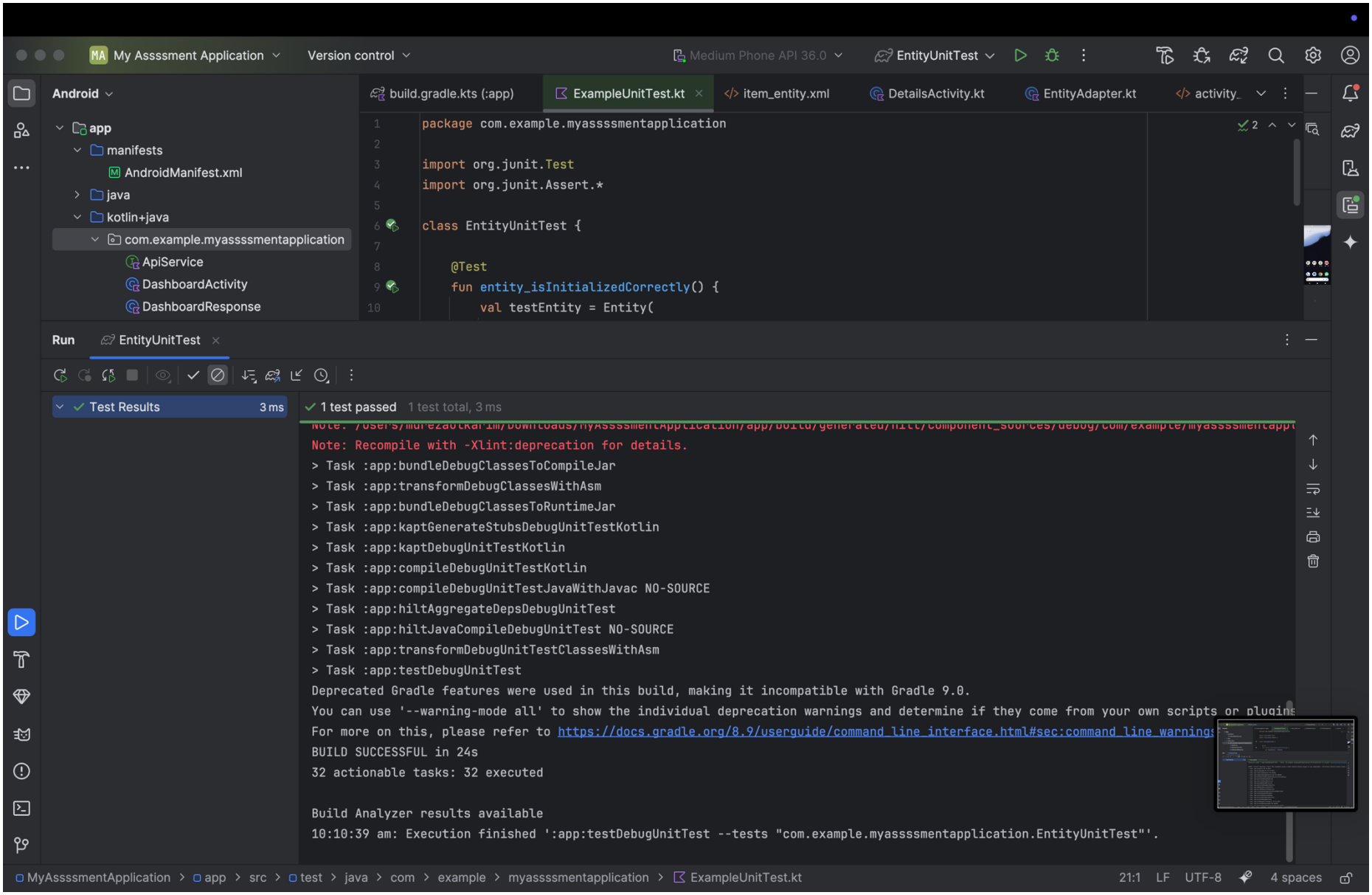


Dependency Injection: Implemented with **Hilt**


My Asssment Application
Version control
Medium Phone API 36.0
app
Android
NetworkModule.kt
activity_dashboard.xml
Entity.kt
AndroidManifest.xml
java
kotlin+java
com.example.myassssmentapplication
ApiService
DashboardActivity
DashboardResponse
DetailsActivity
Entity
EntityAdapter
LoginRequest
LoginResponse
MainActivity
MyAssessmentApp
NetworkModule
RetrofitClient
ExampleInstrumentedTest
ExampleUnitTest.kt
java (generated)
res
drawable
ic_launcher_background.xml
ic_launcher_foreground.xml
layout
activity_dashboard.xml
activity_details.xml
activity_main.xml
item_entity.xml
mipmap
ic_launcher (6)
ic_launcher.webp (hdpi)
package com.example.myassssmentapplication.di
import com.example.myassssmentapplication.ApiService
import dagger.Module
import dagger.Provides
import dagger.hilt.InstallIn
import dagger.hilt.components.SingletonComponent
import retrofit2.Retrofit
import retrofit2.converter.gson.GsonConverterFactory
import javax.inject.Singleton
@Module
@InstallIn(SingletonComponent::class)
object NetworkModule {
@Provides
@Singleton
fun provideRetrofit(): Retrofit =
Retrofit.Builder()
.baseUrl("https://nit3213api.onrender.com/")
.addConverterFactory(GsonConverterFactory.create())
.build()
@Provides
@Singleton
fun provideApiService(retrofit: Retrofit): ApiService =
retrofit.create(ApiService::class.java)
}
Medium Phone API 36.0
10:38
Property 1: VFIAX
Property 2: Mutual Fund
Description:
Vanguard 500 Index Fund Admiral Shares, a low-cost mutual fund tracking the S&P 500 index.
1:1
MyAssssmentApplication
app
src
main
java
com
example
myassssmentapplication
NetworkModule
14:8
LF
UTF-8
4 spaces

Unit Testing: Includes test cases for core components




My Asssment Application
Version control
Medium Phone API 36.0
EntityUnitTest
Android
app
manifests
AndroidManifest.xml
java
kotlin+java
com.example.myassssmentapplication
ApiService
DashboardActivity
DashboardResponse
build.gradle.kts (:app)
ExampleUnitTest.kt
item_entity.xml
DetailsActivity.kt
EntityAdapter.kt
activity_
package com.example.myassssmentapplication
import org.junit.Test
import org.junit.Assert.*
class EntityUnitTest {
@Test
fun entity_isInitializedCorrectly() {
val testEntity = Entity(
Run
EntityUnitTest
Test Results 3 ms
1 test passed 1 test total, 3 ms
Note: Recompile with -Xlint:deprecation for details.
> Task :app:bundleDebugClassesToCompileJar
> Task :app:transformDebugClassesWithAsm
> Task :app:bundleDebugClassesToRuntimeJar
> Task :app:kaptGenerateStubsDebugUnitTestKotlin
> Task :app:kaptDebugUnitTestKotlin
> Task :app:compileDebugUnitTestKotlin
> Task :app:compileDebugUnitTestJavaWithJavac NO-SOURCE
> Task :app:hiltAggregateDepsDebugUnitTest
> Task :app:hiltJavaCompileDebugUnitTest NO-SOURCE
> Task :app:transformDebugUnitTestClassesWithAsm
> Task :app:testDebugUnitTest
Deprecated Gradle features were used in this build, making it incompatible with Gradle 9.0.
You can use '--warning-mode all' to show the individual deprecation warnings and determine if they come from your own scripts or plugins.
For more on this, please refer to https://docs.gradle.org/8.9/userguide/command_line_interface.html#sec:command_line_warnings
BUILD SUCCESSFUL in 24s
32 actionable tasks: 32 executed
Build Analyzer results available
10:10:39 am: Execution finished ':app:testDebugUnitTest --tests "com.example.myassssmentapplication.EntityUnitTest"'.
MyAssssmentApplication > app > src > test > java > com > example > myassssmentapplication > ExampleUnitTest.kt
21:1 LF UTF-8 4 spaces

Unit Tests

- Includes basic unit test for ViewModel logic.
- Test class: `EntityUnitTest.kt`
- Run via `Build > Run Tests` in Android Studio

Tech Stack

- Kotlin
- Android SDK
- Retrofit
- Hilt
- ViewModel + LiveData
- RecyclerView
- JUnit

How to Build & Run

1. **Clone the repository**
2. `git clone https://github.com/your-username/NIT3213-Final-Assignment.git`
3. `cd NIT3213-Final-Assignment`
4. **Open in Android Studio**
5. **Run the App** using emulator or physical device
6. **API Usage:** Ensure internet permission is enabled in `AndroidManifest.xml`.

License

This project is for academic purposes only (Victoria University – NIT3213).